Advanced Micro Devices

MOS Microprocessors and Peripherals Data Book

The International Standard of Quality guarantees a 0.05% AQL on all electrical parameters, AC and DC, over the entire operating range.

INT·STD·500

901 Thompson Place, P.O. Box 3453, Sunnyvale, California 94088
(408) 732-2400 TWX: 910-339-9280 TELEX: 34-6306

Printed in U.S.A.

Index Section

INTRODUCTION TO THE "MMP DATA BOOK"

Advanced Micro Devices is a proven leader in the design and manufacture of high performance state-of-the-art MOS VLSI peripheral circuits ... more than a step ahead of the rest of the industry. Futhermore, AMD offers many of the popular microprocessors, including the industry standard iAPX86 family. These powerful microprocessors, when coupled with the performance of AMD's peripheral circuits, will bring out the maximum potential of your designs.

Section 2 of the "MOS Microprocessors and Peripherals Databook" describes the general purpose peripheral circuits with generic bus structures allowing easy interface to any MOS Microprocessor. Peripherals, such as the Am9516A and Am9517A DMA Controllers, the Am9519A Interrupt Controller, the Am9513A Counter/Timer, the Z8530 Serial Communications Controller, and the Z8536 Counter and I/O, provide functional enhancements of well-established, popular, system requirements.

Of special interest are multi-technology chip sets featuring superior implementation of important peripheral subsystems. These innovative solutions include the Am8052/8152A CRT Controller chip set, the Am9580/9581 Disk Controller chip set, and the Am7990/7992A Ethernet Controller chip set.

A third category of devices integrates specialized functions onto silicon eliminating the need for large numbers of MSI chips. These peripherals include the Am9520/AmZ8065 Burst Error Processors, the Am9511A/9512 Arithmetic Processors, and the NEW Am7970 Compression Expansion Processor. Also included in Section 2 are the Am9518/9568/AmZ8068 Data Ciphering Processors, and the Z8038/8060 FIFO.

The **AMD/Intel Alliance**, a 10-year agreement established in 1981 covering the mutual exchange of **complete** product data bases, not only establishes broad multiple sourcing for the iAPX86 family, but also ensures supplier to supplier device compatibility. AMD's superior N-channel silicon-gate processing brings you the industry's first 10MHz 8086 and 80186. Careful attention is paid to high performance – but not at the expense of compatibility.

For additional system performance, there is the 80286 (6MHz and 8MHz) 16-bit microprocessor, featuring virtual memory, multi-user, and multi-tasking capability and memory protection. Section 3 describes the iAPX86 family devices available from AMD. Numerous popular peripherals with 82XX designations complement the iAPX86 family, including the 8237A, 8232, and 8231A originally designed by AMD.

The popular 8051 family of 8-bit single chip microcomputers is described in Section 4. It includes the 8751H with internal EPROM and the Am9761H with twice the on-chip EPROM. Also covered are the 8051AH ROM based MCU and 8031AH ROM-less MCU with performance to 15MHz. CMOS versions of the 8051AH and 8031AH will be available in 1985.

The Z8000 Family of microprocessors and peripherals, described in Section 5, offers high performance, a clean and regular, minicomputer-like architecture, and efficient bus structure. Most of the peripherals described in Section 2 are easily compatible with these CPUs.

Section 6 covers well-established more mature 8-bit microprocessors available from AMD. The Am9080A is functionally identical to the 8080A, but offers improved electrical parameters and a higher fan-out.

Section 7 details packaging configurations, including the new Plastic Leaded Chip Carriers (PLCC's) in 28, 44, and 68 lead versions. AMD PLCC's meet JEDEC outlines and are ideal for high density, low cost surface mount applications.

AMD is committed to innovative, high performance system solutions using the most appropriate technologies – bipolar, MOS or CMOS, often combining high complexity MOS parts with very high speed bipolar devices for an optimized system solution. Continued introductions of high performance MOS CPU's and microcomputers will be oriented toward the iAPX86 family with MOS VLSI and CMOS microprocessors, microcomputers, and peripheral circuits aimed toward easy interface with not only iAPX86 CPU's but other popular 16- and 32-bit microprocessors.

AMD's strengths are: innovative systems definitions combined with aggressive circuit design of peripheral functions, a broad technological base for high-performance semiconductor processing, high volume assembly, innovative packaging techniques, and the best guaranteed quality in the industry.

Along with bipolar and MOS memories, bipolar microprocessors, array processors, logic and interface, and innovative telecom circuits, the MOS microprocessors and peripheral circuits described in this Data Book are an important part of this strategy.

Other Data Books available from Advanced Micro Devices cover Bipolar and MOS memories, Bipolar microprocessors, Logic and Interface Analog and communications. For these and other AMD product literature, call AMD Literature Distribution or your local AMD sales office.

Refer to page 7-1 for Essential Information on Military Devices

NUMERIC INDEX

FUNCTIONAL INDEX

Advanced General Purpose Peripherals

Part Number	Description
Am7970	Compression Expansion Processor
Am7990	Local Area Network Controller for Ethernet
Am7992A	Serial Interface Adapter
AmZ8016	DMA Transfer Controller
Z8030/Z8530	Serial Communications Controller
Z8031/Z8531	Asynchronous Serial Communications Controller
Z8036/Z8536	Counter I/O
Z8038	FIFO I/O Port
Am8052/8152A/ 8153A	CRT Controller Chip Set – General Information
Am8052	Alphanumeric CRT Controller
Am8152/8153A	Video System Controller
Z8060	Buffer Unit and FIFO Expander
Am9511A	Arithmetic Processor
Am9512	Arithmetic Processor
Am9513A/AmZ8073A	System Timing Controller
Am9516A	Universal DMA Controller
Am9517A	Multimode DMA Controller
Am9518/AmZ8068	Data Ciphering Processor
Am9519A	Universal Interrupt Controller
Am9520/Am9521/ AmZ8065	Burst Error Processor
Am9568	Data Ciphering Processor
Am9580	Hard Disk Controller
Am9581	Floppy/Hard Disk Data Separator

iAPX86 Family

Part Number	Description
80186	High Integration 16-Bit Microprocessor
80286	High Performance 16-Bit Microprocessor with Memory Management and Protection
8086	16-Bit Microprocessor
8087	Numeric Data Coprocessor
8088	8-Bit Microprocessor CPU
82284	Clock Driver and Ready Interface for iAPX 286 Processors
82C288	Bus Controller for the iAPX 286 Processors
8231A	Arithmetic Processor (See the Am9511A in Section 2 for additional specifications.)
8232	Arithmetic Processor (See the Am9512 in Section 2 for additional specifications.)
8237A	Multimode DMA Controller
8251/Am9551	Programmable Communication Interface
8251A	Programmable Communication Interface
8253	Programmable Interval/Timer
8255A	Programmable Peripheral Interface
82C55A	CMOS Programmable Peripheral Interface
8259A	Programmable Interrupt Controller
82C59A	CMOS Programmable Interrupt Controller

Refer to page 7-1 for Essential Information on Military Devices

Single-Chip Microcomputers

Part Number	Description
8051 Family	The 8051 Family Datasheet
8051AH/8031AH	Single-Chip 8-Bit Microcomputer
80C31/80C51	CMOS Single Component 8-Bit Microcomputer
8751H/Am9761H	Single-Chip 8-Bit Microcomputer with 4K/8K Bytes of EPROM Program Memory

Z8000 Family

Part Number	Description
Z8001/Z8002	16-Bit Microprocessors
Z-Bus/68000	Microprogrammable Bus Translator

8-Bit Microprocessors

Part Number	Description
8080A/Am9080A	8-Bit Microprocessor
8085AH	8-Bit Microprocessor
8155H/8156H	2K SRAM with I/O Ports and Timer

Refer to page 7-1 for Essential Information on Military Devices

MOS Microprocessor Family

Selector Guide

	80286	80186	8086/88[t]	Z8001/2[tt]	8085AH-1 8085AH-2	68000
Clock Period	125ns	125ns	200ns	165ns	200ns	100ns
Clock Generator	82284	On-Chip	8284A	8127	On-Chip	N/A
Arithmetic Processing Unit	N/A	9511A-1 9512-1	8087	9511A-4 9512-1	9511A-4 9512-1	9511A 9512
Interrupt Controller	9519A 8259A	9519A 8259A	8259A-5	9519A-1	9519A-4 8259A-5	9519A
DMA Controller	9517A-5 9516A	9517A-5 9516A	8089 9516A 9517A-5	8016A	9517A-5	9517A 9516
Dynamic Memory Controller	2964B 2968/69/70	2964B 2968/69/70	2964B	2964B	2964B	2964B 2968/69/70
Serial I/O	8251A 8530A/8531A 8031A	8251A 8530A/8531A 8031A	8251A 8530A/8531A 8030A/8031A	8030A/8031A	8251A 8530A/8531A	8251A 8530A/8531A
Parallel I/O	8255A 8036A	8255A-5 8036A	8255A 8036A	8036A	8255A-5	8255A-5 8536A
Counter Timer I/O	9513A 8253-5 8073	9513A 8253-5 8073	9513A 8036A 8073	8073	9513A 8253-5	9513A 8253 8073
Fifo I/O	8038 8060	8038 8060	8038 8060	8038 8060	8038 8060	8038 8060
Data Ciphering Processor	8068 9568	8068 9568	8068 9568	8068	8068 9568	9518 8068
Error Detection and Correction	2960	2960	2960	2960	2960	2960
Burst Error Processor	8065 9520	8065 9520	8065 9520	8065 9520	8065 9520	8065 9520
CRT Controller	8052	8052	8052	8052	N/A	8052
I/O Processor	N/A	N/A	8089	N/A	N/A	N/A
R_{AM} I/O	N/A	N/A	N/A	N/A	8155/6-2	N/A
Bus Control/Arbiter	82C288	N/A	8288	N/A	N/A	N/A
Bus Latches	29841-6	29841-6	29841-6	29841-6	29841-6	29841-6
Bus Buffers	29827/28	29827/28	29827/28	29827/28	29827/28	29827/28
Bus Transceivers	29861-4	29861-4	29861-4	29861-4	29861-4	29861-4
EDC Buffers	2961/2	2961/2	2961/2	2961/2	2961/2	2961/2
RAM Drivers	2965/6	2965/6	2965/6	2965/6	2965/6	2965/6
Network Controllers	29380 7990 7991	29380 7990 7991	29380 7990 7991	29380 7990 7991	N/A	29380 7990 7991
Compression/Expression Processor	7970	7970	7970	7970	7970	7970
Disk Controller	9580 9581	9580 9581	9580 9581	9580 9581	9580 9581	9580 9581

[t] 8086 is a trademark of Intel, Inc.

[tt] Z8000 is a trademark of Zilog, Inc.

Refer to page 7-1 for Essential Information on Military Devices

INTERFACE SUPPORT PRODUCTS

STANDARD 20-pin PAL FAMILY:*	
Part Number	**Description**
AmPAL 16R8	20-pin IMOX Programmable Array Logic Elements
AmPAL 16R6	20-pin IMOX Programmable Array Logic Elements
AmPAL 16R4	20-pin IMOX Programmable Array Logic Elements
AmPAL 16L8	20-pin IMOX Programmable Array Logic Elements
AmPAL 16H8	20-pin IMOX Programmable Array Logic Elements
AmPAL 16LD8	20-pin IMOX Programmable Array Logic Elements
AmPAL 16HD8	20-pin IMOX Programmable Array Logic Elements
HALF-POWER 20-pin FAMILY:*	
Part Number	**Description**
AmPAL 16R8L	20-pin IMOX Programmable Array Logic Elements
AmPAL 16R6L	20-pin IMOX Programmable Array Logic Elements
AmPAL 16R4L	20-pin IMOX Programmable Array Logic Elements
AmPAL 16L8L	20-pin IMOX Programmable Array Logic Elements
AmPAL 16H8L	20-pin IMOX Programmable Array Logic Elements
AmPAL 16LD8L	20-pin IMOX Programmable Array Logic Elements
AmPAL 16HD8L	20-pin IMOX Programmable Array Logic Elements
INTERFACE SUPPORT PRODUCTS**	
Part Number	**Description**
25LS2521	8-Bit Comparator
25LS2535	8-Bit Multiplexer for Control Storage
25LS2536	8-Bit Decoder with Control Storage
25LS2548	Chip Select Address Decoder
Am2960	Cascadable 16-Bit Error Detection and Correction
Am2961/2962	4-Bit Error Correction Multiple Bus Buffers
Am2964B	Dynamic Memory Controller
Am2965/2966	Octal Dynamic Memory Driver with Three-State Output
Am29806	6-Bit Chip Select Decoder
Am29809	9-Bit Equal-to Comparator
Am29818	8-Bit Diagnostics Register
Am29821	10-Bit Noninverting Register
Am29822	10-Bit Noninverting Register
Am29823	9-Bit Noninverting Register
Am29824	9-Bit Noninverting Register
Am29825	8-Bit Noninverting Register
Am29826	8-Bit Noninverting Register
Am29827	10-Bit Bus Driver

*Refer to the Programmable Array Logic Databook for complete product information.
**Refer to the Bipolar Microprocessor Logic and Interface Databook for complete product information.

Refer to page 7-1 for Essential Information on Military Devices

Part Number	Description
Am29828	10-Bit Bus Driver
Am29833	9-Bit Bidirectional Bus Transceiver
Am29834	9-Bit Bidirectional Bus Transceiver
Am29841	10-Bit Noninverting Latch
Am29842	10-Bit Inverting Latch
Am29843	9-Bit Noninverting Latch
Am29844	9-Bit Inverting Latch
Am29845	8-Bit Noninverting Latch
Am29846	8-Bit Inverting Latch
Am29853	9-Bit Bidirectional Bus Transceiver
Am29854	9-Bit Bidirectional Bus Transceiver
Am29861	10-Bit Bidirectional Bus Transceiver
Am29862	10-Bit Bidirectional Bus Transceiver
Am29863	9-Bit Bidirectional Bus Transceiver
Am29864	9-Bit Bidirectional Bus Transceiver
Am8120	Octal D-Type Flip-Flop
Am8127	Z8000 Clock Generator
Am8163	Timing, Refresh and EDC Controller
Am8167	Timing, Refresh and EDC Controller
Am8212	8-Bit I/O Port
Am8216	4-Bit Parallel Bidirectional Bus Driver, Noninverting
Am8224	Clock Generator, 8080A Compatible
Am8226	4-Bit Parallel Bidirectional Inverting Bus Driver
Am8228	8080A System Controller and Bus Driver
Am8238	8080A System Controller and Bus Driver with Extended IOW and MEMW
Am8286/87	Octal Bus Transceivers
Am8284A	Clock Generator & Driver for 8086, 8088 Processors
Am8288	Bus Controller

Refer to page 7-1 for Essential Information on Military Devices

Advanced General Purpose Peripherals Index

Am7970

Compression Expansion Processor (CEP)

DISTINCTIVE CHARACTERISTICS

- Compression/Expansion of digital two-tone image data using run-length and relative address coding.
  - Full-duplex capability for simultaneous independent compression and expansion.
  - High-performance 2 to 8 Mbps throughput with a 5-MHz clock.
- Compatible with CCITT recommendations T.4 and T.6 for Group 3 and Group 4 facsimile apparatus.
  - One-Dimensional, Modified Huffman Coding with optional Wraparound Mode.
  - Two-Dimensional, Modified READ (MR and MMR) coding with programmable K-Parameter.
- CPU Bus and optional local Document Store Bus with on-chip, dual-bus DMA controller.
  - 16-Mbyte physical addressing range on each bus.
- Transparent Mode transfer of unmodified data.
- Programmable paper width up to 16K picture elements and programmable top, left, and right margins.
- Optional Express Mode during compression and Granularity Mode during expansion.

GENERAL DESCRIPTION

The Am7970 Compression/Expansion Processor (CEP) is a high-performance peripheral which compresses and expands two-tone bit image data in accordance with internationally-accepted CCITT recommendations. These fully image-preserving compression protocols allow highly efficient storage and transmission of two-tone pictures and documents.

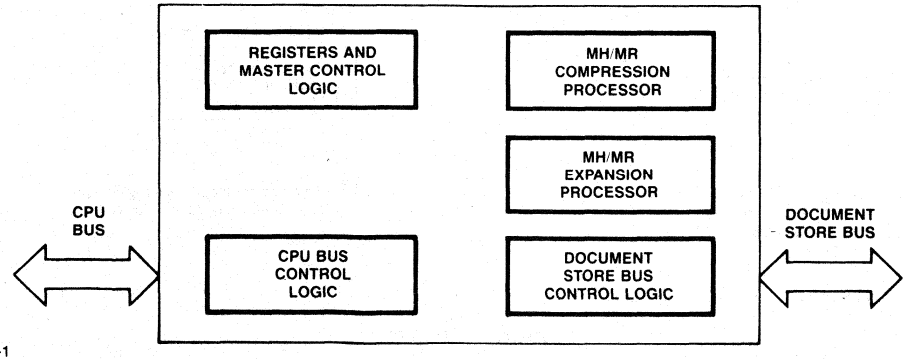
The CEP performs Modified Huffman One-Dimensional Coding or Modified READ Two-Dimensional Coding. This is compatible with CCITT recommendations T.4 and T.6 for Group 3 and Group 4 digital facsimile apparatus. Typical compression of the eight CCITT test documents is 5x to 50x. (The compression ratio is very dependent on the document, the compression mode, and the image resolution.)

The Am7970 has a standard Am8088-like microprocessor bus interface for easy implementation. CEP operation is set by programming internal control registers. CEP status is available through polled registers; exception conditions may be signalled using an external interrupt. The 38 on-chip registers allow very easy and highly flexible system implementation. After initialization, the CEP processes data with minimal intervention by the host processor.

The Compressor and Expander, which operate in full-duplex, can be independently programmed for One-Dimensional encoding/decoding, Two-Dimensional encoding/decoding, or Transparent data transfer.

In Two-Dimensional operation, the programmable K-Parameter defines the number of lines to be encoded in each Two-Dimensional coding sequence. For error-less systems (Group 4), "K=infinity" allows maximum compression.

Accelerated image processing is supported with a Compressor Express Mode, which compresses only every "Nth" line (N=1 to 255) and an Expander Granularity Mode which repeats each line "N" times (N=1 to 7).



05777B-1

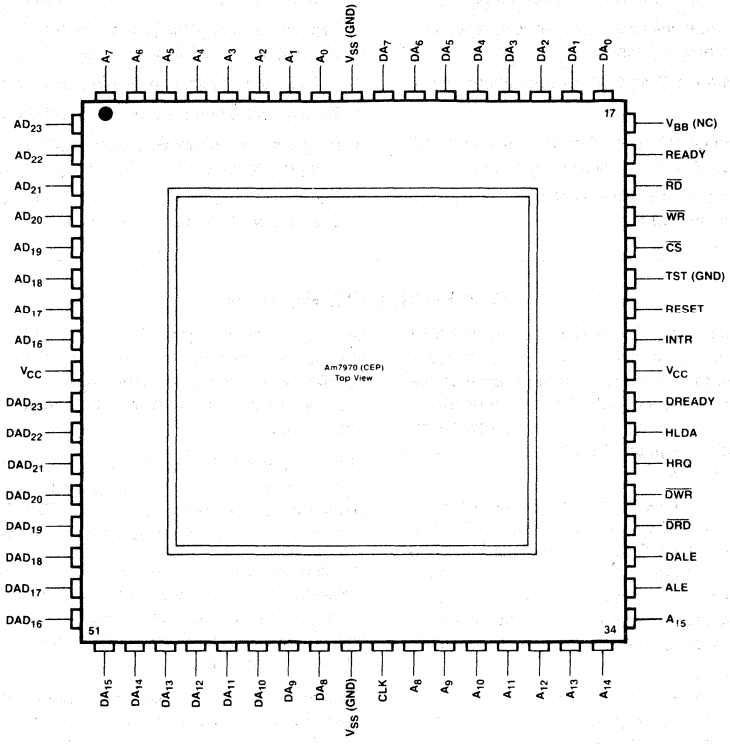
05777B

Refer to page 7-1 for Essential Information on Military Devices

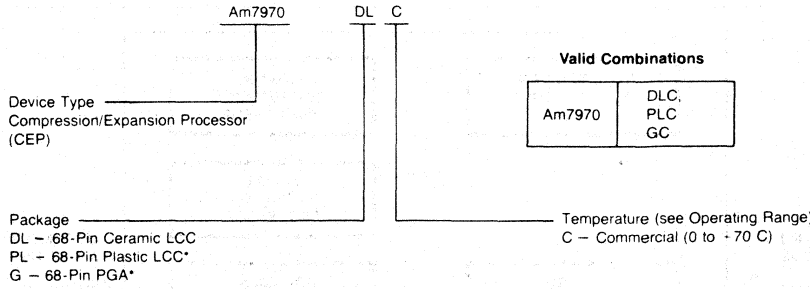
Document format controls include line length and margins. Line lengths or document widths of up to 16K picture elements may be selected. Programmable top, left, and right margins specify "white space" around image data, supporting normal margin requirements, and also "windowing," which is the overlaying of multiple image blocks or image blocks and character blocks.

The Am7970 CEP includes a secondary, local Document Store bus for optional use in conjunction with the CPU bus. The local storage buffer is highly desirable within many system architectures to optimize CPU bus performance. The CEP can linearly address up to 16 Mbytes of memory on each bus, for a total of 32 Mbytes. Starting address, buffer length, and current address for raw and processed data are stored within internal registers independently for both the Compressor and the Expander.

The Am7970 is packaged in a 68-pin LCC or Pin Grid Array and uses a single +5 V power supply.



Am7970 (CEP) Pin Out for LCC Package

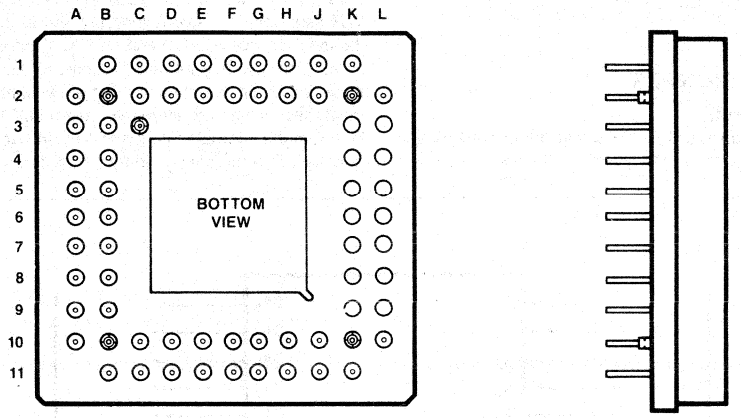


Valid Combinations

Am7970	DLC, PLC GC

*Package to be announced. Contact factory

ORDERING INFORMATION



Am7970 (CEP) Pin Out for PGA Package

Pin Number	Function
A_2	AD_{23}
A_3	AD_{21}
A_4	AD_{19}
A_5	AD_{17}
A_6	V_{CC}
A_7	DAD_{22}
A_8	DAD_{20}
A_9	DAD_{18}
A_{10}	DAD_{16}
B_1	A_7
B_2	A_6
B_3	AD_{22}
B_4	AD_{20}
B_5	AD_{18}
B_6	AD_{16}
B_7	DAD_{23}
B_8	DAD_{21}
B_9	DAD_{19}
B_{10}	DAD_{17}
B_{11}	DA_{15}
C_1	A_5
C_2	A_4
C_{10}	DA_{14}
C_{11}	DA_{13}
D_1	A_3
D_2	A_2
D_{10}	DA_{12}
D_{11}	DA_{11}
E_1	A_1
E_2	A_0
E_{10}	DA_{10}
E_{11}	DA_9
F_1	V_{SS}
F_2	DA_7
F_{10}	DA_8
F_{11}	V_{SS}

Pin Number	Function
G_1	DA_6
G_2	DA_5
G_{10}	CLK
G_{11}	A_8
H_1	DA_4
H_2	DA_3
H_{10}	A_9
H_{11}	A_{10}
J_1	DA_2
J_2	DA_1
J_{10}	A_{11}
J_{11}	A_{12}
K_1	DA_0
K_2	READY
K_3	WR
K_4	TST (NC)
K_5	INTR
K_6	DREADY
K_7	HRQ
K_8	DRD
K_9	ALE
K_{10}	A_{13}
K_{11}	A_{14}
L_2	V_{BB} (NC)
L_3	RD
L_4	CS
L_5	RESET
L_6	V_{CC}
L_7	HLDA
L_8	DWR
L_9	DALE
L_{10}	A_{15}

INTERFACE SIGNAL DESCRIPTION

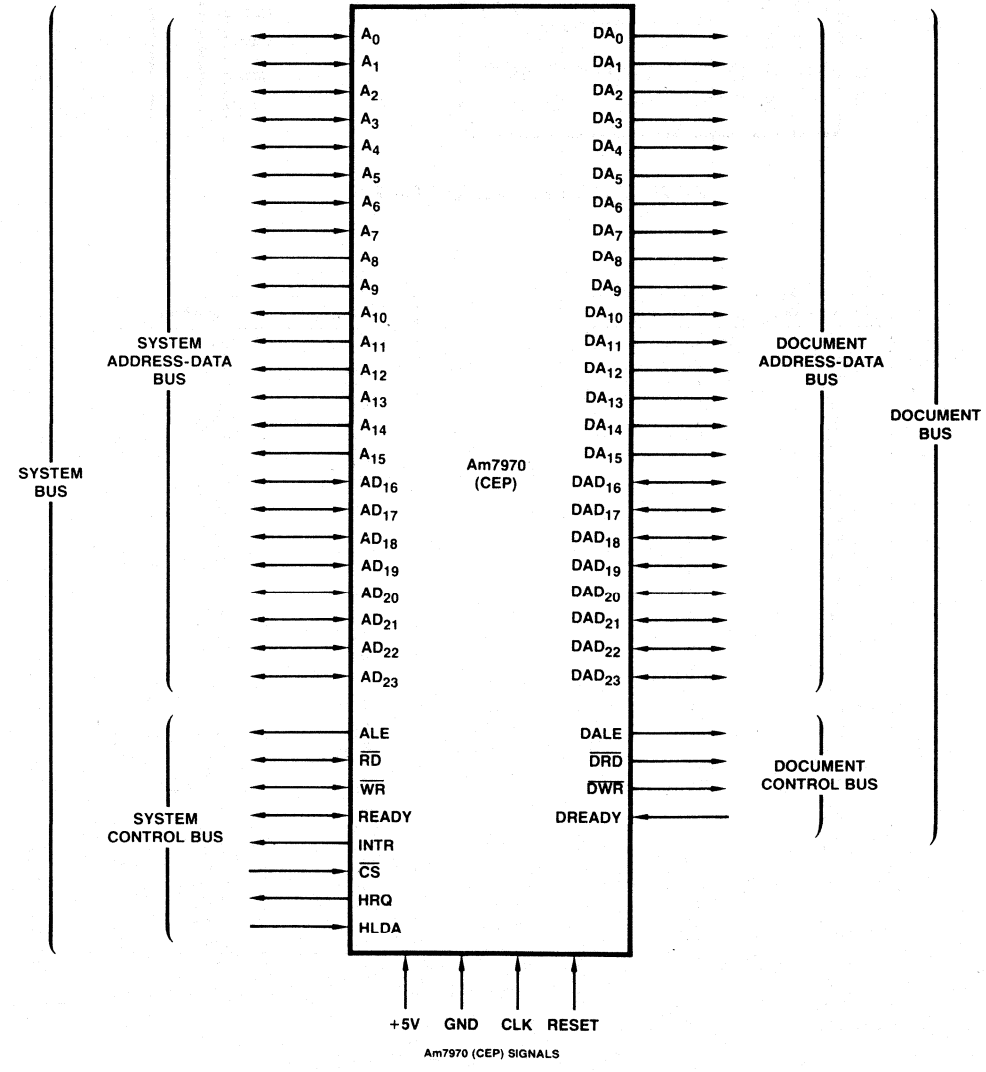
All inputs to the CEP are directly TTL-compatible.

V_{CC}: +5 Volts
V_{SS}: Ground

SYSTEM BUS SIGNALS

CLOCK (Clock, Input)

The Clock signal controls the CEP's internal operations and determines the rates of its data transfers. It is usually derived from a master system clock or the associated CPU clock. The Clock input accepts a TTL voltage level with a maximum undershoot of −0.5 V. The input signals $\overline{CS}$, HLDA, $\overline{RD}$, and $\overline{WR}$ can make transitions independent of the CEP clock. On the input signals READY and DREADY, transitions must meet set-up and hold requirements relative to the CEP clock, since these inputs do not contain internal synchronizers. Failure to meet these timing requirements may result in incorrect operation from the internal state machine with unpredictable consequences. See the timing diagrams for details.



05777B-2

Figure 2. Am7970 (CEP) Signals

05777B

Refer to page 7-1 for Essential Information on Military Devices

$\overline{RD}$(Read, Input/Output, Three-state)
$\overline{RD}$ is a bidirectional, active-Low, three-state signal. A Low indicates that the AD_{16}–AD_{23} bus is being used for a Read Data Transfer. When the CEP is not in control of the system bus and the external system is transferring information from the CEP, $\overline{RD}$ is a timing input used by the CEP to move registered data onto the AD_{16}–AD_{23} bus. The $\overline{RD}$ cycle from the system should be completed only after the CEP's READY output has returned High. After $\overline{RD}$ returns to its High state, AD_{16}–AD_{23} will float. $\overline{RD}$ is an output when the CEP is Bus Master (HRQ and HLDA are both High). The CEP will assert $\overline{RD}$ Low when data from Main Memory is required. The CEP strobes this data into its internal buffers near the Low-to-High transition of $\overline{RD}$. During the first clock period of a $\overline{RD}$ cycle, AD_{16}–AD_{23} contain the upper eight bits of the Main Memory address. Address Lines A_0–A_{15} contain the lower 16 bits of the Main Memory address. These Lower Address Lines are non-multiplexed and remain active throughout the cycle. The Upper Address Lines are time-multiplexed with data. The address is defined as valid at the falling edge of ALE. Data must be valid with respect to $\overline{RD}$. See timing diagrams for details. $\overline{RD}$ and $\overline{WR}$ may both be Low when the CEP is a Bus Slave and $\overline{CS}$ is High. However, the CEP will never drive both $\overline{RD}$ and $\overline{WR}$ Low together.

$\overline{WR}$(Write, Input/Output, Three-state)
$\overline{WR}$ is a bidirectional, active-Low, three-state signal. A Low indicates that the AD_{16}–AD_{23} bus is being used for a Write Data Transfer. When the CEP is not in control of the system bus and the external system is transferring information to the CEP, $\overline{WR}$ is a timing input used by the CEP to move data from the AD_{16}–AD_{23} bus into its internal registers. The data will be loaded into the specified register before the CEP's READY output is driven High. This $\overline{WR}$ cycle from the system should be completed only after the CEP's READY output has returned High. After $\overline{WR}$ returns to its High state, AD_{16}–AD_{23} will float to three-state. $\overline{WR}$ is an output when the CEP is Bus Master (HRQ and HLDA are both High). The CEP will assert $\overline{WR}$ Low when data is to be written into Main Memory. The CEP drives this data onto its AD_{16}–AD_{23} lines near the High-to-Low transition of WR. During the first clock period of a $\overline{WR}$ cycle, AD_{16}–AD_{23} contain the upper eight bits of the Main Memory address. Address Lines A_0–A_{15} contain the lower 16 bits of the Main Memory address. These Lower Address Lines are non-multiplexed and remain active throughout the cycle. The Upper Address Lines are time-multiplexed with data. The address is defined as valid at the falling edge of ALE. The data is defined as valid with respect to $\overline{WR}$. See timing diagrams for details.

$\overline{CS}$(Chip Select, Input)
$\overline{CS}$ is an asynchronous, active-Low input. A CPU or other external device uses $\overline{CS}$ to activate the CEP for reading from or writing to its internal registers. Once asserted Low, this input can remain Low until all register accesses have been completed. Once $\overline{CS}$ is negated High, it may not be re-asserted Low again for at least 100 ns. There are no timing requirements between the $\overline{CS}$ input and the CEP clock; the $\overline{CS}$ input timing requirements are only defined relative to the $\overline{RD}$ and $\overline{WR}$ signals. $\overline{CS}$ is ignored when the CEP is in control of the system bus.

ALE(Address Latch Enable, Output)
This active-High signal is provided by the CEP to latch the address signals AD_{16}–AD_{23} into an address latch. This pin is never floated. ALE is asserted High during address time when the CEP is Bus Master; otherwise it is Low. Address is defined as valid prior to the High-to-Low transition of ALE.

HRQ(Hold Request, Output)
Hold Request is an active-High signal used by the CEP to obtain control of the bus from the system CPU or arbiter. Hold Request lines from multiple devices may be connected to a priority encoder. If the HLDA input is High after the HRQ output has been asserted High, HRQ will remain High until the CEP has completed a single transfer. After this High-to-Low Transition, the HRQ signal will remain Low for a minimum of 2 clocks.

HLDA(Hold Acknowledge, Input)
HLDA is an asynchronous, active-High input indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since HLDA is internally synchronized by the CEP before being used, transitions on HLDA do not have to be synchronous with the CEP clock. The HLDA input can be connected to the HLDA output from the CPU (8086-type), to the output of a priority decoder, or to the output of some other arbitration device. The HLDA input normally remains High until the CEP drives the HRQ output Low.

READY(Ready, Input/Output, Three-state)
READY is a synchronous, active-High, three-state, bidirectional signal. READY is used as an input signal when the CEP is Bus Master. Slow memories may use READY to extend $\overline{RD}$ or $\overline{WR}$ cycles. This is accomplished by negating READY Low at the appropriate times and thus inserting Wait States until READY is returned High. READY must be High before Main Memory data can be accessed by the CEP. Care must be taken, however, to assure that this signal is synchronized to the CEP clock and thus meets its set-up and hold requirements. Failure to do so can result in unpredictable operation. READY is used as an output signal when the CEP is Bus Slave. After $\overline{CS}$ has been asserted Low by the CPU, READY will be driven Low by the CEP until it is able to provide or accept data for the current transaction. Once ready, the CEP will assert READY High at which time the CPU should complete the current read or write cycle by negating $\overline{RD}$ or $\overline{WR}$. See timing diagrams for details.

INTR(Interrupt Request, Output)
Interrupt Request is an active-High output used to interrupt the CPU. It is driven High whenever an exception or termination condition exists in either the Compressor (if the Compressor Interrupt Enable bit is set) or Expander (if the Expander Interrupt Enable bit is set). The INTR line will be reset to Low when the CPU reads the CEP Master Status Register or when the CEP is reset.

RESET(Reset, Input)
RESET is an asynchronous, active-High input which initializes the Am7970 to an idle state. This input must be driven High for at least four clock cycles.

A_0–A_7 /A_8–A_{15} (Lower Address Bus, A_0–A_7 are Input/Output, A_8–A_{15} are Three-State Outputs)
The Lower Address Bus is a non-multiplexed, bidirectional (on the least significant eight address lines, A_0–A_7 only), active-High, three-state bus used in addressing all system bus I/O and memory transactions. A_0 is the least significant bit position and A_{15} is the most significant bit position.

When the CEP is not in control of the system bus (HLDA Low) and the $\overline{CS}$ input is asserted Low, A_0–A_7 are used as input address lines to access the CEP's internal registers. (The CEP's internal registers have been assigned even addresses.) During this time, the address lines A_8–A_{15} are ignored by the CEP. The input addresses on A_0–A_7 must be valid one set-up time before the $\overline{CS}$ input is driven Low and must remain valid throughout the register transaction. See timing diagrams for details.

When the CEP is in control of the main bus (HRQ and HLDA are High), DMA transactions with the Main Memory will occur. The presence of valid address on A_0–A_{15} is defined by the falling edge of ALE. During this Master Mode, A_0–A_{15} are used as non-multiplexed output address lines until HRQ is negated Low. After the High-to-Low transition of HRQ, the A_0–A_{15} lines will float to a three-state condition.

AD_{16}–AD_{23} (Address-Data Bus, Input/Output, Three-state)
The Address-Data Bus is a time-multiplexed (in Master Mode only), bidirectional, active-High, three-state bus used for all system bus I/O and memory transactions. When referring to the data cycle on this bus, AD_{16} is the least significant data bit position and AD_{23} is the most significant. The presence of a valid address during Bus Master operations is defined by the falling edge of ALE and the valid data is defined by the $\overline{WR}$ and $\overline{RD}$ signals; otherwise these lines are floating. The $\overline{RD}$ and $\overline{WR}$ outputs will return to their inactive-High levels only after the READY input has been sampled High. While the CEP $\overline{RD}$ output is Low and when the READY input is High, AD_{16}–AD_{23} must be provided with valid input data from the system. When the CEP $\overline{WR}$ output is asserted Low, AD_{16}–AD_{23} will be driven by the CEP with valid output data. The READY input must then return High to acknowledge receipt of the valid data and to allow the completion of the $\overline{WR}$ cycle. When the CEP is acting as a Bus Slave (HRQ and HLDA Low) and the $\overline{CS}$ input is driven Low, AD_{16}–AD_{23} are used strictly as data lines D_0–D_7. They behave as input data lines when $\overline{WR}$ is asserted Low and as output data lines when $\overline{RD}$ is asserted Low. At all other times they are floated to three-state. During Slave Mode operation, when $\overline{CS}$ is driven Low, the READY output will be asserted Low to signal that the CEP is not ready to complete the transaction. When the CEP is ready, the READY output will be driven High.

DOCUMENT STORE BUS SIGNALS

$\overline{DRD}$(Document Store Read, Output, Three-state)
$\overline{DRD}$ is an active-Low, three-state signal. A Low on this signal indicates that the DAD_{16}–DAD_{23} bus is being used for a Read Data Transfer. When the CEP does not have a Source or Destination Buffer located on the Document Store bus, this pin is floated to three-state. $\overline{DRD}$ is an output when the CEP is in control of the Document Bus. The CEP asserts $\overline{DRD}$ Low when data from Document Memory is required. The CEP strobes this data into its internal buffers near the Low-to-High transition of $\overline{DRD}$. During the first clock period of a $\overline{DRD}$ cycle, DAD_{16}–DAD_{23} contain the upper eight bits of the Document Memory address. Address Lines DA_0–DA_{15} contain the lower 16 bits of the Document Memory address. These Lower Address Lines are non-multiplexed and so remain active throughout the cycle. The Upper Address Lines are time-multiplexed between address and data. The address is defined as valid at the falling edge of DALE. The data is required to be valid with respect to $\overline{RD}$. See timing diagrams for details.

$\overline{DWR}$(Document Store Write, Output, Three-state)
$\overline{DWR}$ is an active-Low, three-state signal. A Low on this pin indicates that the DAD_{16}–DAD_{23} bus is being used for a Document Bus Write Data Transfer. When the CEP does not have a Source or Destination Buffer located on the Document Store bus, this pin is floated to three-state. $\overline{DWR}$ is an output when the CEP is Bus Master. The CEP will assert $\overline{DWR}$ Low when data is to be written into Document Memory. The CEP drives this data onto its DAD_{16}–DAD_{23} lines near the High-to-Low transition of $\overline{DWR}$. During the first clock period of a $\overline{DWR}$ cycle, DAD_{16}–DAD_{23} contain the upper eight bits of the Document Memory address. Document Address Lines DA_0–DA_{15} contain the lower 16 bits of the Document Memory address. These Lower Address Lines are non-multiplexed and remain active throughout the cycle. The upper eight Address Lines are time-multiplexed between address and data. The document address is defined as valid at the falling edge of DALE. The data is defined as valid with respect to $\overline{DWR}$. See timing diagrams for details.

DALE(Document Store Address Latch Enable, Output, Three-state)
This active-High signal is provided by the CEP to latch the Document Store address signals DAD_{16}–DAD_{23} into an address latch. When the CEP does not have a Source or Destination Buffer located on the Document Bus this pin is floated to three-state. DALE is asserted High during address time when the CEP is Bus Master; otherwise it is Low. Address is defined as valid prior to the transition of DALE.

DREADY(Ready, Input, Three-state)
DREADY is a synchronous, active-High, three-state signal. DREADY is used as an input signal when the CEP is Bus Master. Slow memories may use DREADY to extend $\overline{DRD}$ or $\overline{DWR}$ cycles. This is accomplished by negating DREADY Low at the appropriate times and thus inserting Wait States until DREADY is returned High. DREADY must be High before Document Memory data can be accessed by the CEP. Care must be taken, however, to ensure that this signal is synchronized to the CEP clock and thus meets its set-up and hold requirements. Failure to do so can result in unpredictable operation. DREADY may also be useful, in conjunction with transceiver and control logic, in hardware arbitration schemes on the Document Bus.

DA_0–DA_{15} (Document Store Lower Address Bus, Output, Three-state)
The Document Store Lower Address bus is a non-multiplexed, active-High, three-state bus used in addressing all local Document Memory transactions. DA_0 is the least significant bit position and DA_{15} is the most significant bit position.

When the CEP is in control of the Document Store Bus, the presence of valid address on DA_0–DA_{15} is defined by the falling edge of DALE. During this Master Mode, DA_0–DA_{15} will be used as non-multiplexed output address lines whenever the Compressor or the Expander is using the Document Store as a Source or Destination Buffer for the current transaction, otherwise this bus is floated to three-state.

DA_{16}–DA_{23} (Document Store Upper Address-Data Bus, Input/Output, Three-state)

The Document Store Upper Address-Data bus is a time-multiplexed, bidirectional, active-High, three-state bus used for all local Document Memory transactions. When referring to the data cycle on this bus, DAD_{16} is the least significant data bit position and DAD_{23} is the most significant. The presence of a valid address during Bus Master operations is defined by the falling edge of DALE and the valid data is defined by the $\overline{DWR}$ and $\overline{DRD}$ signals; otherwise these lines are floating. The $\overline{DRD}$ and $\overline{DWR}$ outputs will return to their inactive-High levels only after the DREADY input has been sampled High. While the CEP $\overline{DRD}$ output is Low and when the DREADY input is High, DAD_{16}–DAD_{23} must be provided with valid input data from the system. When the CEP $\overline{DWR}$ output is asserted Low, DAD_{16}–DAD_{23} will be driven by the CEP with valid output data. The DREADY input must then return High to acknowledge receipt of the valid data and to allow the completion of the $\overline{DWR}$ cycle.

REGISTER DESCRIPTION

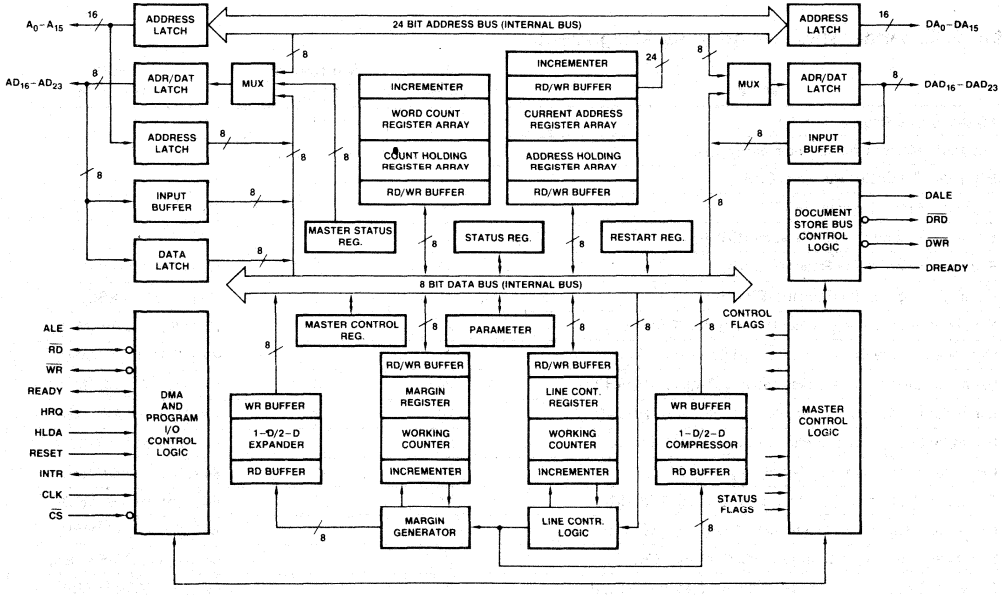
The diagram shown in Figure 3 illustrates the Am7970 CEP's internal registers. Tables 1 and 2 list each register along with its size and access address. Table 3 cross-references these registers by address.

The state of the Status Register, Master Status Register, and GO bits after initialization by the RESET input are "0." The status of other bits after initialization by the RESET input is not specified.

MASTER CONTROL REGISTER (CMCR/EMCR)

The 8-bit Master Control Register, shown in Figure 4, is used to specify the desired mode of operation (One- or Two-Dimensional), the location of the Source and Destination Buffers (Main Memory or Document Store), the Interrupt Enable, the operation controls (Reset, Single-Line or Multi-Line), and the initiation of processing (Start or Stop Processing). The function of each of the Master Control bits is described in the following paragraphs.

The Least Significant Bit (Bit 0) is called the GO bit. The system program initiates compression or expansion operation by setting this bit to "1." Once set, the appropriate Status Register (Compressor Status Register or Expander Status Register) indicates that the selected processor is busy. Upon completion of the selected operation, this bit is reset to "0" automatically. Any attempt to load the Compressor Master Control Register when the Compressor is busy, or the Expander Master Control Register when the Expander is busy, terminates the selected processor with the appropriate error bits set in the Compressor Status Register or Expander Status Register respectively.



05777B-3

Figure 3. Am7970 (CEP) Block Diagram

05777B

Refer to page 7-1 for Essential Information on Military Devices

Table 1. Compressor Register Address Assignments

Name	Size (Bits)	Number of Bytes	Port Address(es)
Master Status Register (MSR)*	8	1	FE
Compressor Master Control Register (CMCR)	8	1	76
Compressor Parameter Register (CPR)	8	1	74
Compressor Status Register (CSR)	8	1	78
Compressor Express Register (CER)	8	1	68
Compressor Restart Control Register (CRCR)	8	1	48
Time Fill Register (TFLR)	8	1	44
Compressor Wraparound Register (CWR)	16	2	50 (LSB)/52 (MSB)
Left Margin Register (LMGR)	16	2	40 (LSB)/42 (MSB)
Right Margin Register (RMGR)	16	2	60 (LSB)/62 (MSB)
Top Margin Register (TMGR)	16	2	30 (LSB)/32 (MSB)
Compressor Page Width Register (CPWR)	16	2	70 (LSB)/72 (MSB)
Compressor Source Address Holding Register (CSAHR)	24	3	3A (LSB)/3C/3E (MSB)
Compressor Source Current Address Register (CSCAR)	24	3	0A (LSB)/0C/0E (MSB)
Compressor Destination Address Holding Register (CDAHR)	24	3	4A (LSB)/4C/4E (MSB)
Compressor Destination Current Address Register (CDCAR)	24	3	2A (LSB)/2C/2E (MSB)
Compressor Source Count Holding Register (CSCHR)	24	3	14 (LSB)/16/18 (MSB)
Compressor Source Working Count Register (CSWCR)	24	3	04 (LSB)/06/08 (MSB)
Compressor Destination Count Holding Register (CDCHR)	24	3	34 (LSB)/36/38 (MSB)
Compressor Destination Working Count Register (CDWCR)	24	3	24 (LSB)/26/28 (MSB)
Compressor Source Line Start Address Register (CSLSR)	24	3	5A (LSB)/5C/5E (MSB)
Compressor Destination Line Start Address Register (CDLSR)	24	3	6A (LSB)/6C/6E (MSB)

*This register is common to both the compressor and the expander.

Table 2. Expander Register Address Assignments

Name	Size (Bits)	Number of Bytes	Port Address(es)
Master Status Register (MSR)*	8	1	FE
Expander Master Control Register (EMCR)	8	1	F6
Expander Parameter Register (EPR)	8	1	F4
Expander Status Register (ESR)	8	1	F8
Expander Restart Control Register (ERCR)	8	1	C8
Expander Wraparound Register (EWR)	16	2	D0 (LSB)/D2 (MSB)
Expander Page Width Register (EPWR)	16	2	F0 (LSB)/F2 (MSB)
Expander Source Address Holding Register (ESAHR)	24	3	BA (LSB)/BC/BE (MSB)
Expander Source Current Address Register (ESCAR)	24	3	8A (LSB)/8C/8E (MSB)
Expander Destination Address Holding Register (EDAHR)	24	3	CA (LSB)/CC/CE (MSB)
Expander Destination Current Address Register (EDCAR)	24	3	AA (LSB)/AC/AE (MSB)
Expander Source Count Holding Register (ESCHR)	24	3	94 (LSB)/96/98 (MSB)
Expander Source Working Count Register (ESWCR)	24	3	84 (LSB)/86/88 (MSB)
Expander Destination Count Holding Register (EDCHR)	24	3	B4 (LSB)/B6/B8 (MSB)
Expander Destination Working Count Register (EDWCR)	24	3	A4 (LSB)/A6/A8 (MSB)
Expander Source Line Start Address Register (ESLSR)	24	3	DA (LSB)/DC/DE (MSB)
Expander Destination Line Start Address Register (EDLSR)	24	3	EA (LSB)/EC/EE (MSB)

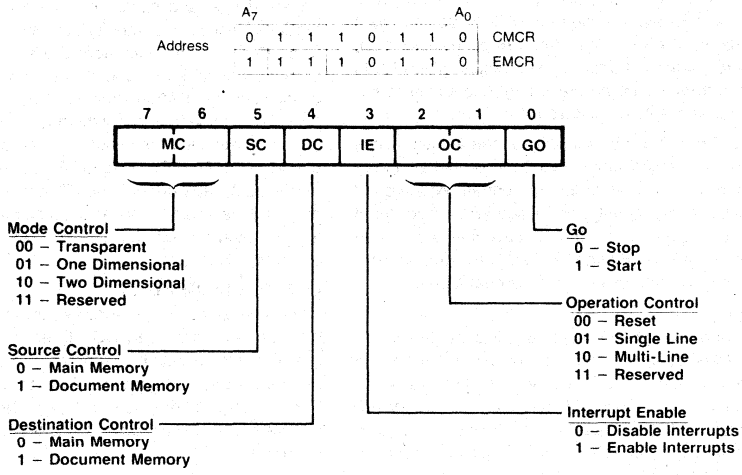
Note: All register addresses are even, the bytes in a register are, therefore, not addressed with contiguous addresses.

*This register is common to both the compressor and expander.

Table 3. Am7970 Internal Registers

	0	2	4	6	8	A	C	E
0			CSWCR (L)	CSWCR (M)	CSWCR (H)	CSCAR (L)	CSCAR (M)	CSCAR (H)
1			CSCHR (L)	CSCHR (M)	CSCHR (H)			
2			CDWCR (L)	CDWCR (M)	CDWCR (H)	CDCAR (L)	CDCAR (M)	CDCAR (H)
3	TMGR (L)	TMGR (H)	CDCHR (L)	CDCHR (M)	CDCHR (H)	CSAHR (L)	CSAHR (M)	CSAHR (H)
4	LMGR (L)	LMGR (H)	TFLR		CRCR	CDAHR (L)	CDAHR (M)	CDAHR (H)
5	CWR (L)	CWR (H)				CSLSR (L)	CSLSR (M)	CSLSR (H)
6	RMGR (L)	RMGR (H)			CER	CDLSR (L)	CDLSR (M)	CDLSR (H)
7	CPWR (L)	CPWR (H)	CPR	CCR	CSR			
8			ESWCR (L)	ESWCR (M)	ESWCR (H)	ESCAR (L)	ESCAR (M)	ESCAR (H)
9			ESCHR (L)	ESCHR (M)	ESCHR (H)			
A			EDWCR (L)	EDWCR (M)	EDWCR (H)	EDCAR (L)	EDCAR (M)	EDCAR (H)
B			EDCHR (L)	EDCHR (M)	EDCHR (H)	ESAHR (L)	ESAHR (M)	ESAHR (H)
C					ERCR	EDAHR (L)	EDAHR (M)	EDAHR (H)
D	EWR (L)	EWR (H)				ESLSR (L)	ESLSR (M)	ESLSR (H)
E						EDLSR (L)	EDLSR (M)	EDLSR (H)
F	EPWR (L)	EPWR (H)	EPR	ECR	ESR			MSR

(L): Low Byte
(M): Middle Byte
(H): High Byte



05777B-4

Figure 4. Compressor/Expander Master Control Register (CMCR/EMCR)

05777B

Refer to page 7-1 for Essential Information on Military Devices

Bits 1 and 2 of the Compressor Master Control Register or Expander Master Control Register are collectively referred to as the Operation Control (OC) bits. Three out of four possible operations are initiated in this field. The OC bits specify the desired operation according to the following table:

Bit 2 OC1	Bit 1 OC0	Operation
0	0	RESET
0	1	SINGLE-LINE
1	0	MULTI-LINE
1	1	RESERVED

RESET (00)

This operation selectively sets the Compressor or Expander to the same state generated by a hardware RESET. This flushes the input queue and clears the contents of the internal working registers, process control flags and appropriate status register. It does not clear the user-programmable control registers, the Compressor Master Control Register, Expander Master Control Register, Compressor Parameter Register, and Expander Parameter Register.

Following a Reset operation, the selected processor is in an idle state. If resumable operation of the current processing activity is necessary, the system should not issue a Reset operation since the clearing of the various registers invalidates any subsequent Restart operation. On the other hand, the system program *must* issue a Reset operation before starting a new sequence of events when no resumable operation from the previous processing is necessary; for example, when the CEP is going to process a new page. This software RESET between contexts is necessary to ensure that the input queue is properly flushed and that the upcoming sequence of operations is interpreted correctly.

SINGLE-LINE (01)

When Single-Line operation is initiated, one effective line of data from the Source Buffer is processed before the GO bit in the Master Control Register is cleared to "0." Such an event might be normal or in error. (See Status Register description for details of error status.) The next line or multiple lines of data can be restarted from where the preceding Single-Line operations ended if no errors have occurred.

The meaning of "effective line" depends on the mode that has been selected (One-Dimensional, Two-Dimensional, or Transparent) and thus, the contents of the appropriate Wraparound Register (One-Dimensional only) and Page Width Register. If the Wraparound Register is "0," the effective line is one scan line as defined in the Page Width Register. This "effective line" includes the reference line in Two-Dimensional Mode.

MULTI-LINE (10)

The Multi-Line operation processes data until either the appropriate Source Working Count Register (SWCR) or Destination Working Count Register (DWCR) reaches "0." At this time the selected processor terminates and the GO bit in the Master Control Register (MCR) is cleared to "0." Such an ending might be normal or in error. (See Status Register description for details of error status.)

RESERVED (11)

If the Reserved operation is initiated, the CEP terminates with an illegal operation error in the CSR or ESR, respectively (see Status Register section).

Bit 3 of the MCR is the Interrupt Enable (IE) bit. When the processor has terminated an operation, the CEP will interrupt the CPU if the IE bit is set to "1." However, if the IE bit is "0," interrupts are disabled and the Am7970 will not interrupt the CPU.

Bit 4 is the Destination Control (DC) bit and Bit 5 is the Source Control (SC) bit of the MCR register. This field specifies the residency of the Source and Destination Buffers. These control bits determine the buffer locations for the selected processor according to the following table:

Bit 5 SC	Bit 4 DC	Location
0	0	Source and Destination in Main Memory
0	1	Source in Main Memory, Destination in Document Store
1	0	Source in Document Store, Destination in Main Memory
1	1	Source and Destination in Document Store

Bits 6 and 7 of the MCR are collectively called the Mode Control Field: MC0 corresponds to Bit 6 and MC1 corresponds to Bit 7. The MC bits specify the desired mode for the selected processor according to the following table:

Bit 7 MC1	Bit 6 MC0	Mode
0	0	TRANSPARENT
0	1	ONE-DIMENSIONAL
1	0	TWO-DIMENSIONAL
1	1	RESERVED

TRANSPARENT (00)

Transparent Mode means that no data modification occurs in the selected processor; data merely passes through the processor via DMA. For example, consider the following: the Code Buffer is contained in the Document Store and all data transmissions take place from the Main Memory. Somehow, facilities must exist to transfer the required information from the Code Buffer in the Document Store to the Main Memory. By initiating Transparent Mode operation with the Document Store as the source and Main Memory as the destination, the processor can be made to perform the required information transfer in the same way as a conventional DMA controller. However, the effects of the Auto-End-of-Line (EOL) insertion feature, Margin Registers, Wraparound, Time Fill, and Express Registers must be fully considered before attempting such information moves. In Transparent Mode, the EOL code always consists of 16 bits on a byte boundary.

ONE-DIMENSIONAL (01)

One-Dimensional Mode specifies the standard Modified Huffman Code according to CCITT recommendation T.4. During this mode of operation, the processor takes into account the relevant Margin Registers, Wraparound Register, Express Register, Page Width, Auto-EOL, and Time Fill features. These registers are discussed individually under their respective headings.

TWO-DIMENSIONAL (10)

Two-Dimensional Mode specifies the standard Modified READ Code according to CCITT recommendations T.4 and T.6. During this mode of operation, the processor takes into account the Page Width, K-Parameter, Margins, Express Mode,

Auto-EOL, and Time Fill features. The Wraparound feature is not available with Two-Dimensional coding.

The Two-Dimensional Mode (MC1=1, MC0=0) utilizes both One-Dimensional and Two-Dimensional processes in conjunction with the K-Register and the K-Working-Register (not visible to the CPU). The K-Working-Register is decremented by one after each compressed line has been processed. If the K-Register contains "0" (K=infinity), the K-Working-Register continues to be decremented until the Source Buffer or the Destination Buffer overflows. Otherwise, the contents of the K-Register are used to update the K-Working-Register at "K" intervals.

The Am7970 Expander detects CCITT recommended extension codes (including exit codes). The CEP's response to detected extension codes are as follows:

1) If the three least significant bits of the detected extension code are **not** all "1s," the Extension Code Detected (ECD) bit in the Master Status Register is set to "1," the Extension (EXT) bits in the Master Status Register are loaded with the three least significant bits of the detected extension code and the Expander immediately terminates.
2) If the three least significant bits of the detected extension code are all "1s," all subsequent data is treated as uncompressed data until a CCITT recommended exit code is detected. After a CCITT recommended exit code has been detected, the CEP resumes its normal Two-Dimensional Expansion Mode of operations.

"Uncompressed data" is passed from the Source Buffer to the Destination Buffer without being expanded. "Uncompressed data" that has been written into the Destination Buffer differs from the Source Buffer data in two ways:

1) Extension code and Exit code will have been removed by the Expander.
2) Each time that the pattern 000001 occurs within the "uncompressed" data, it will be replaced by the pattern 00000.

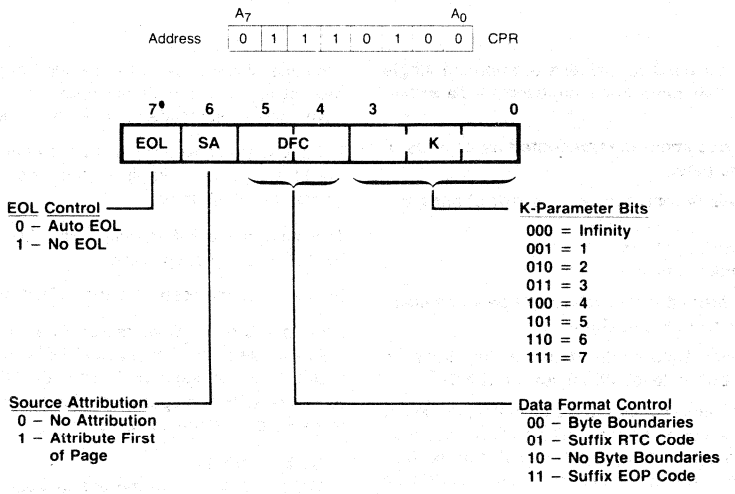
When the EOL bit within the Expander Parameter Register has been set to "0," the uncompressed data will be checked for errors as it is written into the Destination Buffer.

The Two-Dimensional Expander requires that each scan line be assigned a specific reference line. The Am7970 requires that each Two-Dimensional coded scan line use as its reference line the immediately preceding scan line.

RESERVED (11)
If the Reserved operation is initiated, the CEP terminates with an illegal operation error in the CSR or ESR, respectively (see Status Register section).

PARAMETER REGISTER (CPR/EPR)

This 8-bit register, shown in Figures 5 and 6, contains the Granularity factor (G-Parameter) for the Expander (used to repeat G scan lines) and specifies the data format for the Compressor and the beginning of the page attributes, as well as EOL control. The function of each of the Parameter bits is described in the following paragraphs.

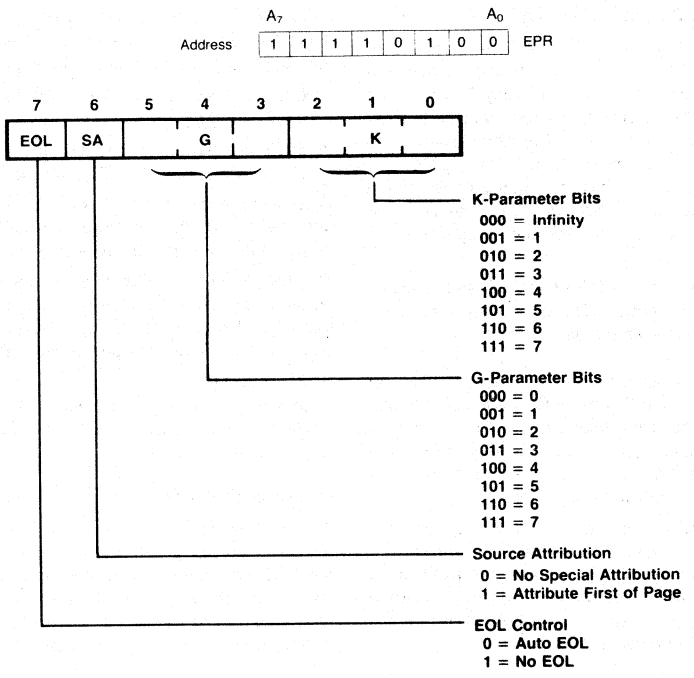


05777B-5

Figure 5. Compressor Parameter Register (CPR)

05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B-6

Figure 6. Expander Parameter Register (EPR)

The Am7970 can be used to process (compress) single page documents that have been digitized as described below.

1) **A single page document is represented by an array of black and white pels.**
2) **Each row of pels is represented by a row of ones and zeros, i.e.:**
   each white pel = 0
   each black pel = 1
3) **A bit stream is formed from each row of pels by a scanner that moves from left to right.**
4) **The first bit each byte of the resulting bit stream is considered to be the least significant bit (bit 0).**

The first bit of each byte of compressed data that is provided by the Am7970 is considred to be the least significant bit (bit 0). The first bit of each byte of expanded data that is provided by the Am7970 compressed data is expanded by the Am7970 a scanner that moves from left to right is assumed.

Compressor Parameter Bits

Bits 0, 1, and 2 of the Compressor Parameter Register contain the K-Parameter. The K-Parameter specifies the number of two-dimensional lines to be encoded or decoded (K-1) with respect to each one-dimensional line. Values from one to seven and infinity (code 000) can be selected.

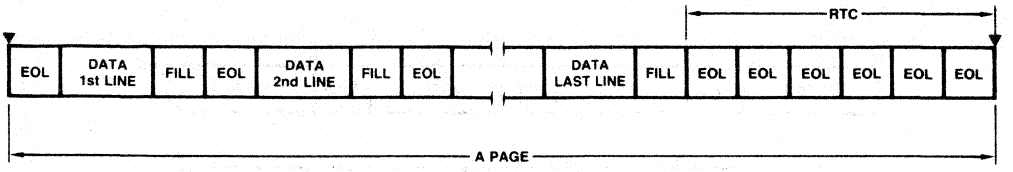
If One-Dimensional coding or Transparent Mode is specified in the Master Control Register, the selected processor logic ignores the K-Parameter.

The state of the K-Parameter bits after initialization by the RESET input is not specified.

Bit 3 in the compressor Parameter Register is reserved.

Bits 4 and 5 of the Compressor Parameter Register are collectively called the Compressed Data Format Control Field (CDF), CDF0 corresponds to Bit 4 and CDF1 corresponds to Bit 5. The CDF bits specify the desired compressed data format according to the following table:

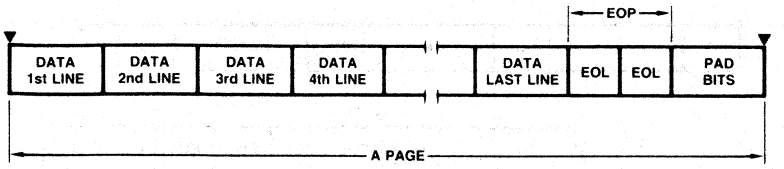
Bit 5 CDF1	Bit 4 CDF0	Compressed Data Format
0	0	Process on Byte Boundaries
0	1	Suffix Return-to-Control (RTC) Code
1	0	No Byte Boundaries
1	1	Suffix End-of-Page (EOP) Code



▼: Byte Boundaries Mark

05777B-7

Figure 7. G-3 Compressed Data Format

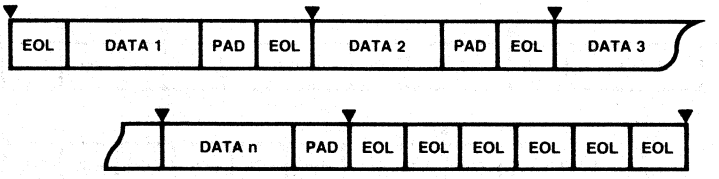


▼: Byte Boundaries Mark

05777B-8

Figure 8. G-4 Compressed Data Format

Compressed Data Format of the 1-DMode

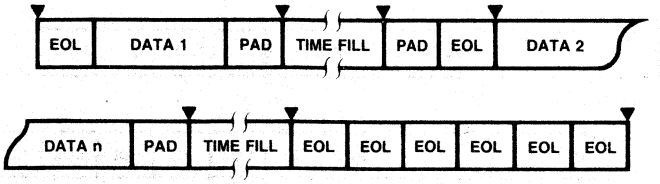


▼: Byte boundary mark

PAD: Consecutive any numbers of "0"s (if any)

05777B-9

Figure 9. Byte Boundary Conditioned with Auto EOL and No Time Fill



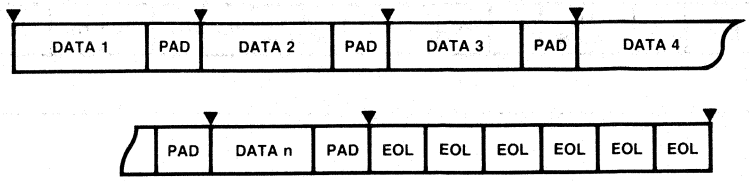
▼: Byte boundary mark

PAD: 1 to 7 "0"s (if any)

05777B-10

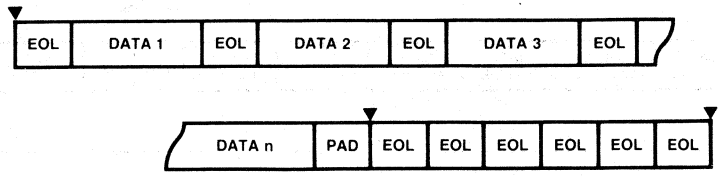
Figure 10. Byte Boundary Conditioned with Auto EOL and Time Fill

05777B



05777B-11

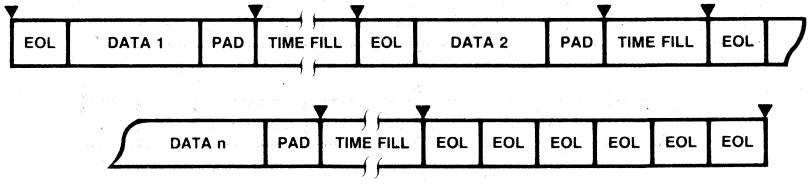
Figure 11. Byte Boundary Conditioned without EOL and Time Fill



▼: Byte boundary mark
PAD: 1 to 7 "0"s (if any)

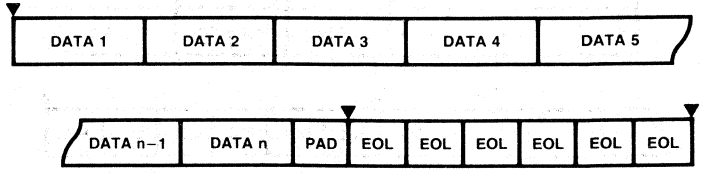
05777B-12

Figure 12. No Byte Boundary Conditioned with Auto EOL and No Time Fill



05777B-13

Figure 13. No Byte Boundary Conditioned with Auto EOL and Time Fill

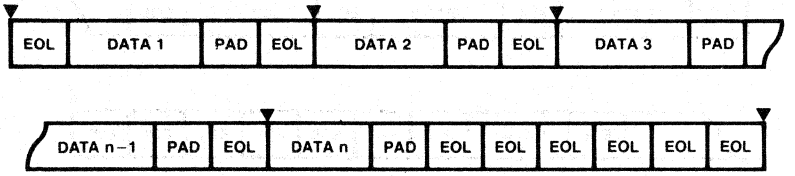


▼: Byte boundary mark
PAD: 1 to 7 "0"s (if any)

05777B-14

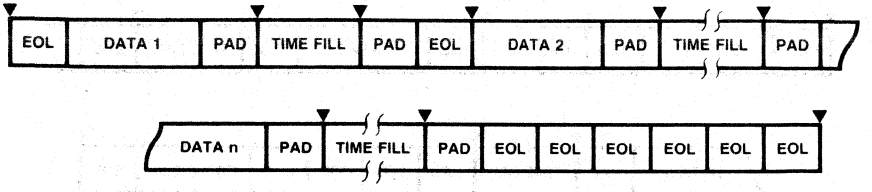
Figure 14. No Byte Boundary Conditioned without EOL and Time Fill

Compressed Data Format of the 2-D Mode



05777B-15

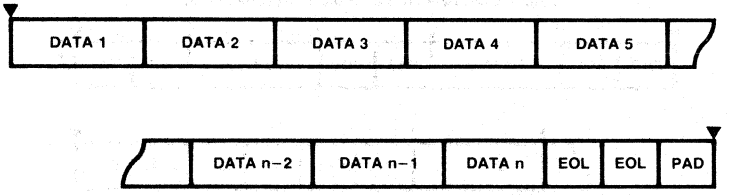
Figure 15. Byte Boundary Conditioned with Auto EOL and No Time Fill



▼: Byte boundary mark
PAD: 1 to 7 "0"s (if any)
1: Byte boundary conditioned with no EOL and no time fill is same as Figure.

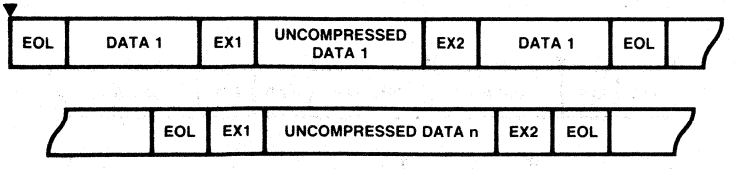
05777B-16

Figure 16. Boundary Conditioned with Auto EOL and Time Fill



05777B-17

Figure 17. No Byte Boundary Conditioned without EOL and Time Fill (G-4)



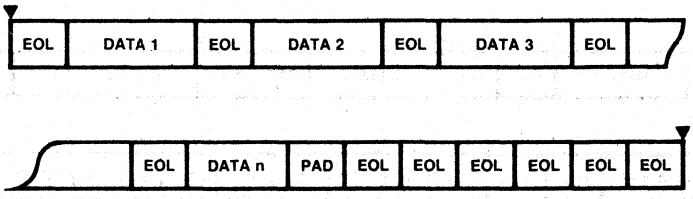
▼: Byte boundary mark
PAD: Consecutive any numbers of "0"s (if any)
EX1: Extention code (entry code)
EX2: Extension code (exit code)

05777B-18

Figure 18. Uncompressed Data Format

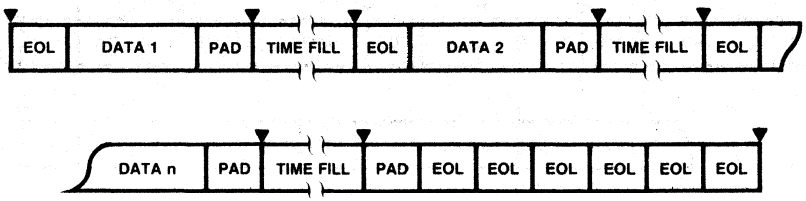
05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B-19

Figure 19. No Byte Boundary Conditioned with Auto EOL and No Time Fill



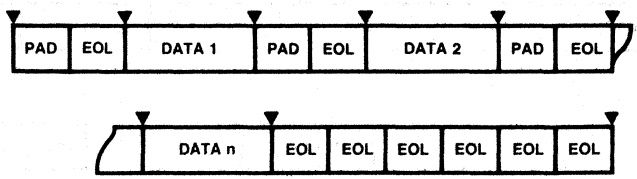
▼: Byte boundary mark

PAD: 1 to 7 "0"s (if any)

05777B-20

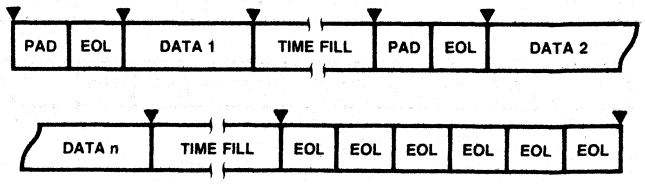
Figure 20. No Byte Boundary Conditioned with Auto and Time Fill

Compressed Data Format of the Transparent Mode



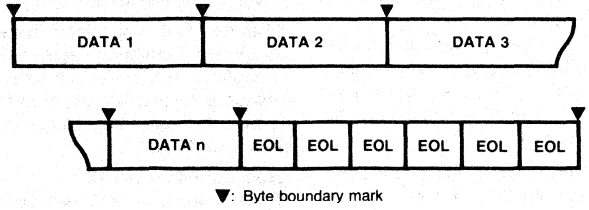
05777B-21

Figure 21. Byte Boundary Conditioned with Auto EOL and Not Time Fill



05777B-22

Figure 22. Byte Boundary Conditioned with Auto EOL and Time Fill



05777B-23

Figure 23. Byte Boundary Conditioned without EOL and Time Fill

Process on Byte Boundaries (00)
With this option, the Am7970 (CEP) is conditioned to end lines on byte boundaries. In conjunction with this option, if the Auto-EOL feature is suppressed (no EOL insertion), and if a coded line does not end on a byte boundary, the Am7970 adds enough Dummy Fill bits to end a line on a byte boundary. These Dummy Fill bits consist of one to seven consecutive "0s." If Auto-EOL is enabled, and a coded line including Time Fill (if any) plus the EOL code does not end on a byte boundary, the Am7970 adds enough Dummy Fill bits between the compressed DATA and EOL codes so that the line ends on a byte boundary. These Dummy Fill bits are also "0s" and are not distinguishable from Time Fill bits.

Caution: The Expander Byte Boundary Control Bit is located in the Restart Control Register.

Return-to-Control (RTC) Code (01)
The Suffix RTC Code specifies that the Return-to-Control code has six consecutive EOL codes as specified in the CCITT recommendation T.4. With this option, the Compressor will suffix an RTC code to the end of the terminal scan line. The Compressor is conditioned to end the RTC code on a byte boundary.

In normal Multi-Line operation, the RTC code will be suffixed following the end of the terminal line in which the Compressor Source Buffer overflow occurs, assuming line processing has been successfully completed. If a coded line including Time Fill (if any) plus the RTC code does not end on a byte boundary, the Compressor adds enough Dummy Fill bits between the compressed DATA and the RTC code to end the line on a byte boundary.

When error conditions arise during Multi-Line operation, the RTC code will not be suffixed. These error conditions are indicated by the Line Processing Incomplete (LPI) bit in the Compressor Status Register (CSR). These conditions are resumable because the rest of the coded line (including RTC code, if any) has been maintained by the Compressor internally and the suffix RTC command is suspended in the Compressor. If the GO command is issued after the Compressor Source Buffer or the Compressor Destination Buffer has been prepared, the Compressor resumes its previous process from where it left off.

For example, when a Compressor Source Buffer overflow occurs along with a Compressor Destination Buffer overflow, the processor terminates its operation promptly and the RTC code won't be suffixed. As another example, if the Compressor Source Buffer overflow has occurred with the Line Processing Incomplete and Wraparound Incomplete (if any), the Compressor terminates its operation promptly and the RTC code won't be suffixed.

In normal Single-Line operation, an RTC code will be suffixed following an end of line when that line's processing is completed. If a coded line including Time Fill (if any) and the RTC code does not end on a byte boundary, the Compressor adds enough Dummy Fill bits between the compressed DATA and the RTC code to end the line on a byte boundary. If the line processing has not been completed and the Compressor Destination Buffer overflow occurs, the Compressor terminates its operation promptly and the RTC code won't be suffixed.

As mentioned for Multi-Line operation, cases of error termination are indicated by the Line Processing Incomplete (LPI) bit in the Compressor Status Register (CSR). Although the Auto-EOL feature may be enabled, an EOL code won't be suffixed following an end of line when the suffix RTC code is defined.

No Byte Boundaries Control (10)
The No Byte Boundaries Control is the counterpart to the Process on Byte Boundaries (00) operation. The compressed data (including the Time Fill and EOL codes, if any) is not conditioned to end lines on a byte boundary.

For example, if the Auto-EOL feature is suppressed (no EOL insertion), and a coded line does not end on a byte boundary, the Compressor holds the excess compressed data beyond the last full byte boundary (one to seven bits) in an internal register. Then this remaining compressed data is combined with the next line of compressed data without Dummy Fill bits between the scan lines.

In Multi-Line operation on a Source Buffer overflow or in Single-Line operation, the Compressor processes the last full byte of source data but truncates any excess bits (one to seven) that do not comprise a byte at the end of the terminal line. In this case, the Compressor will indicate an error termination and the LPI bit will be set in the Compressor Status Register.

If a Compressor destination overflow has occurred, the Compressor terminates its operation promptly and the Compressor retains the remaining bits of the compressed data in an internal register. Under these circumstances, processing may be resumed by issuing the Compressor Go operation after preparing the Compressor Source Buffer or the Compressor Destination Buffer respectively. At this point, the Compressor combines the excess bits from the point of termination with the rest of the data to be compressed.

> As another example, if the Auto-EOL feature is enabled and a coded line (including the EOL and the Time Fill, if any) does not end on a byte boundary, the Compressor retains the excess bits of the EOL code (one to seven) in an internal register. Then the rest of the EOL code is combined with the next line of compressed data from where it left off (i.e., the EOL code won't be conditioned on a byte boundary).

Transparent Mode operation cannot be specified with the No Byte Boundaries Control. The Transparent Mode operation is always conditioned to end a line on a byte boundary, regardless of the Byte Boundary Control bit.

Bit 6 of the Compressor Parameter Register is called the Source Attribute (SA) bit. When the Source Attribute bit has been set to "1" with the K-Parameter at infinity, and the EOL bit is set to "1," the reference line for the beginning of a page will be an imaginary all-white line. In other words, all lines will be processed Two-Dimensionally according to the Group 4 recommendations. Also, for G4 compatibility, the first line of the Source Buffer must correspond to the first scan line of a page with no EOL set, No Byte Boundaries set, in Two-Dimensional Mode with "K" set to infinity, and with no Time Fill specified.

The SA bit will be sampled by the CEP as a Go operation is received. This Go operation must be specified only after the CEP is initialized by the RESET input or the Reset operation, or in other words, the CEP must be initialized at the beginning of a page. The SA bit in the Parameter Register will be cleared automatically after completing the first line of processing.

When the CEP is in a resumable condition in Two-Dimensional Mode without the K-Parameter set to infinity, and the EOL bit is set to "0," the processor will ignore the SA bit. If, however, the SA bit is set to "1" with the EOL bit set to "0," the processor will insert or detect a prefixed EOL code at the beginning of a page.

When the SA bit and EOL bit are both set to "0," the selected processor will operate without the prefixed EOL code. Thus, when the SA bit has been set to "1," the processor Source Buffer is attributed to the first portion of the page, but if the SA bit has been set to "0," the processor follows an ordinary operation procedure.

Bit 7 of the Compressor Parameter Register is called the EOL bit.

When the **Compressor** Parameter Register Bit 7 is set to "0," the Compressor will automatically suffix an EOL code to the end of a compressed line or to the end of a transparent line. With this option enabled, a compressed line or transparent line consists of data followed by Time Fill bits (if any) and an EOL code. This line will end on a byte boundary when the Byte Boundaries Control is specified; otherwise, it will not.

If this bit is set to "1," such automatic suffixing is suppressed. With this option enabled, a compressed line will consist of data only (no Time Fill bits or EOL codes).

When Auto-EOL is enabled with the Source Attribute bit in the Compressor Parameter Register set to "1," an EOL code will be prefixed to the first compressed line of the page. This prefixed EOL code always ends on an exact byte boundary when in the Transparent Mode because, in this case, it happens to be 16 bits.

Expander Parameter Bits

Bits 0, 1, and 2 of the Expander Parameter Register contain the K-Parameter. The K-Parameter specifies the number of two-dimensional lines to be encoded or decoded (K-1) with respect to each one-dimensional line. Values from one to seven and infinity (code 000) can be selected.

If One-Dimensional coding or Transparent Mode is specified in the Master Control Register, the selected processor logic ingores the K-Parameter.

The state of the K-Parameter bits after initialization by the RESET input is not specified.

Bits 3, 4, and 5 of the Expander Parameter Register are used to specify the G-Parameter granularity control. G0, G1, and G2 refer to **Bits 3, 4, and 5** of the Expander Parameter Register respectively. The following table lists the G-Parameters and the corresponding code:

Bit 5 G2	Bit 4 G1	Bit 3 G0	G-Parameter
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The G-Parameter is used to specify the number of times that each scan line (a scan line has been specified by the value in the Expander Page Width Register) should be duplicated in the Destination Buffer. For instance, when G=3, each scan line that is accessed from the Source Buffer and then expanded is written into the Destination Buffer a total of four times. Note that an error condition will exist if the Expander Wraparound Register (EWR) is not "0" when the G-Parameter is non-zero. Such an error would be indicated by the EIC bit in the Expander Status Register. The Expander Wraparound Register, Expander Source Working Count Register, and Expander Destination Working Count Register are more fully described in their separate section of this document.

Bit 6 of the Expander Parameter Register is called the Source Attribute (SA) bit. When the Source Attribute bit has been set to "1" with the K-Parameter at infinity, and the EOL bit is set to "1," the reference line for the beginning of a page will be an imaginary all-white line. In other words, all lines will be processed Two-Dimensionally according to the Group 4 recommendations. Also, for G4 compatibility, the first line of the Source Buffer must correspond to the first scan line of a page with no EOL set, No Byte Boundaries set, in Two-Dimensional Mode with "K" set to infinity, and with no Time Fill specified.

The SA bit will be sampled by the CEP as a Go operation is received. This Go operation must be specified only after the CEP is initialized by the RESET input or the Reset operation, or in other words, the CEP must be initialized at the beginning of a page. The SA bit in the Parameter Register will be cleared automatically after completing the first line of processing.

When the CEP is in a resumable condition in Two-Dimensional Mode without the K-Parameter set to infinity, and the EOL bit is set to "0," the processor will ignore the SA bit. If, however, the SA bit is set to "1" with the EOL bit set to "0," the processor will insert or detect a prefixed EOL code at the beginning of a page.

When the SA bit and EOL bit are both set to "0," the Expander will operate without the prefixed EOL code. Thus, when the SA bit has been set to "1," the processor Source Buffer is attributed to the first portion of the page, but if the SA bit has been set to "0," the processor follows an ordinary operation procedure.

The state of the SA bit after initialization by the RESET input is not specified.

Bit 7 of the Expander Parameter Register is called the EOL bit.

When the **Expander** Parameter Register Bit 7 is set to "0," the Source Buffer data is assumed to contain EOL codes. This data will be checked for data errors. If the EOL bit is not set to "0," the data will not be checked for data errors; error-free data is assumed.

When the EOL bit is a "1" during the Transparent Mode or the One-Dimensional or the Two-Dimensional Modes, the Source Buffer data is assumed to contain no EOL codes and no Time Fill bits.

If the Expander GO bit is set to "1" with the EOL bit set to "1" and Expander SA bit set to "1," the Expansion Processor will start its operation when it has detected an EOL code from the previous line or prefixed from the beginning of a page, and it will continue its operation until it detects an EOL code on the current processing line. When this EOL has been detected, the Expansion Processor will check that the bit-length of the expanded current line is equal to

$$L = (EPWR * 8) * (EWR + 1)$$

Where: EPWR = The value of the Expander Page Width Register.
EWR = The value of the Expander Wraparound Register.

If the current line is not equal to this value, the Expansion Processor will terminate in an error-state with the DER bit in the Expander Status Register set to "1."

Caution: The Expander Byte Boundary Control bit is located in the Restart Control Register.

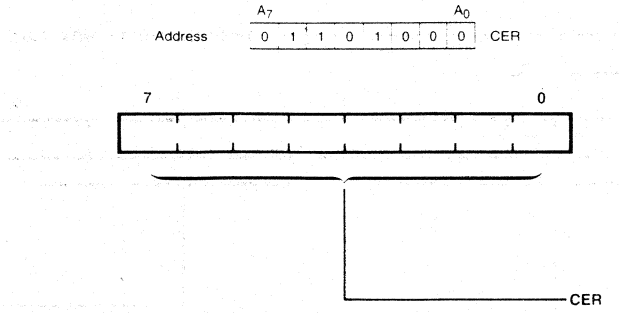
EXPRESS REGISTER (CER)

This 8-bit register specifies how many scan lines to skip before compressing the next line after the processing of the current line has been completed.

Figure 24 illustrates the Express Register. This register is used to specify (in binary) how many scan lines to skip after compressing the current line. If this register is loaded with a "0," every scan line will be compressed; this is the normal operating mode. The Compressor logic will not modify this register during its operation.

If the Express Mode is defined with Two-Dimensional Compression, the current compressed line will become the reference line for the next compressed line which is located "n" scan lines below it.

The Am7970 does not allow Wraparound and Express Modes to be specified simultaneously. If such a condition is specified, an error status will be indicated. The scan line length is obtained from the Compressor Page Width Register.



05777B-24

Figure 24. Compressor Express Register (CER)

05777B

Refer to page 7-1 for Essential Information on Military Devices

WRAPAROUND REGISTER (CWR/EWR)

Figure 25 shows the Wraparound Register layout. This 16-bit register is used to specify the number of additional scan lines that will be grouped into one effective line. This effective line will be used for encoding and decoding. If the Wraparound Register is loaded with "0," then the effective line is identical to a scan line. This is the normal operating mode.

If this register is loaded with a "1," two scan lines will make up an effective line and so on. Wraparound Mode and Express Mode cannot be used simultaneously, and Wraparound Mode cannot be specified with Two-Dimensional Compression. Either of these conditions will result in an error. The Wraparound Register is not modified by the selected processor during its operation.

LEFT MARGIN REGISTER (LMGR)

Figure 26 shows the Left Margin Register layout. This 16-bit register is used to specify the width of the left-hand margin. If

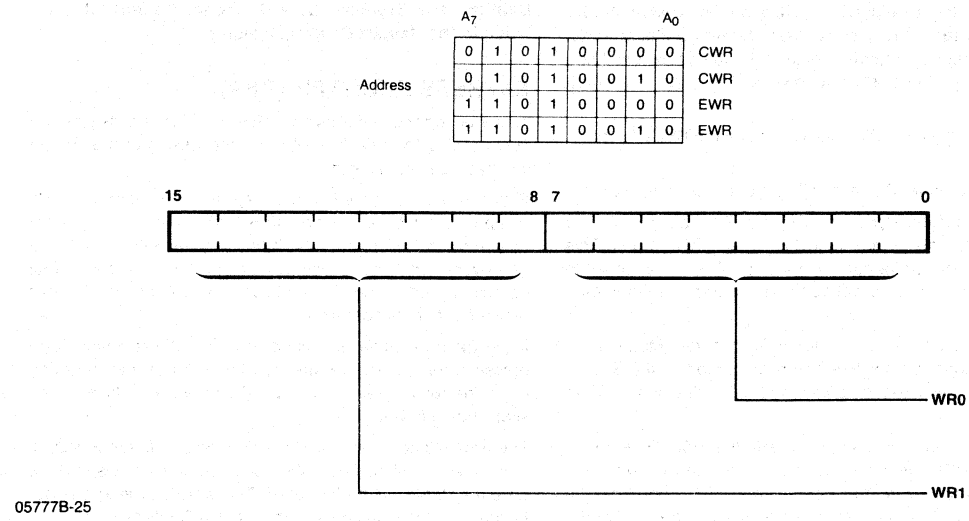


Figure 25. Compressor/Expander Wraparound Register (CWR/EWR)

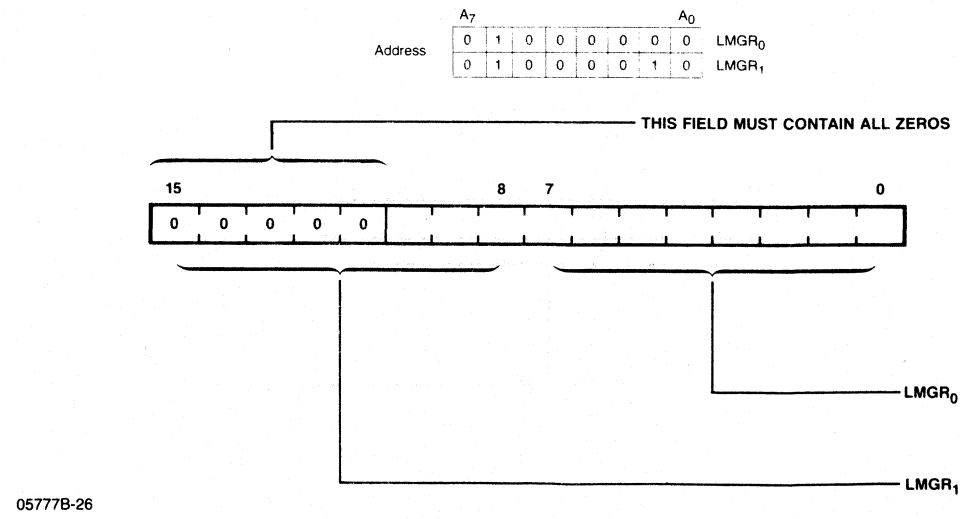


Figure 26. Left Margin Register (LMGR)

the value in the Left Margin Register is "0," then the original scan line will be used with no modification. When the Left Margin Register is non-zero, the number contained in this register specifies the left margin width in bytes.

For example, a "1" in the Left Margin Register means 8 pels of margin. When a compression operation is initiated with the left margin specified, then the Compressor obtains data from the Source Buffer via DMA as usual. However, the margin specification overrides the actual image data and forces the pels to be "white." Such overriding continues until the programmed margin requirements are satisfied.

Compression of the rest of the scan line proceeds as usual (see also Right Margin Register). The left margin is effective in Wraparound and Express Mode and will be included in One-Dimensional, Two-Dimensional, and Transparent Modes of operation.

Hence, if the left margin is specified while using Transparent Mode to accomplish a transfer of data from the Source Buffer to the Destination Buffer, the final data in the destination will differ from the data in the source because of the margin. The Compressor does not modify the Left Margin Register during its operation.

The sum of the left and right margin specifications must not be greater than the paper width specified. This would result in an error condition.

Bits 11 through 15 of the Left Margin Register must be set to "0."

RIGHT MARGIN REGISTER (RMGR)

Figure 27 shows the Right Margin Register layout. This 16-bit register is used to specify the width of the right hand margin. If the value in the Right Margin Register is "0," the original scan line will be used with no modification. When the Right Margin Register is non-zero, the number contained in this register specifies the right margin width in bytes.

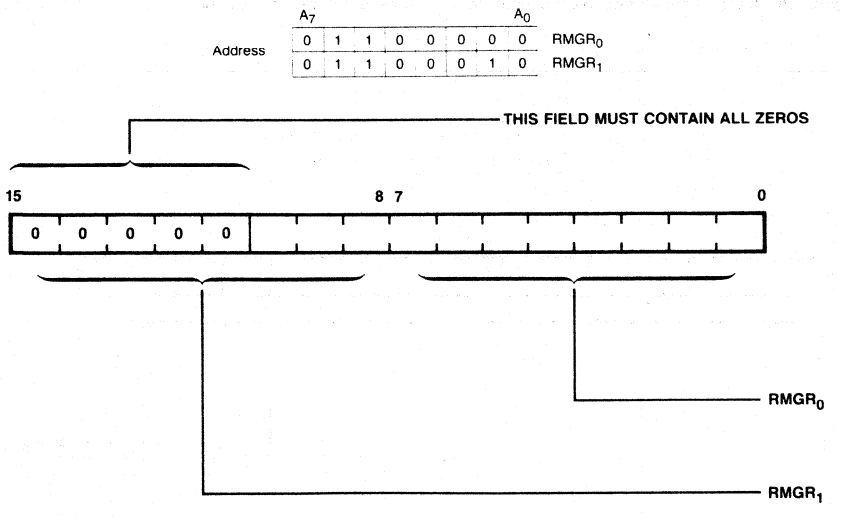
For example, a "1" in the Right Margin Register means 8 pels of margin. When a compression operation is initiated with the right margin specified, then the Compressor obtains data from the Source Buffer via DMA as usual. However, the margin specification overrides the actual image data and forces the pels to be "white." Such overriding continues until the programmed margin requirements are satisfied.

Compression of the rest of the scan line proceeds as usual (see also Left Margin Register). The right margin is effective in Wraparound and Express Mode and is included in One-Dimensional, Two-Dimensional, and Transparent Modes of operation.

Hence, if the right margin is specified while using Transparent Mode to accomplish a transfer of data from the Source Buffer to the Destination Buffer, the final data in the destination will differ from the data in the source because of the margin. The Compressor does not modify the Right Margin Register during its operation.

The sum of the left and right margin specifications must not be greater than the paper width specified. This would result in an error condition.

Bits 11 through 15 of the Right Margin Register must be set to "0."



05777B-27

Figure 27. Right Margin Register (RMGR)

TOP MARGIN REGISTER (TMGR)

Figure 28 shows the Top Margin Register layout. This register is used to specify the top margin width of a document. If the Top Margin Register is loaded with a "0," no top margin is specified.

If the Top Margin Register is non-zero, the number contained in this register specifies the desired top margin height in increments of one scan line. When a compression operation is initiated with the top margin specified, then the Compressor reads data from the Source Buffer via DMA as usual. However, the top margin specification overrides the data and forces "white" into the Compressor until the top margin requirements are satisfied. From then on, the usual compression operation takes place (also see Left and Right Margin Registers).

Since, by definition, the Top Margin white space is to occur only once per document, the Compressor logic decrements the Top Margin Register by one after processing each scan line until it reaches "0", at which time normal compression proceeds.

> The top margin is effective in both Wraparound and Express Modes. However, caution must be exercised when specifying Express Mode with a top margin since the Compressor logic of the Am7970 picks every "Nth" line ("N" being a function of the Express Register) to compress in Express Mode. For example, assume that Top Margin Register specifies "8" and the Express Register specifies "1." The Compressor then processes every other scan line (scan line 1, 3, etc.) in accordance with the Express Register specification, yet the Top Margin is affected by only those scan lines that were presented to the Compressor. Therefore, since the Top Margin is assigned to be "8," and every other scan line has been skipped, scan line "17" of the original picture will be the first coded line with real picture data on it in this example.

The top margin controls are effective in One-Dimensional, Two-Dimensional, and Transparent Modes. Hence, consideration must be given to the effects of Top Margin Register when using Transparent Mode to transfer data from the Source Buffer to the Destination Buffer.

TIME FILL REGISTER (TFLR)

Figure 29 shows the Time Fill Register layout. This 8-bit register is used to specify the minimum length of a coded line expressed in bytes. If the number of bytes in a compressed line is smaller than this number, Time Fill bits are required. Time Fill bits will be added to the compressed line such that the sum of the code bits and Time Fill bits is equal to or greater than the required line length. Time Fill bits are simply all "0s."

Specifying "0" in the Time Fill Register means that no time fill is desired, or in other words, zero minimum length is acceptable.

When the Auto-EOL feature is suppressed, the Am7970 will ignore the time fill requirement; no time fill is inserted. When the Auto-EOL is enabled and a coded line including Time Fill (if any) and EOL code does not end on a byte boundary, the Am7970 will add enough Dummy Fill bits between the compressed data and the EOL code to end the line on a byte boundary when the Byte Boundary Control has been specified. When the No Byte Boundaries Control is specified, the Am7970 Compressor will **not** condition the Time Fill bits to end lines on a byte boundary.

SOURCE ADDRESS HOLDING REGISTER (CSAHR/ESAHR)

Figure 30 shows the Source Address Holding Register layout. This 24-bit register contains the starting address of the Source Buffer for the selected processor. When a Restart process is initiated, the Source Address Holding Register provides the initial value to the working register. The contents of Source Address Holding Register are loaded automatically into the Source Current Address Register whenever the GO bit in the Master Control Register is set, in conjunction with the Source Address Control bit in the respective Restart Control Register being "0." The contents of the Source Address Holding Register are not modified by the processor during its operation.

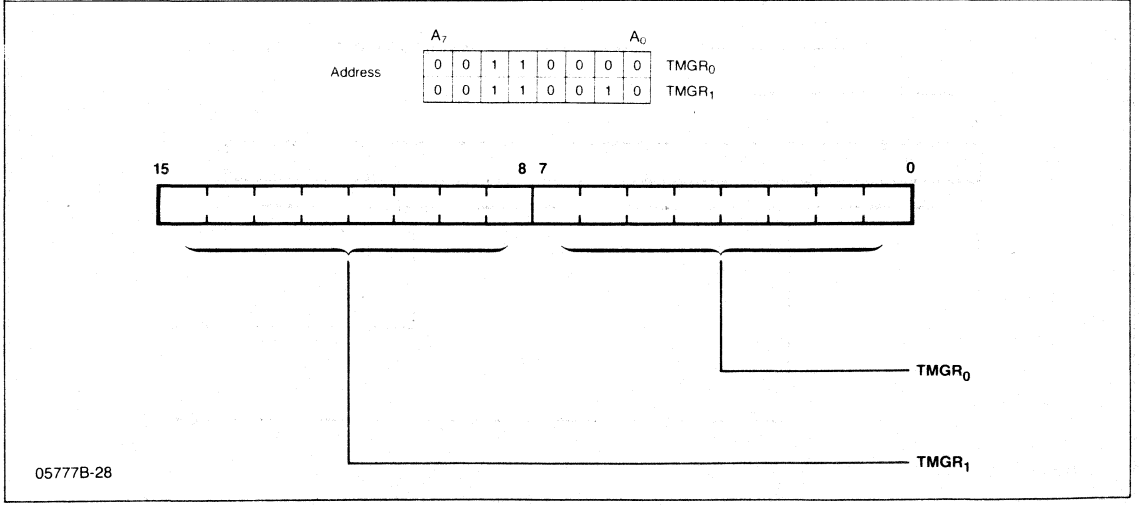
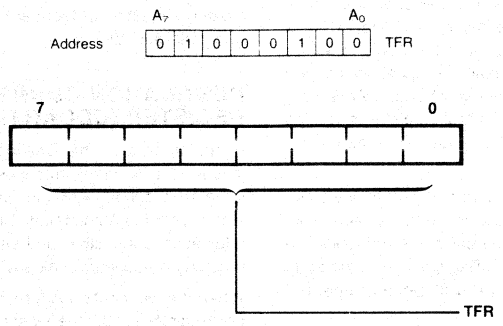


Figure 28. Top Margin Register (TMGR)

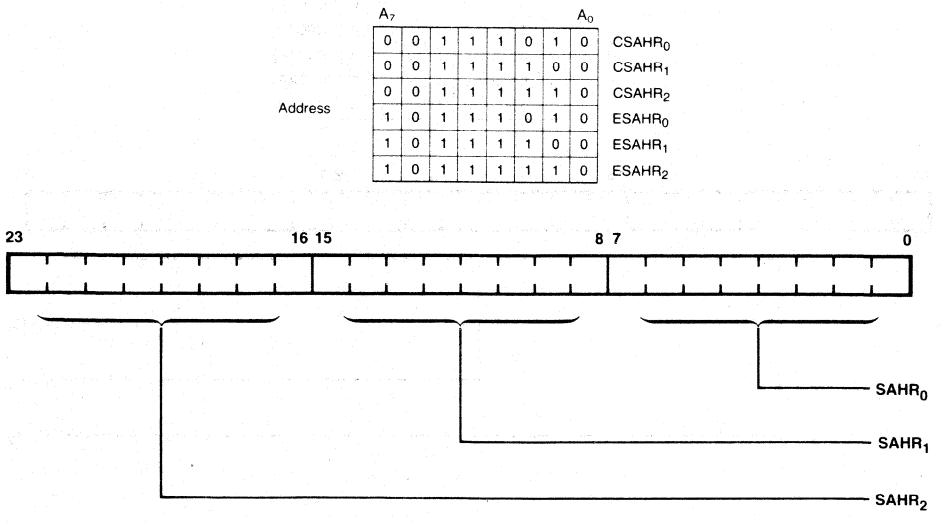
05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B-29

Figure 29. Time Fill Register (TFR)



05777B-30

Figure 30. Compressor/Expander Source Address Holding Register (CSAHR/ESAHR)

05777B

Refer to page 7-1 for Essential Information on Military Devices

SOURCE CURRENT ADDRESS REGISTER (CSCAR/ESCAR)

Figure 31 shows the Source Current Address Register layout. This 24-bit register provides the current address for all Compressor transactions with the Source Buffer. After each transaction this register is incremented by one and will wraparound through "0" after reaching a maximum value of all "1s." If the Source Address Control bit in the selected Restart Control Register is "0," the Source Current Address Register will be loaded from the Source Address Holding Register whenever a new Go operation is initiated. On the other hand, if the Source Address Control bit in the Restart Control Register is "1," the Source Current Address Register continues from its current value.

Two-Dimensional processing requires not only data for the current line but also corresponding data from the previous line. The Am7970 calculates the initial address of the reference line using the Source Line Start Address Register. From there on, the address of the reference line will be incremented appropriately.

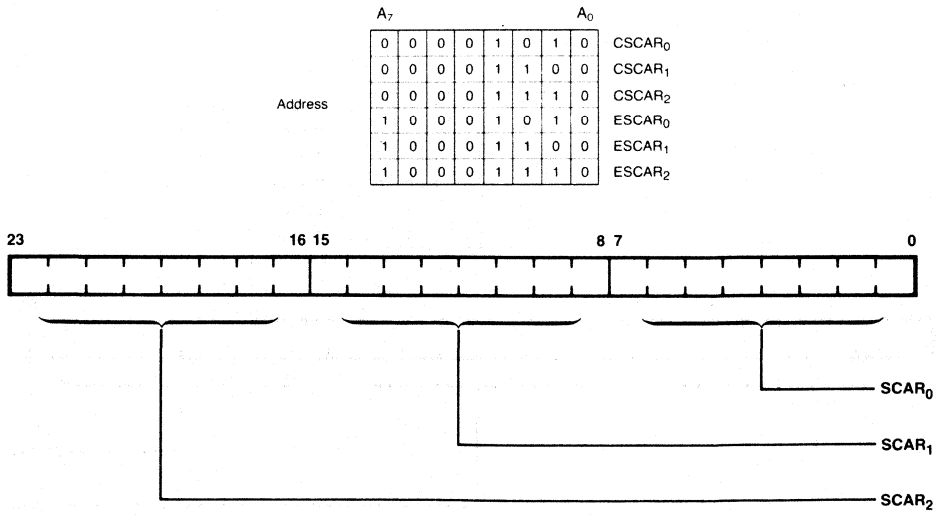
DESTINATION ADDRESS HOLDING REGISTER (CDAHR/EDAHR)

Figure 32 shows the Destination Address Holding Register layout. This 24-bit register is the counterpart to the Source Address Holding Register. It specifies the starting address of the Destination Buffer during Restart operations. The contents of the Destination Address Holding Register will be automatically transferred into the Destination Current Address Register whenever the GO bit in the Master Control Register is set in conjunction with the Destination Address Control bit in the Restart Control Register being "0." The Destination Address Holding Register contents are not modified by the Compressor during its operation.

DESTINATION CURRENT ADDRESS REGISTER (CDCAR/EDCAR)

Figure 33 shows the Destination Current Address Register layout. This 24-bit register is used to obtain the current address for a transaction with the Destination Buffer. The Am7970 increments the Destination Current Address Register by one after each transaction and will wraparound through "0" after reaching a maximum value of all "1s."

When the Destination Address Control bit in the Restart Control Register is "0," the Destination Current Address Register will be loaded from the Destination Address Holding Register whenever the GO bit in the Master Control Register is set to "1." When the Destination Address Control bit is "1," the Destination Current Address Register will continue from the current value upon receipt of the Go operation.

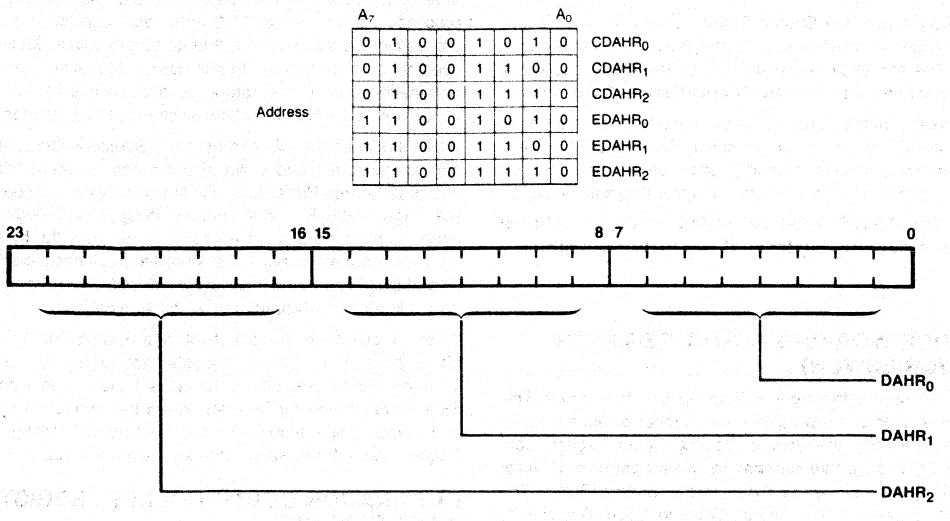


05777B-31

Figure 31. Compressor/Expander Source Current Address Register (CSCAR/ESCAR)

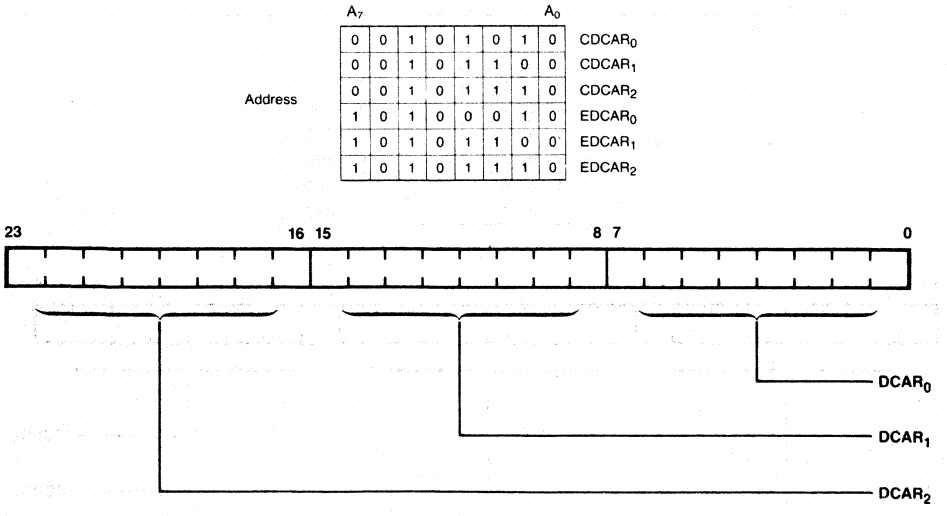
05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B-32

Figure 32. Compressor/Expander Destination Address Holding Register (CDAHR/EDAHR)



05777B-33

Figure 33. Compressor/Expander Destination Current Address Register (CDCAR/EDCAR)

SOURCE COUNT HOLDING REGISTER (CSCHR/ESCHR)

Figure 34 shows the Source Count Holding Register layout. This register is used to specify the Source Buffer length in bytes. The Am7970 requires that this buffer length be specified as a **negative number in two's complement form.**

The contents of the Source Count Holding Register are loaded automatically into the Source Working Count Register whenever a Go operation is initiated in conjunction with the Source Count Control bit in the Restart Control Register being "0."

Note: The CSCHR must be loaded with Source Buffer Length divided by contents of the (Express Register + "1").

SOURCE WORKING COUNT REGISTER (CSWCR/ESWCR)

Figure 35 shows the Source Working Count Register. This register is used to keep track of the number of Source Buffer accesses. Initially, the Source Working Count Register contains a 24-bit **negative number in two's complement form** specifying the number of bytes in the Source Buffer. The Am7970 increments the Source Working Count Register by one after completing each source transaction. The Source Overflow (SO) bit in the appropriate Status Register will be set to "1" immediately after the Source Working Count Register is exhausted.

It must be emphasized that a transaction does not necessarily mean one access. For example, for Two-Dimensional processing, the Am7970 will access the Source Buffer both for the current line and for the reference line. Thus, a transaction in Two-Dimensional will include twice as many Source Buffer accesses as a One-Dimensional transaction of the same type.

When the last byte of the Source Buffer corresponds to the end of an effective line, the Am7970 will process this byte and will attempt to store it in the destination and then terminate the processor. When the Am7970 terminates after storing a fully processed line successfully (that is, no premature destination overflow or error status), this is normal operation. Abnormal termination due to premature destination overflow or error status will be discussed in later sections of this document.

When the last byte obtained from the Source Buffer does not correspond to the end of an effective line, it means that the assigned Source Buffer length is less than what is needed by the Page Width Register and the Wraparound Register, if used. In this condition the Am7970 will process the byte and then terminate abnormally. The Wraparound Incomplete (WPI) bit and Line Processing Incomplete (LPI) bit in the Status Register will reflect the appropriate error termination.

When an operation is initiated with the Source Count Control bit in the Restart Control Register set to "0," the Source Working Count Register obtains its initial value from the Source Count Holding Register. When this control bit is a "1" and a Restart operation is initiated, the Source Working Count Register uses the existing value as the initial value.

DESTINATION COUNT HOLDING REGISTER (CDCHR/EDCHR)

Figure 36 shows the Destination Count Holding Register. This 24-bit register is used to specify the length (in bytes) of the Destination Buffer. The buffer length must be specified as a **negative number in two's complement form**. The contents of the Destination Count Holding Register are loaded automatically into the Destination Working Count Register whenever a Restart operation is initiated with the Destination Count Control bit in the Restart Control Register set to "0."

The Compressor logic does not modify this register during its operation.

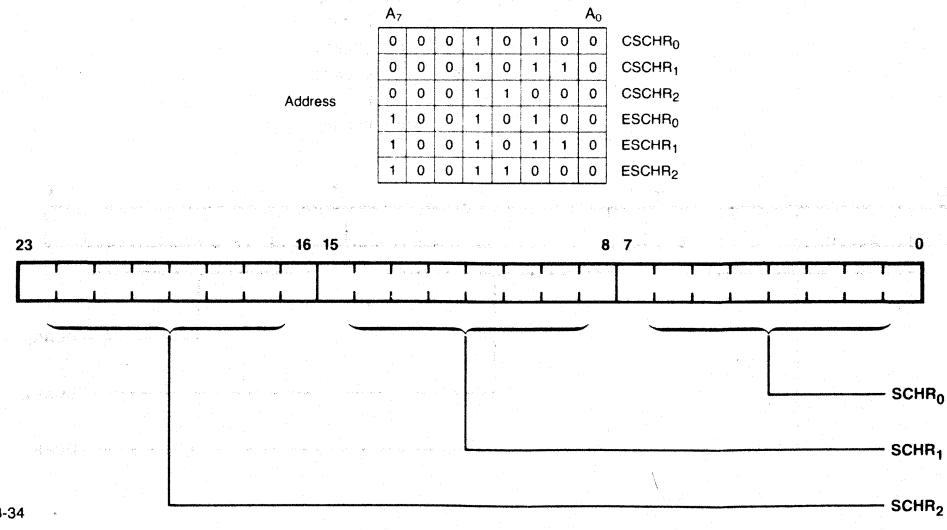
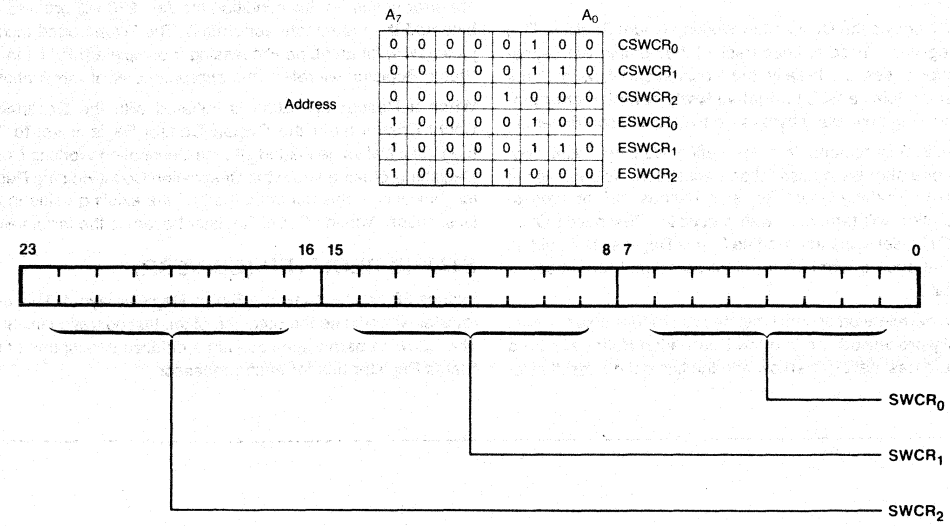


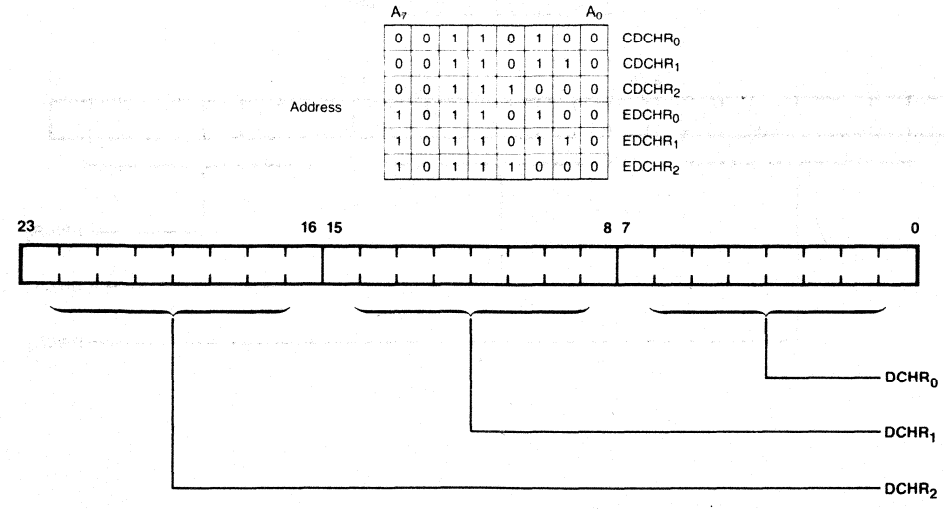
Figure 34. Compressor/Expander Source Count Holding Register (CSCHR/ESCHR)

05777B



05777B-35

Figure 35. Compressor/Expander Source Working Count Register (CSWCR/ESWCR)



05777B-36

Figure 36. Compressor/Expander Destination Count Holding Register (CDCHR/EDCHR)

05777B

Refer to page 7-1 for Essential Information on Military Devices

DESTINATION WORKING COUNT REGISTER (CDWCR/EDWCR)

Figure 37 shows the Destination Working Count Register. This 24-bit register is used to keep track of the number of Destination Buffer accesses. Initially, the Destination Working Count Register contains a 24-bit **negative two's complement number** specifying how many bytes long the Destination Buffer is.

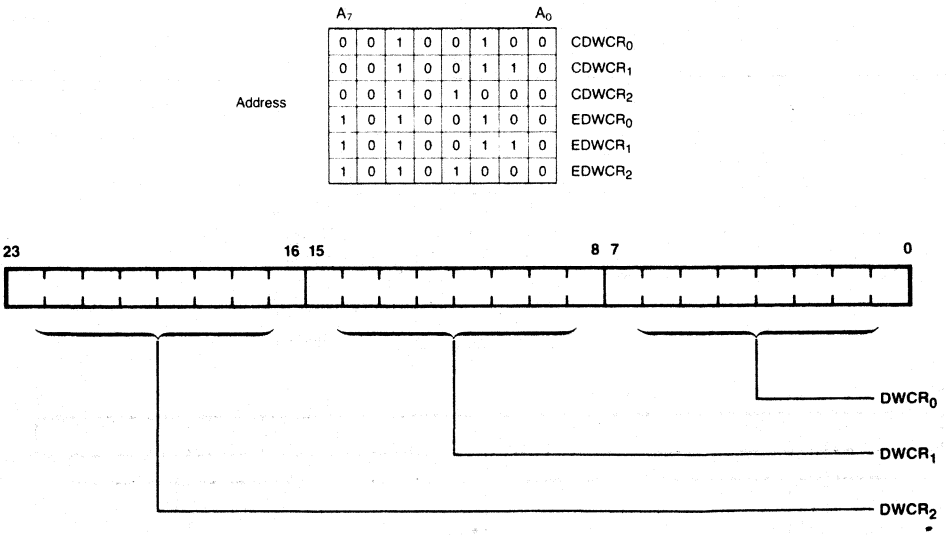
The Am7970 increments the Destination Working Count Register by one after each destination transaction. As soon as the Destination Working Count Register reaches "0," the processor operation will terminate with processor Destination Overflow (DO) bit set in the appropriate Status Register. It should be noted that such a termination might be normal (error-free) or abnormal.

A normal termination occurs when the Am7970 is able to transfer a fully-processed line into the Destination Buffer without a premature destination overflow and the line did not result in an error such as negative compression. An abnormal termination due to premature destination overflow will result in an appropriate error status. In this condition the Am7970 will process the byte and then terminate abnormally. The Wraparound Incomplete (WPI) bit and Line Processing Incomplete (LPI) bit in the Status Register will reflect the appropriate error termination.

When a Restart operation is initiated with the Destination Count Control bit in the Restart Control Register set to "0," then the initial value loaded into the Destination Working Count Register is obtained from the Destination Count Holding Register. However, if this control bit is "1," the existing value in the Destination Working Count Register becomes the initial value.

STATUS REGISTER (CSR/ESR)

Figures 38 and 39 show the Status Register layout. This 8-bit register will indicate the outcome of the last operation initiated. The following paragraphs contain a detailed description of the Status Register bits for each processor.

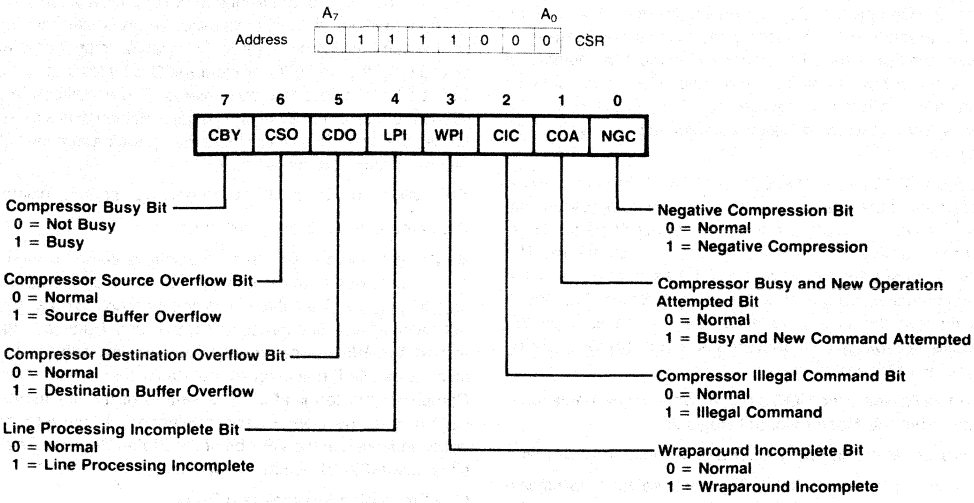


05777B-37

Figure 37. Compressor/Expander Destination Working Count Register (CDWCR/EDWCR)

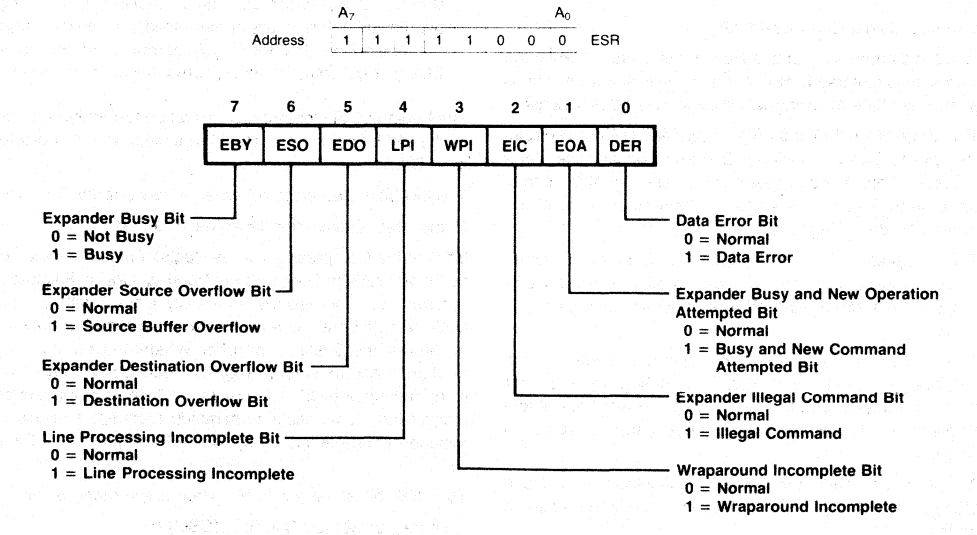
05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B-38

Figure 38. Compressor Status Register (CSR)



05777B-39

Figure 39. Expander Status Register (ESR)

Compressor Status Bits

Negative Compression (NGC) bit:

Bit 0 in the Compressor Status Register is set to "1" to indicate that the process of compressing the current line resulted in negative compression. This means that the total number of bytes in the compressed line is more than the number of bytes contained in the original effective line. The Am7970 checks for negative compression only after *completely processing* an effective line.

Prepositional EOL (if any) and Dummy Fill (if any) are included in determining the total number of bytes in a compressed line. Thus, the NGC indication reflects the effect of prepositional EOL and Dummy Fill bits in addition to the actual data. The Am7970 computes the number of bytes contained in the original effective line based on the contents of the Page Width Register and the Wraparound Register (when appropriate). Negative compression conditions are not checked during Transparent Mode of operation.

The Am7970 clears the NGC bit to "0" when a new operation is initiated from the Master Control Register.

Compressor Busy and New Operation Attempted (COA) bit:

Bit 1 in the Compressor Status Register is set to "1" to indicate that an attempt was made to write a new operation into the Master Control Register while the Compressor is still busy.

All registers which require user-specification in the Am7970 can be read as well as written by the host CPU. However, modifying the registers while the Compressor is operating is not allowed. Hence, the Am7970 will ignore any attempt to write into a register while the Compressor is busy. However, any erroneous attempt to write into the Master Control Register at such times sets the COA bit to "1" and marks the beginning of a Compressor termination. The actual termination will be indicated by the Compressor Busy (CBY) bit.

The COA bit will be cleared to "0" when a new operation is initiated.

Compressor Illegal Command (CIC) bit:

Bit 2 in the Compressor Status Register is set to "1" when the Compressor is directed to start operating with illegal conditions present. Any of the following will result in this status indication.

a) The Mode bits (CM0 and CM1) in the Master Control Register specify Two-Dimensional Compression and the Compressor Wraparound Register is non-zero. In other words, specifying Two-Dimensional Compression with Wraparound Mode is illegal.

b) The Compressor Express Register and Compressor Wraparound Register are both non-zero. In other words, specifying Express Mode and Wraparound Mode together is illegal.

c) The sum of the left and right margins represented by the Left Margin Register and Right Margin Register is greater than the page width specified in Compressor Page Width Register. In other words, specifying overlapping margins is illegal.

d) The Mode bits (CM0 and CM1) or the Control bits (CC0 and CC1) in the Master Control Register specify the reserved code.

e) The Compressor Page Width Register has been specified as "0."

From the above, it is apparent that the CIC bit is a check on the set-up conditions in the Compressor before it starts an operation. When the GO bit is set to "1" (that is, "start Compressor operation"), the Am7970 will clear the CIC bit to "0" and set the CBY bit to "1" to indicate busy status. The conditions for CIC status are then checked. If any illegal condition exists as stated above, the Am7970 will terminate the operation with the CIC bit set to "1" and CBY bit set to "0."

The state of this bit is "0" after a new operation is initiated.

Wraparound Incomplete (WPI) bit:

Bit 3 in the Compressor Status Register is set to "1" when the Compressor has terminated prior to successfully compressing an effective line. This status signifies that the Compressor has not satisfied the Compressor Wraparound Register requirements. The WPI bit being set in conjunction with the LPI bit gives a detailed indication of the status that exists when the Compressor is terminated. The setting of this bit marks the beginning of the Compressor termination. Actual termination will be indicated in the CBY bit of the Status Register. The WPI bit is cleared to "0" when a new operation is initiated.

Line Processing Incomplete (LPI) bit:

Bit 4 in the Compressor Status Register is set to "1" to indicate that the Compressor terminated without successfully processing a complete line; either a source or destination overflow occurred prematurely. There are three situations in which this occurs:

a) When the last byte obtained from the Source Buffer did not correspond to the last byte of the page and the Compressor Source Working Count Register overflowed.

b) When a Compressor Destination Working Count Register overflowed before the Compression operation reached the end of a scan line.

c) When a Compressor Destination Working Count Register overflowed before a fully compressed line could be stored in the destination. The term "fully compressed line" includes EOL (if any), data, Time Fill (if any), and Dummy Fill (if any).

The beginning of Compressor operation termination is marked by the LPI bit being set. Actual termination will be indicated by the CBY bit.

The LPI bit is cleared to "0" when a new operation is initiated.

Compressor Destination Overflow (CDO) bit:

Bit 5 of the Compressor Status Register is set to indicate that the Compressor Destination Working Count Register has reached "0." This register is initially loaded with a negative two's complement value and is incremented after each transaction with the Destination Buffer. When the Compressor Destination Working Count Register reaches "0" due to such incrementing, the CDO bit is set to "1," which marks the beginning of the Compressor termination process. The actual termination will be indicated by the CEP clearing the CBY bit to "0."

The CDO bit is cleared to "0" when a command is initiated.

Compressor Source Overflow (CSO) bit:

Bit 6 of the Compressor Status Register is set to "1" to indicate that the Compressor Source Working Count Register has reached "0." This register is initially loaded with a negative two's complement value and is incremented by one after each Source Buffer transaction. When the CSWCR reaches "0" due to such incrementing, source overflow has occurred and the CSO bit is set to "1." The Compressor will begin to terminate its operation after setting the CSO bit. Thus, there will be some elapsed time between setting the CSO bit and the actual termination as indicated by the CBY bit.

The CSO bit will be cleared to "0" when a command is initiated.

Compressor Busy (CBY) bit:

Bit 7 of the Compressor Status Register is set to "1" by the CEP to indicate that the Compressor is busy. Whenever a new operation is initiated (that is, the GO bit is set), the CBY will indicate busy status. The CBY bit will automatically become "0" when the Compressor terminates its operation. External access to any of the Am7970 registers is valid only when the CBY is "0," indicating "not busy." Otherwise the Am7970 might be modifying the registers as part of its normal operation.

If the CIE bit in the Master Control Register is "1," an interrupt to the CPU is asserted when the CBY becomes "not busy." The CSR bit can be polled by the host CPU for an indication of the completion of an operation.

Expander Status Register bits

Data Error (DER) bit:

Bit 0 in the Expander Status Register is called the Data Error (DER) bit. This bit indicates if a data error has been detected in the current effective line. If Bit 0 is set to "1," a premature termination of the Expander results. The address of the erroneous effective line will be recorded in the Expander Source Line Start Register. An error exists when the bit-length of an expanded effective line is not equal to

$$L = (EPWR * 8) * (EWR + 1)$$

Where: EPWR = The value of the Expander Page Width Register
EWR = The value of the Expander Wraparound Register.

An error might have resulted from a hardware failure or the inadvertent transformation of a valid Modified Huffman code word to another bit pattern due to noise. An unrecognizable code word would be ignored by the Expander. An effective line that contained an unrecognizable code word would be perceived by the Expander as too short and would cause the DER bit to be set to "1." **Note that the DER bit will only be set if the EOL bit in the Expander Parameter Register is set to Auto-EOL.**

The state of the DER bit, after a new command has been initiated by the the CPU, is "0."

Expander Busy and New Operation Attempted (EOA) bit:

Bit 1 is called the Expander Busy and New Operation Attempted (EOA) bit. This bit is set to "1" when an attempt is made to write a new command into the Expander Master Control Register while the Expander is still busy. Once the EOA bit is set to "1," the Expander will prematurely terminate and the Expander Busy (EBY) bit is cleared to "0."

After command has been initiated by the CPU, the state of the EOA bit is "0."

Expander Illegal Command (EIC) bit:

Bit 2 is set to "1" to indicate that the Expander was directed to start operating with any of the following illegal conditions present.

1) A Two-Dimensional Expansion Mode and a non-zero Wraparound Register have both been specified.
2) A non-zero Granularity Parameter and a non-zero Wraparound Register have both been specified.
3) EM0 and EM1 are both specified as "1."
4) EC0 and EC1 are both specified as "1."
5) The EPWR has been specified as "0."

If an Expander command is received when one or more of these conditions are present, the EIC bit will be set to "1" and the EBY bit will be cleared to "0."

After a new command has been initiated by the CPU, the state of the EIC bit is "0."

Wraparound Incomplete (WPI) bit:

Bit 3 is set to "1" to indicate that either the Source Buffer or the Destination Buffer overflowed after a complete scan line was expanded but before an entire effective line could be expanded.

After a new command has been initiated by the CPU, the state of the WPI bit is "0."

Line Processing Incomplete (LPI) bit:

Bit 4 is is set to "1" to indicate that either the Source Buffer or the Destination Buffer overflowed before an entire scan line could be expanded and written into the Destination Buffer.

The state of the LPI bit is "0" after a new command has been initiated by the CPU.

Expander Destination Overflow (EDO) bit:

Bit 5 is set to "1" to indicate that the Expander Destination Working Count Register has reached "0." This register is initially loaded with a negative two's complement value and is incremented after each transaction with the Destination Buffer. When the Expander Destination Working Count Register reaches "0" due to such incrementing, the EDO bit is set to "1." Setting of the EDO bit marks the beginning of the Expander termination process. The actual termination will be indicated by the CEP clearing the EBY bit to "0."

The state of the EDO bit is "0" after a new command has been initiated by the CPU.

Expander Source Overflow (ESO) bit:

Bit 6 is set to "1" to indicate that the Expander Source Working Count Register has reached "0." This register is initially loaded with a negative two's complement value and is incremented by one after each Source Buffer transaction. When the Expander Source Working Count Register reaches "0" due to such incrementing, overflow has occurred and the ESO bit will be set to "1." The Expander will begin to terminate its operation at this time. Thus there will be some elapsed time between setting the ESO bit to "1" and the actual Expander termination as indicated by the EBY bit.

The state of the ESO bit is "0" after a new command has been initiated by the CPU.

Expander Busy (EBY) bit:

Bit 7 is set to "1" to indicate that the Expander is busy. Whenever a new operation is initiated, the EBY bit is set to "1." When the Expander has completed its operation, the EBY bit is reset to "0."

PAGE WIDTH REGISTER (CPWR/EPWR)

Figure 40 shows the Page Width Register layout. This 16-bit register specifies the page width or length of a scan line in increments of 8 pels. The largest line the Am7970 can handle is 16K pels long because only 11 of these 16 bits are significant. **Bits 11 through 15** must be set to "0."

CCITT recommendation T.4 covers compression and expansion of scan lines no longer than 2560 bits. The Am7970 accommodates larger page widths up to 16K bits by allowing the use of multiple make-up codes.

Before starting a processor operation, the Am7970 checks the Page Width Register value against the Left Margin Register and the Right Margin Register values to ensure that the page width is greater than or equal to the sum of left and right margins. When the margin specifications are not consistent with the page width, the Am7970 will abort after setting the Illegal Command (IC) bit in the appropriate Status Register to "1."

The processor logic does not modify the Page Width Register during its operation.

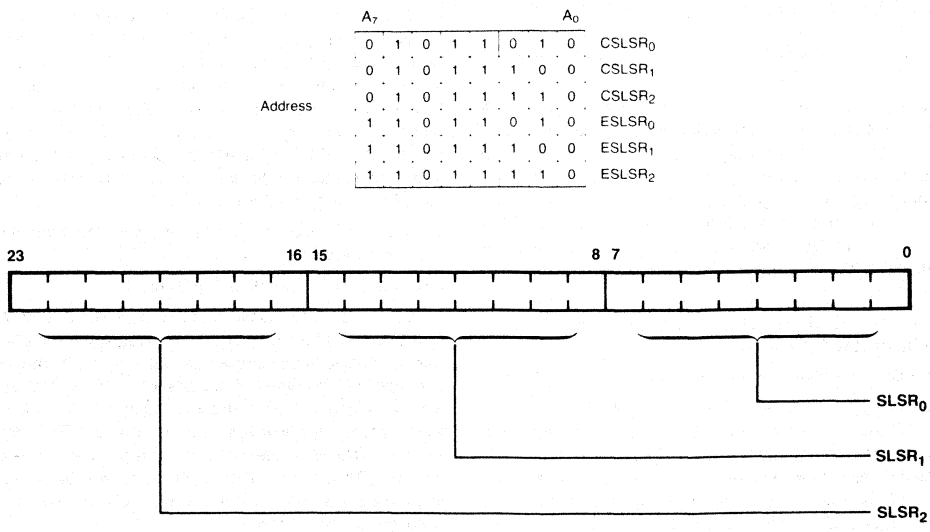
RESTART CONTROL REGISTER (CRCR/ERCR)

Figure 41 shows the Restart Control Register layout. This 8-bit register contains control bits for specifying the initial values of the Source Current Address Register, Destination Current Address Register, Source Working Count Register, Destination Working Count Register, Source Line Start Address Register, and the Destination Line Start Address Register when a new processor operation is initiated. The following is a detailed description of the individual bits.

Source Count Control (SCC) bit:
When **Bit 0** in the Restart Control Register is "0," the contents of the Source Count Holding Register will be loaded into the Source Working Count Register when a new operation is initiated. When this bit is "1," such loading will not take place and the existing contents of the Source Working Count Register will be used for the new operation.

Source Address Control (SAC) bit:
When **Bit 1** in the Restart Control Register is "0," the contents of Source Address Holding Register will be loaded into the Source Current Address Register when a new operation is initiated. When this bit is "1," such loading will not take place and the existing contents of the Source Current Address Register will be used in the new operation.

Destination Count Control (DCC) bit:
When **Bit 2** of the Restart Control Register is "0," the contents of the Destination Count Holding Register will be loaded into the Destination Working Count Register when a new operation

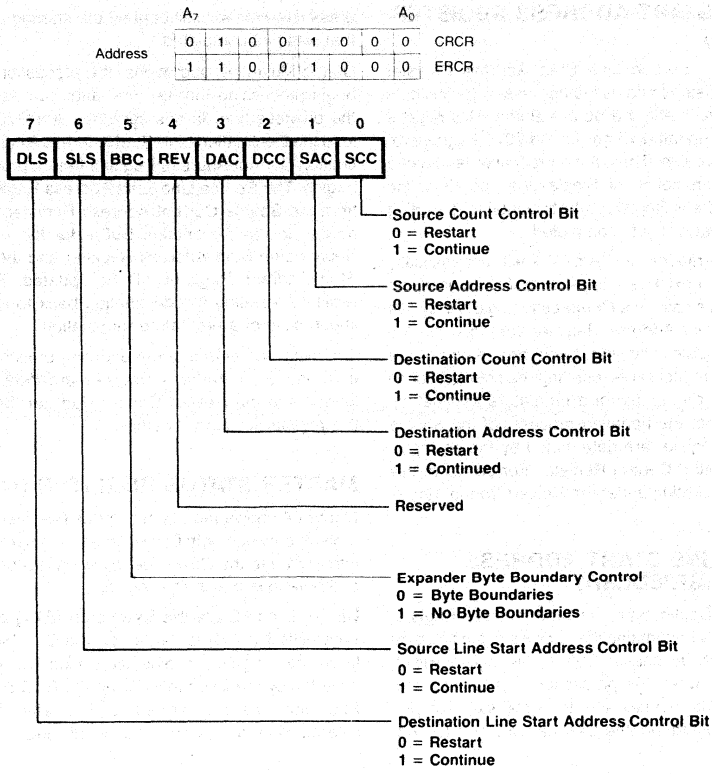


05777B-40

Figure 40. Compressor/Expander Page Width Register (CPWR/EPWR)

05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B-41

Figure 41. Compressor/Expander Restart Control Register (CRCR/ERCR)

is initiated. When this bit is "1," such loading will not take place and the existing contents of the Destination Working Count Register will be used in the new operation.

Destination Address Control (DAC) bit:
When **Bit 3** of the Restart Control Register is "0," the contents of the Destination Address Holding Register will be loaded into the Destination Current Address Register when a new operation is initiated. When this bit is "1," such loading will not take place and the existing contents of the Destination Current Address Register will be used in the new operation.

Bit 4 is RESERVED

Expander Byte Boundary Control (BBC) bit:

Bit 5 in the Restart Control Register is the Expander Byte Boundary Control (BBC) bit. This bit is used to distinguish byte padding "0s" at the end of a coded line from the starting code of the next coded line, when the compressed data does not have EOL codes. If the compressed data was byte-adjusted without EOL codes, this bit should be set to "0."

When this bit is set to "1," the Expander will transfer the one to seven Dummy Fill bits, if incurred after an EOL, into the Destination Buffer, but will keep them in an internal register to be used in the beginning of the next line.

Source Line Start Address Control (SLC) bit:
When **Bit 6** of the Restart Control Register is "0," the Source Line Start Address Register will be loaded from either the Source Address Holding Register or the Source Current Address Register depending on whether the Source Address Control bit is "0" or "1" when a new operation is issued. When the Source Line Start Address Control bit is "1," the contents of the Source Line Start Address Register will not be modified when a new command is issued.

Destination Line Start Address Control (DLC) bit:
When **Bit 7** in the Restart Control Register is "0," the Destination Line Start Address Register will be loaded from the Destination Address Holding Register or the Destination Current Address Register, depending on whether the Destination Address Control bit is "0" or "1" when a new operation is issued. When the Destination Line Start Address Control bit is "1," the contents of the Destination Line Start Address Register will not be modified when a new command is issued.

SOURCE LINE START ADDRESS REGISTER (CSLSR/ESLSR)

Figure 42 shows the Source Line Start Address Register layout. This 24-bit register contains the address of the first byte of the current effective line in the Source Buffer. This register will be automatically updated when the Am7970 begins processing a new effective line. Thus, if the processor terminated after a successful completion of Single-Line operation, the Source Line Start Address Register will still contain the starting address of the line that was just completed.

During Multi-Line operations, the Am7970 will automatically begin processing the next line after successfully processing the previous line. In this case, the Compressor logic will always load the Source Current Address Register contents into the Source Line Start Register and then begin processing of the new line. This operation facilitates resetting the source back to the beginning of the line in case of a premature termination. The initial contents (not the updating process) of the Source Line Start Address Register are determined by the setting of certain bits in the Restart Control Register. For details of this operation see the description under the Restart Control Register section.

DESTINATION LINE START ADDRESS REGISTER (CDLSR/EDLSR)

Figure 43 shows the Destination Line Start Address Register layout. This 24-bit register contains the address of the first byte of the current line in the Destination Buffer. This register will be updated automatically when the processor starts a new line. Thus, if the processor terminates after the successful completion of a Single-Line operation, the Destination Line Start Address Register will still contain the starting address of the line that was just completed.

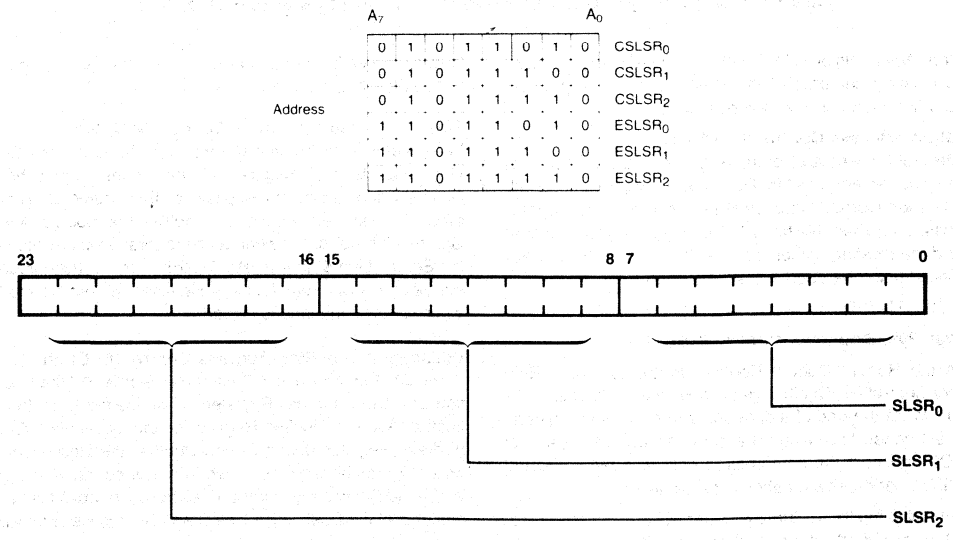
During Multi-Line operations, the processor will automatically begin processing the next line after successful completion of the previous line. In this case, the Am7970 logic will always load the Destination Line Start Address Register from the Destination Current Address Register before processing of the line begins. The Source Line Start Address Register is also loaded from the Source Current Address Register. Thus, before any access to the Destination Buffer for the new line, both the Source Line Start Address Register and the Destination Line Start Address Register will be updated. The operation will facilitate resetting the destination back to the beginning of a line in case of a premature termination.

The initial contents (not the updating process) of the Destination Line Start Address Register are determined by values loaded into the Restart Control Register. See the section on this register for further details.

MASTER STATUS REGISTER (MSR)

Figure 44 shows the Master Status Register layout. This 8-bit register provides both Expander and Compressor global status information to the CPU. The following is a detailed description of the various bits in this register.

Bits 0, 1, and 2 are the Extension (EXT) bits:
When **Bit 3** (ECD) is set to "1," the EXT bits will display the three least significant extension code bits which have been detected by the Expander. When the ECD bit is set to "0," the Extension bits are cleared to "0s." The EXT bits are also cleared when a new operation is initiated.



05777B-42

Figure 42. Compressor/Expander Source Line Start Address Register (CSLSR/ESLSR)

05777B

Refer to page 7-1 for Essential Information on Military Devices

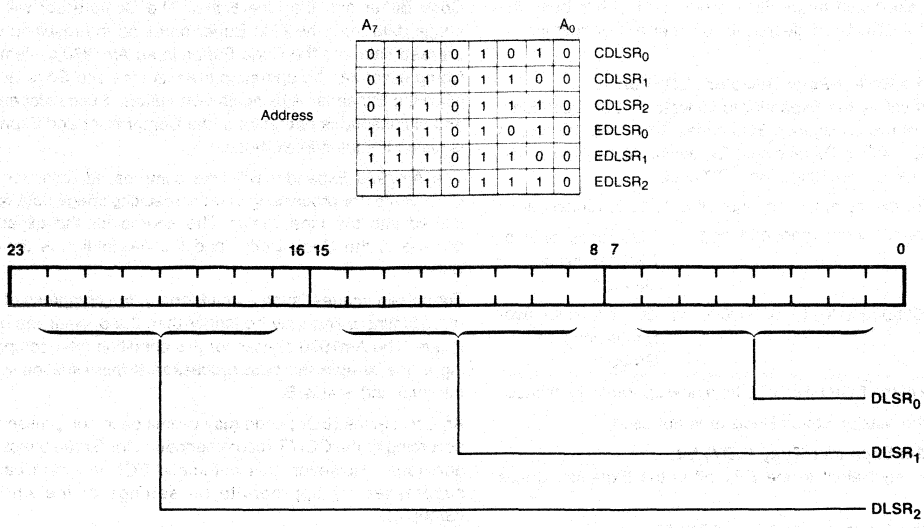


Figure 43. Compressor/Expander Destination Line Start Address Register (CDLSR/EDLSR)

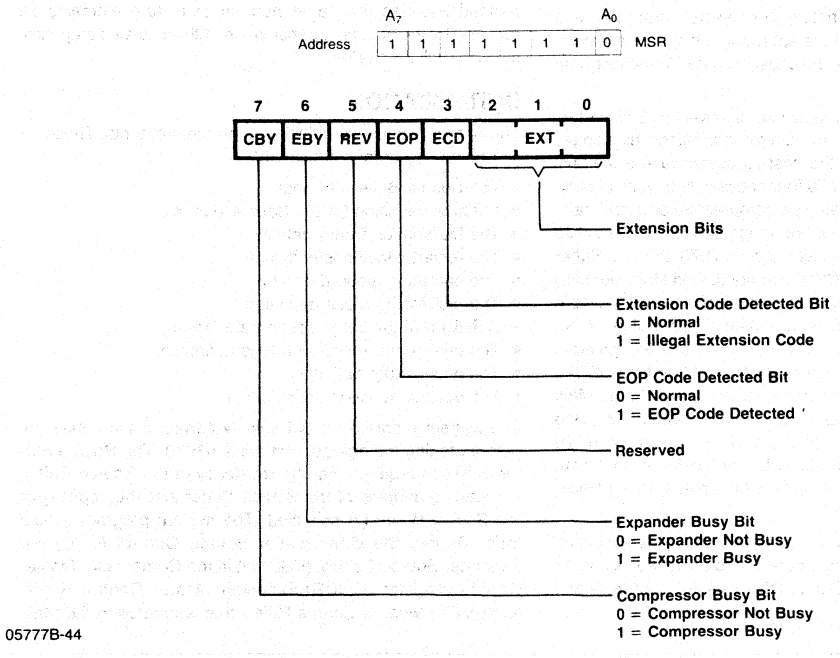


Figure 44. Master Status Register (MSR)

05777B

Refer to page 7-1 for Essential Information on Military Devices

Bit 3 is the Extension Code Detected (ECD) bit: This bit is set by the Expander to indicate that an extension code, for which the least significant three bits are not all "1s," has been detected. The ECD bit is cleared to "0" after a new operation is initiated.

Bit 4 is the End-of-Page Detected (EOP) bit:
This bit is set by the Expander to indicate that an EOP code, made up of two contiguous EOL codes, has been detected in either One- or Two-Dimensional Expansion Mode. The Transparent Mode cannot detect an EOP code.

For example, two contiguous EOL code formats are:

000000000001000000000001	EOL codes for One-Dimensional or Group 4
000000000001100000000011	EOL codes for Two-Dimensional or Group 3

The state of the EOP bit is "0" after a new operation is initiated.

Bit 5 of the Master Status Register is not used.

Bit 6 is the Expander Busy (EBY) bit:
This bit is equivalent to the EBY bit in the Expander Status Register.

Bit 7 is the Compressor Busy (CBY) bit:
This bit is equivalent to the CBY bit in the Compressor Status Register.

FUNCTIONAL DESCRIPTION

All CEP operations consist of at least three phases. In the first phase, the selected processor's registers are initialized to specify and control the desired operation. In the second phase, the processing operation itself is started and performed. The final phase involves terminating the selected processor and performing any actions that are appropriate to that termination. These different phases are described in detail in the following sections.

The Am7970 contains two separate buses—the System bus and Document Store bus. All control information exchanges between the Am7970 and the host processor take place only on the System bus. The Am7970 processes two types of data: uncompressed or image data and compressed or coded data. That portion of a memory where image data is held is called the Line Buffer. In a system using the Am7970, the Line Buffer can be located in the host CPU memory (called Main Memory) or in the Document Store. That portion of a memory where compressed data will reside is called Code Buffer. In an Am7970 system, the Code and Line Buffers can be located in the Main Memory or the Document Store. When the Am7970 needs data from the Line or Code Buffer located in the Main Memory, the transfer will take place on the System bus using the on-chip DMA facilities of the CEP. When the Am7970 needs data from the Line or Code Buffer located in the Document Store Memory, the transfer will take place on the Document Store bus.

The Am7970 is equipped with registers to specify the location of the Line Buffer, its starting address and assigned length, as well as the location of the Code Buffer, its starting address and assigned length.

The Am7970 Compressor is completely independent from the Expander. The Compressor has its own facilities to specify the Code Buffer and the Line Buffer. The Compressor will take image data from the Line Buffer and load the resulting compressed data into the Code Buffer. In an Am7970 system, the Compressor can be operating from its Line and Code Buffers while the Expander is using its own buffers. Such independent and simultaneous operation of the Compressor and Expander is called the full-duplex mode.

The Am7970 Expander will take compressed data from the Code Buffer for processing and the resulting image data will be loaded into the Line Buffer. The source for the expansion process is the Code Buffer and the destination is the Line Buffer.

For certain images (such as half-tones), the compressed data representing a line may be longer than the original line of the image. The Am7970 checks for this condition after compressing a line to alert the host processor of this situation via an Interrupt and Status Bit.

An End-of-Line (EOL) code may delimit each compressed line according to the CCITT recommendation for Group 3 facsimile apparatus. However, this automatic EOL insertion can be suppressed by appropriate bit settings of the Am7970 (Group 4).

The CCITT recommendation T.4 for Group 3 equipment requires that each coded line be a certain minimum length. Time Fill bits will be added by the CEP to a shorter line when necessary to meet this requirement. The Am7970 contains a Time Fill Register to specify the minimum line lengths (including zero).

Data is vulnerable to modification by transmission errors. When such data is expanded, the resulting image will be different from the original. The Am7970 checks whether the expanded line has the same number of picture elements as required by the specified paper width. If there is a discrepancy, the CPU will be alerted.

INITIALIZATION

The Am7970 has several initialization requirements. Those initialization requirements include:

- The Source Buffer definition,
- Attribute definition for the Source Buffers,
- The Destination Buffer definition,
- The restart conditions definition,
- The operating mode definition,
- The processing mode definition,
- Definition of the compressed data format,
- The minimum transmission time definition,
- The paper width definition,
- The optional function definitions.

The system program should specify certain initial conditions before starting the operation of the Am7970. The Source Buffer definition requires that the residency of the Source Buffer, the starting address of the Source Buffer and the capacity of the Source Buffer be specified. The system program should load "0" into the Compressor Source Control (CSC) bit/ Expander Source Control (ESC) bit in the Compressor Master Control Register (CMCR)/Expander Master Control Register (EMCR) when a Source Buffer that is located in the Main

Memory is required. If a Source Buffer that is located in the Document Store is required, then the system program should load a "1" into the CSC/ESC bit.

The system program should load a starting address into the Compressor Source Address Holding Register (CSAHR)/ Expander Source Address Holding Register (ESAHR) and the Compressor Source Current Address Register (CSCAR)/ Expander Source Current Address Register (ESCAR). Also, the system program should load the length (in bytes) of the Source Buffer into the Compressor Source Count Holding Register (CSCHR)/Expander Source Count Holding Register (ESCHR) and the Compressor Source Working Count Register (CSWCR)/Expander Source Working Count Register (ESWCR). The length of the Source Buffer has additional requirements that are discussed under the specific Source Register sections. The system program should adhere to these recommendations. The system program must set the Source Attribute (CSA/ESA) bit in the CPR/EPR Register when the CEP will process a new page. More detailed information is discussed in the sections on the CPR Register and the EPR Register.

The Destination Buffer definition requires that the residency of the Destination Buffer, the starting address of the Destination Buffer, and the capacity of the Destination Buffer be specified. The system program should load "0" into the Compressor Destination Control (CDC) bit/Expander Destination Control (EDC) bit in the CMCR/EMCR register when the Destination Buffer is located in the Main Memory. If the Destination Buffer is located in the Document Store, the system program should load "1" into the CDC/EDC bit. The system program should load the starting address into the Compressor Destination Address Holding Register (CDAHR)/Expander Destination Address Holding Register (EDAHR) and Compressor Destination Current Address Register (CDCAR)/Expander Destination Current Address Register (EDCAR). Also, the system program should load the negative two's complement of the length (in bytes) of the Destination Buffer into the Compressor Destination Count Holding Register (CDCHR)/Expander Destination Count Holding Register (EDCHR) and the Compressor Destination Working Count Register (CDWCR)/Expander Destination Working Count Register (EDWCR). The length of the Destination Buffer has some conditions that are are discussed in the specific sections which cover the Destination Registers and the system program should adhere to those recommendations.

The system program should load any Restart information into the Compressor Restart Control Register (CRCR)/Expander Restart Control Register (ERCR). The system program should follow the recommendations of the Restart Control Register section. The system program must specify a compressed data format by the Data Format Control (DFC) bits in the Compressor Parameter Register (CPR). This is discussed in more detail in the description of the CPR Register.

The system program should load the operating mode into the Compressor Master Control Register (CMCR)/Expander Master Control Register (EMCR). The system program should follow the recommendations of the Mode Control section under the Master Control Register description. Also the system program should load the operation control into the CMCR/EMCR, the minimum transmission condition into the Time Fill Register (TFLR) and the horizontal resolution into the Compressor Page Width Register (CPWR)/Expander Page Width Register (EPWR).

The definitions consist of eleven different functions. These functions are called:

- Optional Auto-EOL
- Optional byte boundary control
- Wraparound
- Express
- Top Margin
- Left Margin
- Right Margin
- Bottom Margin
- K-Parameter
- G-Parameter
- Interrupt Enable

The system program should load the Auto-EOL option into the AEOL bit in the Compressor Parameter Register (CPR). If the automatic insertion of an EOL code is required, the system program should load "0" into the AEOL bit. If this bit is "1," such automatic insertion of EOL is suppressed.

The system program should specify the Wraparound, Express, and the Top, Left, and Right Margin options by loading the corresponding registers. These optional functions are described in detail in their individual sections. The system program should adhere to those recommendations.

The system program should specify the K-Parameter in the K-Register when Two-Dimensional processing is required. If the granularity option is required, the system program should specify the G-Parameter in the Expander Parameter Register. The system program should load "1" into the Compressor Interrupt Enable (CIE) bit in the Compressor Master Control Register (CMCR)/Expander Interrupt Enable (EIE) bit in the Expander Master Control Register (EMCR) if an interrupt request is required upon CEP termination. If an interrupt request is not required, the system program should load "0" into the CIE/EIE bit of the CMCR/EMCR register.

The system program should load "0" into the EOL bit in the Expander Parameter Register (EPR) when data with attached EOL is going to be received. If the data that is to be received contains no EOL codes, the system program should load "1" into the EOL bit in the EPR register.

START PROCESSING PROCEDURES

The Am7970 has two operating configurations. The Expander and the Compressor may be operated simultaneously in a full-duplex mode, or the Expander or the Compressor may be operated without the other processor (half-duplex). The system program can initiate the full-duplex mode by loading "1s" into the GO bits of the Compressor Master Control Register (CMCR) and the Expander Master Control Register (EMCR). If the half-duplex operation is required, the system program should load a "1" into the GO bit in the Compressor Master Control Register or the system program should load a "1" into the GO bit in the Expander Master Control Register. Further, the CEP has two different types of starting conditions: resumable operation and non-resumable operation. If resumable operation is required, the system program must not

issue a Reset operation via the appropriate Master Control Register or assert the RESET input. On the other hand, the system program must specify a Reset operation before starting its processing when no resumable operation is required; for example, the CEP is going to process a new page.

HOW TO USE THE STATUS REGISTERS

The CEP has three status registers: the Master Status Register (MSR), the Compressor Status Register (CSR), and the Expander Status Register (ESR). Bits 6 and 7 (EBY, CBY) in the MSR register provide both the Compressor and Expander general information to the CPU. These bits are known as the Expander Busy (EBY) bit and the Compressor Busy (CBY) bit. The system program should test Bits 6 and 7 if the MSR register general status information is required. If the system program requires detailed status information, then the system program should test the Compressor Status Register (CSR) or the Expander Status Register (ESR) directly. Bits 0 to 3 (EXT, ECD) in the Master Status Register (MSR) indicate the uncompressed mode status of the Expander. Bit 4 in the Master Status Register (MSR) indicates that the Expander detected an End-of-Page (EOP) code. If interrupts have not been enabled, the system program should periodically poll Bit 6 (EBY) and Bit 7 (CBY) in the MSR register. If the system program is enabled to respond to an interrupt, it should test the EBY bit and the CBY bit in the Master Status Register (MSR) when a CEP interrupt occurs.

INTERRUPT HANDLING

The Am7970 will drive its interrupt line (INTR) High when the Compressor Busy (CBY) bit in the Compressor Status Register (CSR) or the Expander Busy (EBY) bit in the Expander Status Register (ESR) changes from "1" to "0" while the Compressor Interrupt Enable (CIE) bit in the Compressor Master Control Register (CMCR) or the Expander Interrupt Enable (EIE) bit in the Expander Master Control Register (EMCR) has been set to "1." The INTR line will remain High until the Master Status Register (MSR) has been accessed by the system program. The system program may test the MSR register to distinguish Compressor interrupts from Expander interrupts. The system program should isolate the cause of the interrupt by reading the appropriate status register (CSR or ESR). The system program may then execute its interrupt service routine to respond to the interrupt.

STOPPING THE CEP

The CEP may be terminated by writing to the CEP's Master Control Register while the CEP is busy, or by issuing a hardware RESET. The soft abort will occur if the system program loads new information into the Master Control Register (Compressor or Expander) while the CEP is busy. If the system program is immediately required to stop, the system program should assert the RESET input of the Am7970. This is called a hardware stop. If the system program executes a hardware stop, the CEP will not save the current status. If the system program executes a software stop, the CEP will terminate its operation and keep the Compressor Busy and the New Operation Attempted (COA) bit or Expander Busy and New Operation Attempted (EOA) bit; however, this is not a resumable operation.

COMPRESSOR ERROR RECOVERY PROCEDURES

The Compressor will detect several error conditions: a premature source overflow, a premature destination overflow or an illegal command. An error condition will also be detected if a new command is attempted while the Compressor is busy or if negative compression occurs. A pre-mature source or destination overflow will be illuminated by the Wraparound Incomplete (WPI) bit or the Line Processing Incomplete (LPI) bit of the Compressor Status Register (CSR).

The error recovery procedure for an LPI error must include the redefinition of the Source Buffer or the Destination Buffer as follows:

1) Premature Source Overflow

   New CSCHR = N * Hr * Apw/8

   New CSWCR = 2's complement of new CSCHR − old CSCHR

   New CSCAR = CSLSR, new CDCAR = CDLSR

   CRCR = All one (X'FF')

   New CDWCR = old CDWCR = (CDCAR − CDLSR)

2) Premature Destination Overflow

   New CDCHR = N * Hr * Apw/8 (See Section 12)

   New CDWCR = 2's complement of new CDCHR − old CDCHR

   New CDCAR = CDLSR, new CSCAR = CSLSR

   CRCR = All one (X'FF')

   New CSWCR = old CSWCR − (CSCAR − CSLSR)

The error recovery procedure for a WPI error without an LPI error may include restarting the Source Buffer or the Destination Buffer. If an illegal command is detected, the system program should load Continue Operation into the CEP or reissue a new command to the CEP when the Compressor or Expander Busy and New Operation Attempted error (COA or EOA) is detected. If negative compression is detected, the system program should load a Continue Operation command into the CEP or the system program should replace the line of "negatively compressed" data with transparent data.

EXPANDER ERROR RECOVERY PROCEDURES

The Expander will detect several error conditions: a premature source overflow, a premature destination overflow, an illegal command, an Expander Busy and New Operation Attempted error, a data error or an undefined extension code. A premature source or destination overflow will be illuminated by the WPI bit and the LPI bit of the Expander Status Register (ESR). The error recovery procedure for a premature overflow requires that the Source Buffer and the Destination Buffer be redefined as follows:

1) Premature Source Overflow and LPI without WPI

   New ESCHR = N * Hr * Apw/8

   New ESWCR = 2's complement of new ESCHR − old ESCHR

   ERCR = All one (X'FF')

New EDWCR = old EDWCR – (EDCAR – EDLSR)

New ESCAR = ESLSR, new EDCAR = EDLSR

2) Premature Destination Overflow and LPI without WPI

New EDCHR = N * Hr * Apw/8

New EDWCR = 2's complement of new EDCHR – old EDCHR

New EDCAR = EDLSR, new ESCAR = ESLSR

ERCR = All one (X'FF')

New ESWCR = old ESWCR – (ESCAR – ESLSR)

If the system program detects a premature overflow and the WPI error bit is set without the LPI error bit being set, the system program should restart the Source or Destination Buffer. If an illegal command is detected, the system program should load Continue Operation into the CEP or reissue a new command to the CEP when the Expander Busy and New Operation Attempted (EOA) error is detected.

If a data error is detected, the system program should replace the error line with a copy of the previous line as follows:

New ESCAR = EDLSR N * Hr * Apw/8

N = EWR + 1

New ESWCR = 2's complement N * Hr * Apw/8

New EDCAR = EDLSR

New EDWCR = EDWCR – (EDCAR – EDLSR)

ERCR = All one (X'FF')

EM0 and EM1 = 0 (Transparent Mode)

CPU ACCESS OPERATIONS

The procedure by which a CEP register is accessed by the CPU is as follows:

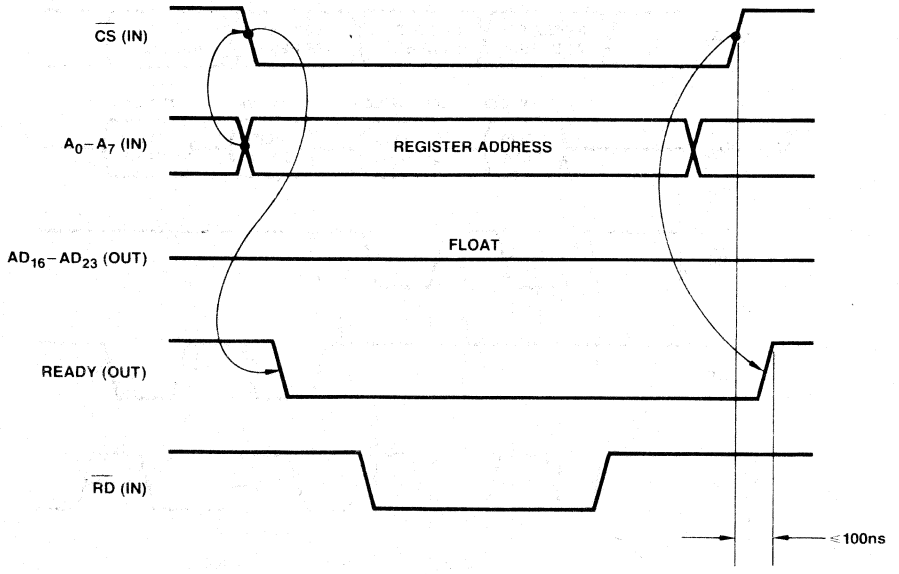
Read Access Operation

1) The CPU places an address on the CPU address lines A_0–A_7 that specifies the intended register.
2) The CPU address decoder (external to the CEP) drives the $\overline{CS}$ input Low.
3) The CEP will drive the READY output Low.
4) The CPU will drive the $\overline{RD}$ input Low.
5) The CEP READY output will be driven High when the CEP register data is available.
6) AD_{16}–AD_{23} will be driven by the CEP with valid data.
7) The CPU will drive the CEP $\overline{RD}$ input High.
8) The $\overline{CS}$ input will be driven High. Further read accesses can be initiated by executing Step 1.

Note: If Step 7 precedes Step 5, the read access will be aborted by the CEP. If the read access is aborted, READY will be driven High and AD_{16}–AD_{23} will float. Further read accesses can be initiated by executing Step 1.

Write Access Operation

1) The CPU places an address on the CPU address lines A_0–A_7 that specifies the intended register.
2) The CPU address decoder (external to the CEP) drives the $\overline{CS}$ input Low.



05777B-45

Figure 45. CPU Program Read Access Abort Timing

05777B

Refer to page 7-1 for Essential Information on Military Devices

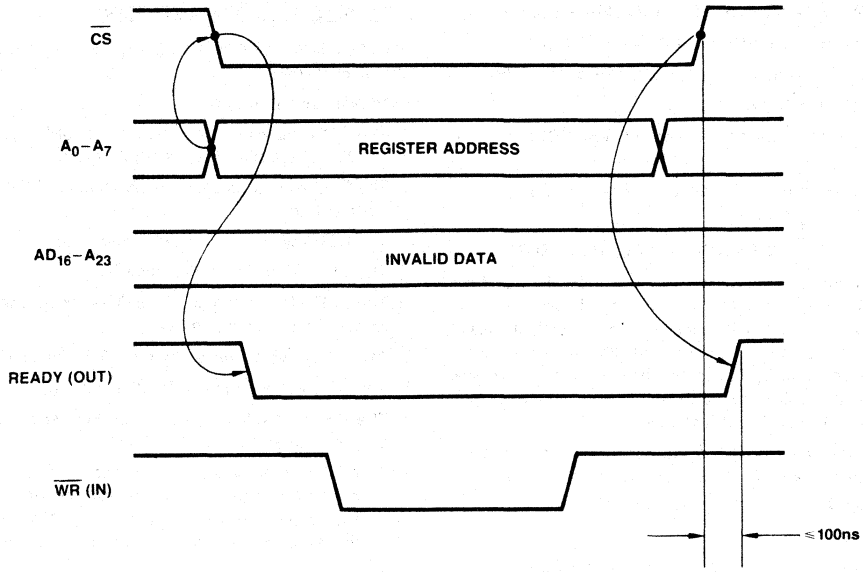


Figure 46. CPU Program Write Access Abort Timing

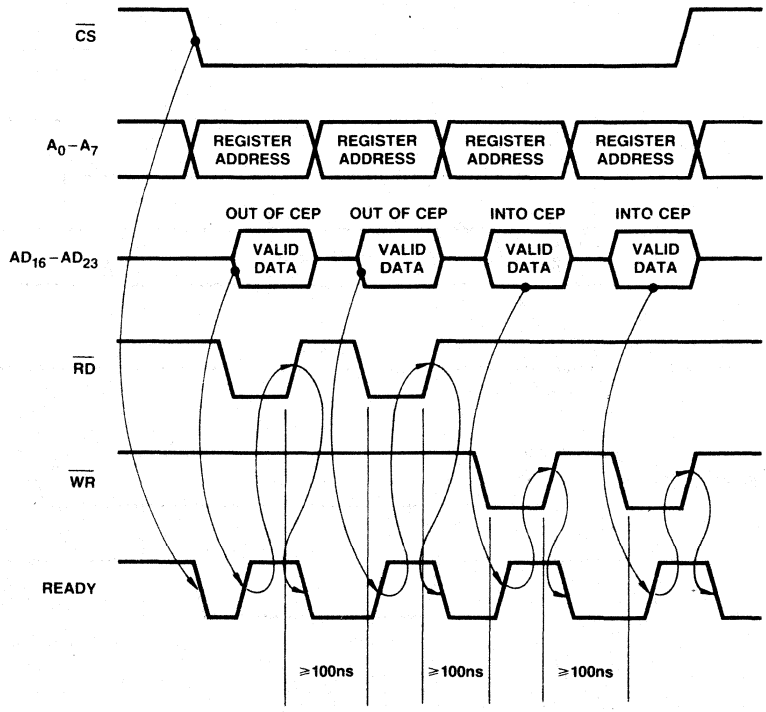


Figure 47. CPU Block I/O Transaction Timing (CEP is slave)

05777B

Refer to page 7-1 for Essential Information on Military Devices

3) The CEP will drive the READY output Low.

4) The CPU will drive the $\overline{WR}$ input Low.

5) AD_{16}–AD_{23} will be driven by the CPU with valid data.

6) The CEP READY output will be driven High after data has been loaded into the appropriate register.

7) The CPU will drive the CEP $\overline{WR}$ input High.

8) The $\overline{CS}$ input will be driven High. Further write accesses can be initiated by executing Step 1.

Note: If Step 7 precedes Step 6, the write access will be aborted. If the write access is aborted, READY will be driven High. The contents of the specified register will not be altered. Further write accesses can be initiated by executing Step 1.

DMA OPERATION

The procedure by which the CEP executes a DMA operation is as follows:

Read Access Operation

1) The CEP drives the HRQ output High.

2) The CPU drives the HLDA input High.

3) The CEP drives the ALE output High and places a memory address (24 bits) on the CPU bus during CEP state T1. The address will be valid during the High to Low transition of the ALE output (CEP state T2, falling edge).

4) The CEP drives the $\overline{RD}$ output Low (CEP state T2, rising edge).

5) The state (High or Low) of the READY input is sampled by the rising edge of T2.

6) If the READY input is Low, a Wait State will be inserted. The READY input should become High when the memory location becomes available (CEP Wait State, rising edge).

7) If the READY input is High, then AD_{16}–AD_{23} must be driven with valid data (from the indicated memory location). The rising edge of T3 samples the data on AD_{16}–AD_{23}.

8) The CEP drives the $\overline{RD}$ output High (CEP state T3, rising edge).

9) The CEP drives the HRQ output Low.

10) The CPU drives the HLDA input Low.

Note: The CEP repeats Steps 1 through 8 until no further processing is required. The HRQ output will then be driven Low for at least two clock cycles. If additional bus transactions are required by the CEP, the CEP will drive the HRQ output High after a minimum of two clocks have elapsed (Step 1).

Write Access Operation

1) The CEP drives the HRQ output High.

2) The CPU drives the HLDA input High.

3) The CEP drives the ALE output High and places a memory address (24 bits) on the CPU bus (state T1, falling edge). The address will be valid during the High to Low transition of the ALE output.

4) AD_{16}–AD_{23} will be driven by the CEP with valid data (CEP state T2, falling edge).

5) The CEP drives the $\overline{WR}$ output Low (CEP state T2, rising edge).

6) The state (High or Low) of the READY input is sampled by the CEP at the state T2 rising edge.

7) If the READY input is Low, a Wait State will be inserted. The READY input should be driven High after data has been loaded into the appropriate memory location (CEP Wait State, rising edge).

8) The CEP drives the $\overline{WR}$ output High (CEP state T3, rising edge).

9) The CEP drives the HRQ output Low.

10) The CPU drives the HLDA input Low.

Note: The CEP repeats Steps 1 through 8 until no further processing is required. The HRQ output will then be driven Low for two clock cycles. If additional bus transactions are required by the CEP, the CEP will drive the HRQ output High (Step 1).

DOCUMENT STORE BUS OPERATION

The procedure by which the CEP executes a Document Store memory operation is as follows:

Read Access Operation

1) The CEP places a Document Store address (24 bits) on the Document Store bus (CEP state T1). The address will be valid during the High to Low transition of the DALE output (CEP state T2, falling edge).

2) The CEP drives the $\overline{DRD}$ output Low (CEP state DT2, rising edge).

3) The state (High or Low) of the READY input is sampled by DT2, rising edge.

4) If the DREADY input is Low, a Wait State will be inserted. The READY input should become High when the Document Store location becomes available (CEP Wait State, rising edge).

5) If the DREADY input is High, then DAD_{16}–DAD_{23} must be driven with valid data from the indicated Document Store location (CEP state DT3, falling edge).

6) The CEP drives the $\overline{DRD}$ output High (CEP state T3, rising edge).

Write Access Operation

1) The CEP places a Document Store address (24 bits) on the Document Store bus (CEP state T1, falling edge). The address will be valid during the High to Low transition of the DALE output (CEP state T2, falling edge).

2) DAD_{16}–DAD_{23} will be driven by the CEP with valid data (T2 falling edge).

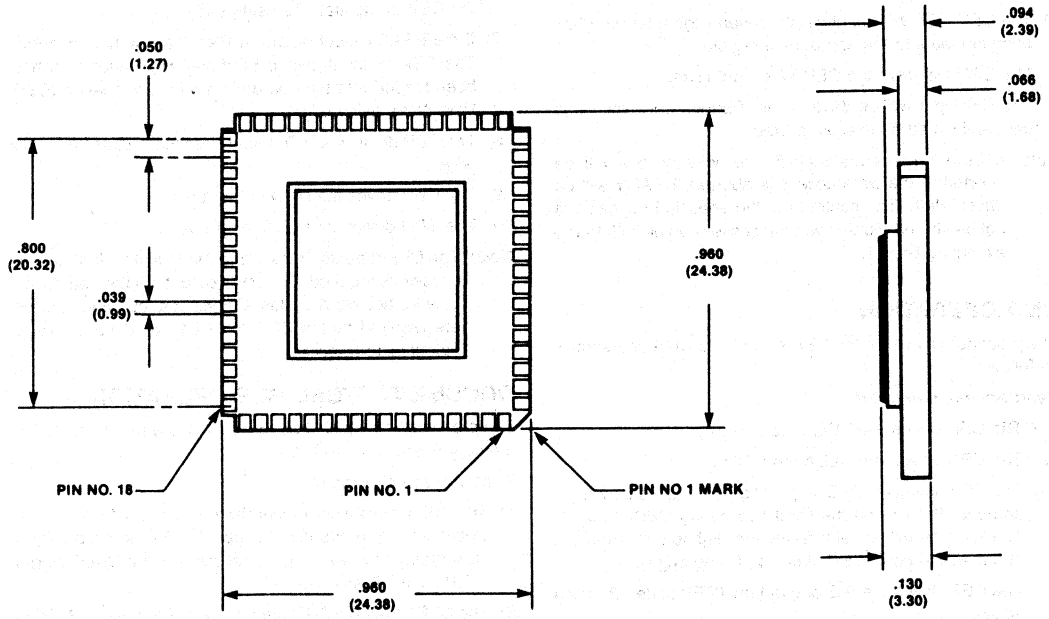
3) The CEP drives the $\overline{DWR}$ output Low (CEP state T2, rising edge).

4) The state (High or Low) of the DREADY input is sampled by the T2 rising edge.

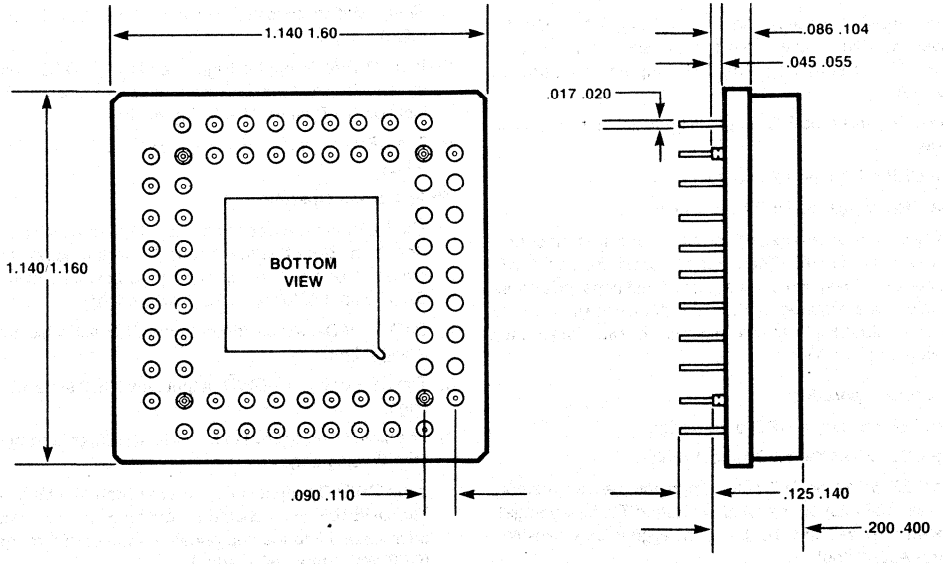
5) If the DREADY input is Low, a Wait State will be inserted. The DREADY input should be driven High after data has been loaded into the appropriate Document Store location (CEP Wait State, rising edge).

6) The CEP drives the $\overline{DWR}$ output High (CEP state DT3, rising edge).

PACKAGE SIZE



LCC PACKAGE



PGA PACKAGE

05777B

Electrical Specifications
5 MHz Am7970 CEP

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	−65°C to +150°C
Voltage on Any Pin With Respect to Ground	−1.0V to +7.0V
Power Dissipation	

DC CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5V±10%)

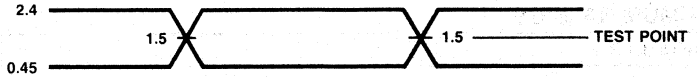
Symbol	Parameter	Test Conditions	Min	Max	Unit
VIL	Input Low Voltage		−0.5	0.8	V
VIH	Input High Voltage		2.0	V_{CC} + 0.5	V
VOL	Output Low Voltage	IOL = 3.2 mA		04	V
VOH	Output High Voltage	IOH = 400 μA	2.4		V
ICC	Power Supply Current	Ta = 25C			mA
ILL	Input Leakage Current	0V Vin V_{CC}		±10	μA
ILO	Output Leakage Current	0.45V V_{OUT} V_{CC}		±10	μA
VCL	Clock Input Low Voltage		−0.5	+0.8	V
VCH	Clock Input High Voltage		2.0	V_{CC}+0.5	V
CIN	Capacitance of Input Buffer (All input except A0–15, AD16–23, DA0–16, DAD16–23)	fc = 1 MHz		10.0	PF
CIO	Capacitance of I/O Buffer (A0–15, AD16–23, Da0–15, DAD16–23)	fc = 1 MHz		10.0	PF
CO	Capacitance of Output Buffers	fc = 1 MHz		10.0	PF

2

05777B

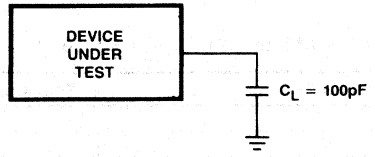
Refer to page 7-1 for Essential Information on Military Devices

INPUT/OUTPUT AC Testing: Input, Output Waveform



AC Testing: Inputs are driven at 2.4V for logic "1" and 0.45V for a logic "0" timing measurements are made at 1.5V for both a logic "1" and "0"

AC Testing Load Circuit



C_L includes Jig Capacitance

05777B

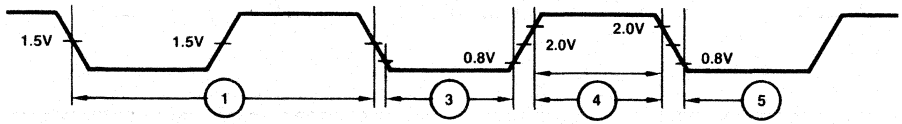
SWITCHING CHARACTERISTICS (TA = 0°C to 70°C, V_{CC} = 5V ±10%)
TIMING REQUIREMENTS

#	Parameter	Description	Test Conditions	Min	Typ	Max	Unit
1	TCLCL	CLK Cycle Period	From 0.8V to 0.8V	200.0		2000.0	ns
2	TCHCL	CLK HIGH Time	From 2.0V to 2.0V	85.0			ns
3	TCLCH	CLK LOW Time	From 0.8V to 0.8V	85.0			ns
4	TCH1CH2	CLK Rise Time	From 0.8V to 2.0V			15.0	ns
5	TCL2CL1	CLK Fall Time	From 2.0V to 0.8V			15.0	ns
6	TRHVH	Power Supply HIGH to RESET LOW Time		4TCLCL			ns
7	TRHRL	RESET HIGH TIME		4TCLCL			ns
8	TRLSL	RESET LOW to First $\overline{CS}$		2TCLCL			ns
9	THAHCH	HLDA RE Set-up Time		30.0			ns
10	THALCH	HLDA FE Set-up Time		30.0			ns
11	TAVSL	Address Valid to $\overline{CS}$ LOW		20.0			ns
12	TSLRDL	$\overline{CS}$ LOW to $\overline{RD}$ FE Set-up Time		20.0			ns
13	TRDHSH	$\overline{RD}$ HIGH to Next Address Active		0.0			ns
14	TSLWRL	$\overline{CS}$ LOW to $\overline{WR}$ LOW		30.0			ns
15	TDVRYH	Data Valid to READY RE		30.0			ns
15a	TDVWH	Data Valid to $\overline{WR}$ High		30.0			ns
16	TWRHDV	DATA Hold Time		20.0			ns
17	TWRHSH	$\overline{WR}$ to Address Active		0.0			ns
18	TRYLCH	READY FE Set-up Time		60.0			ns
19	TCHDXY	READY Hold Time		20.0			ns
20	TRYHCH	READY		20.0			ns
21	TDVCH	DATA IN Set-Up Time		30.0			ns
22	TCHDX	DATA IN Hold Time		10.0			ns

Note: Switching characteristics are targetted numbers and are subject to change without notice.

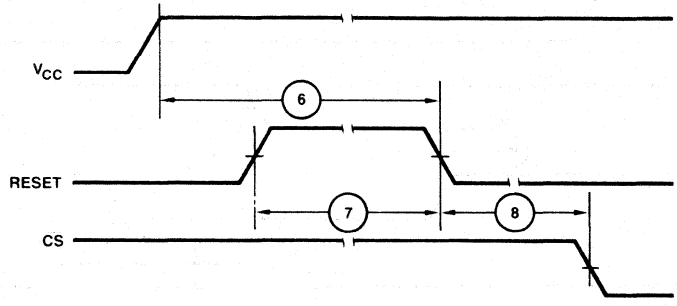
05777B

Refer to page 7-1 for Essential Information on Military Devices



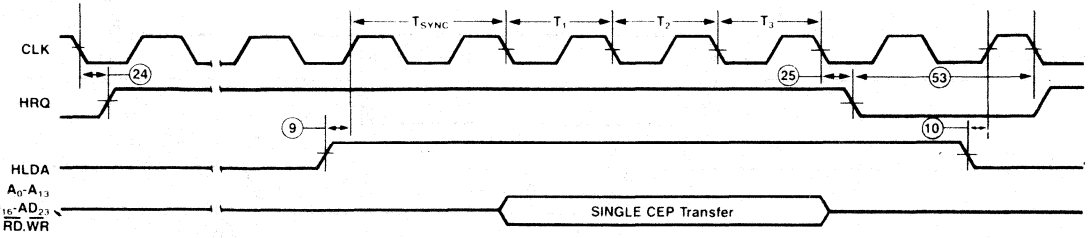
05777B

Timing Diagram 1. Clock Timing



05777B

Timing Diagram 2. RESET Timing

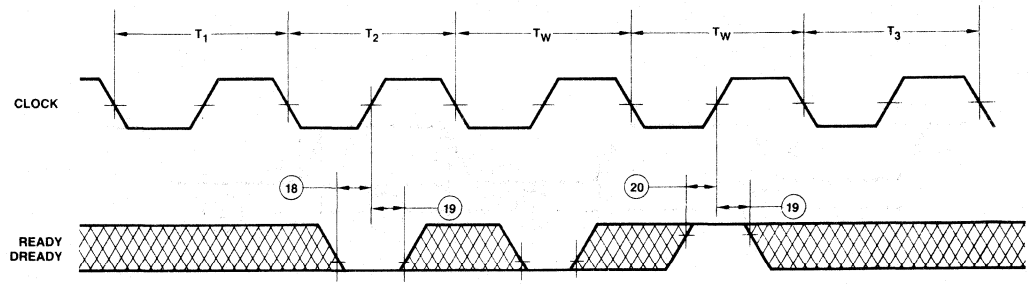


05777B

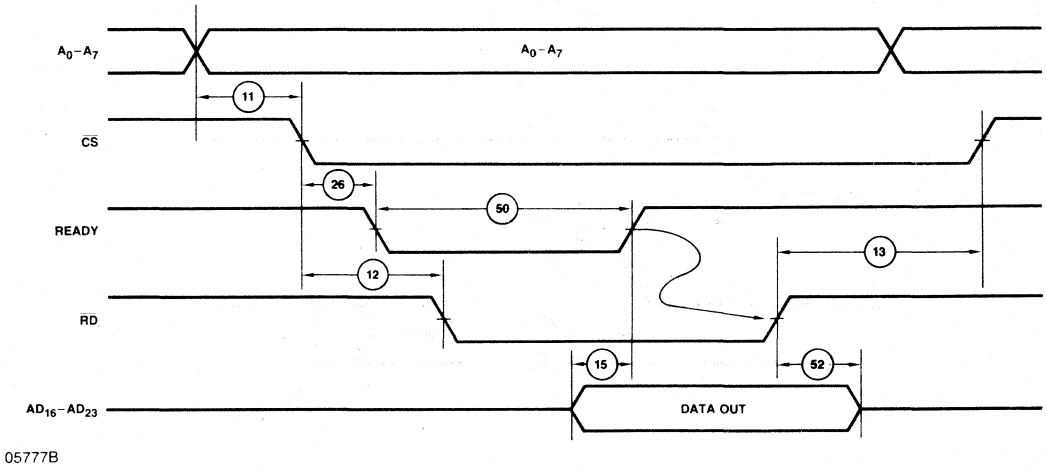
Timing Diagram 3. Bus Exchange Timing

05777B

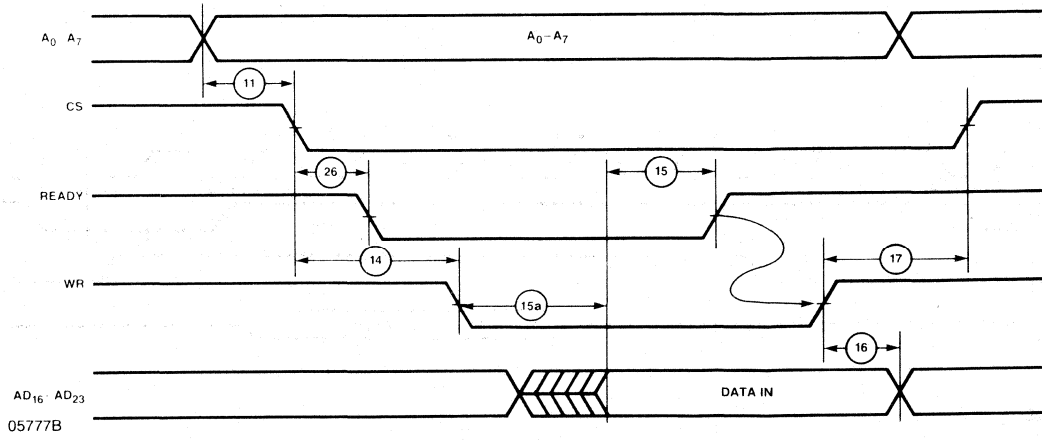
Refer to page 7-1 for Essential Information on Military Devices



Timing Diagram 4. READY, DREADY Input Timing



Timing Diagram 5. CPU Program Read Timing



Timing Diagram 6. CPU Program Write Timing

05777B

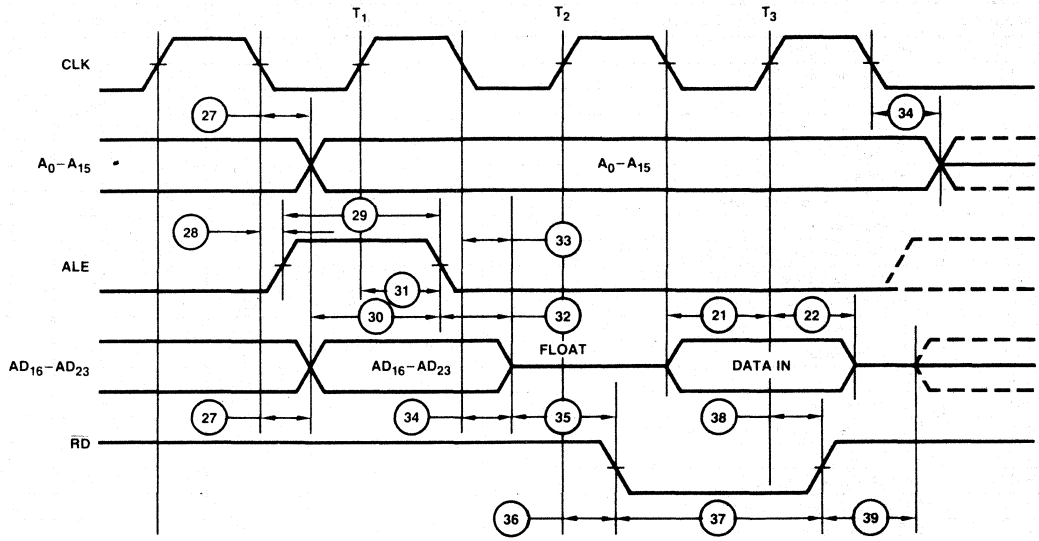
Refer to page 7-1 for Essential Information on Military Devices

SWITCHING CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5V±10%)

Timing Responses

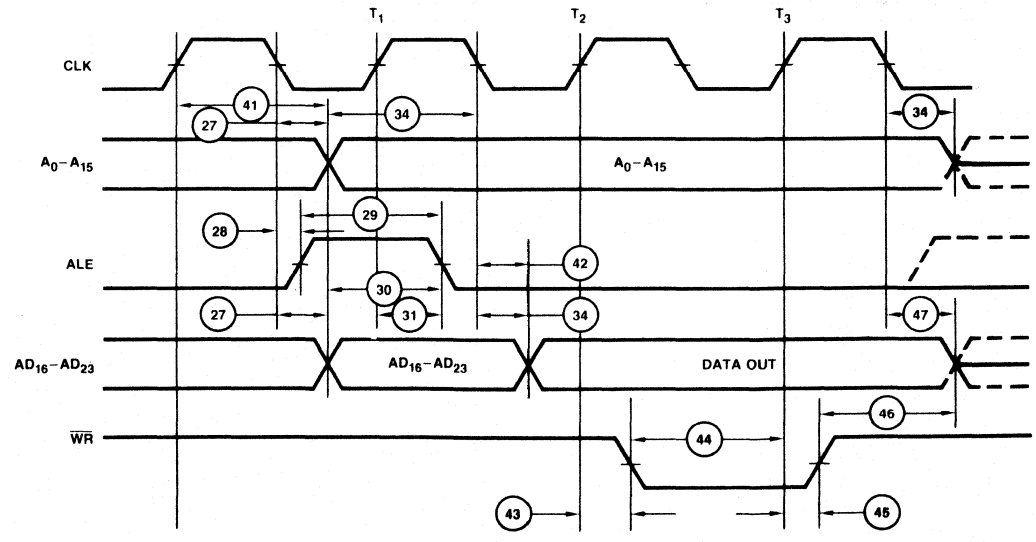
#	Parameter	Description	Test Conditions	Min	Typ	Max	Unit
24	TCLHRH	HRQ Active Delay	CL = 20–100PF for all 7970 Outputs (In addition to 7970 selfload)			80.0	ns
25	TCLHRL	HRQ Inactive Delay				85.0	ns
26	TSLRYL	READY Active Delay				80.0	ns
27	TCLAV	Address Valid Delay		10.0		80.0	ns
28	TCLLH	ALE Active Delay				65.0	ns
29	TLHLL	ALE Width		70.0			ns
30	TAVAL	Address Valid to ALE LOW		55.0			ns
31	TCHLL	ALE Inactive Delay				65.0	ns
32	TLLAX	Address Hold Time to ALE Inactive (A16–A23)		50.0			ns
33	TCLAZ	Address Float Delay From Clock FE				50.0	ns
34	TCLAX	Address Hold Time (A0–A15)		15.0			ns
35	TAZRL	Address Float to $\overline{RD}$ Active		0.0			ns
36	TCHRL	$\overline{RD}$ Active Delay		10.0		50.0	ns
37	TRLRH	$\overline{RD}$ Width		TCLCL-40			ns
38	TCHRH	$\overline{RD}$ Inactive Delay		10.0		50.0	ns
39	TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL			ns
40	TCLDX	DATA Hold Time		10.0			ns
41	TCHCTV	Control Active Delay 2		10.0		110.0	ns
42	TCLDV	DATA Valid Delay From Clock FE				110.0	ns
43	TCHWL	$\overline{WR}$ Active Delay		10.0		50.0	ns
44	TWLWH	$\overline{WR}$ Width		TCLCL-40			ns
45	TCHWH	$\overline{WR}$ Inactive Delay		0.0		50.0	ns
46	TWHDX	DATA Hold Time After $\overline{WR}$		60.0			ns
47	TCHDX	DATA Hold Time		10.0			ns
48	TOLOH	Output Rise Time	From 0.8V to 2.0V			15.0	ns
49	TOHOL	Output Fall Time	From 2.0V to 0.8V			10.0	ns
50	TRYW	READY Width		2TCLCL-75			ns
51	TDVRYH	DATA Valid to READY HIGH		30.0		90.0	ns
52	TRDHDV	DATA OUT Not Valid Delay		0.0		50.0	ns
53	THRLHR	HRQ Low To HRQ High		2TCLCL			ns

Note: Switching characteristics are targetted numbers and are subject to change without notice.



05777B

Timing Diagram 7. DMA Read Operation

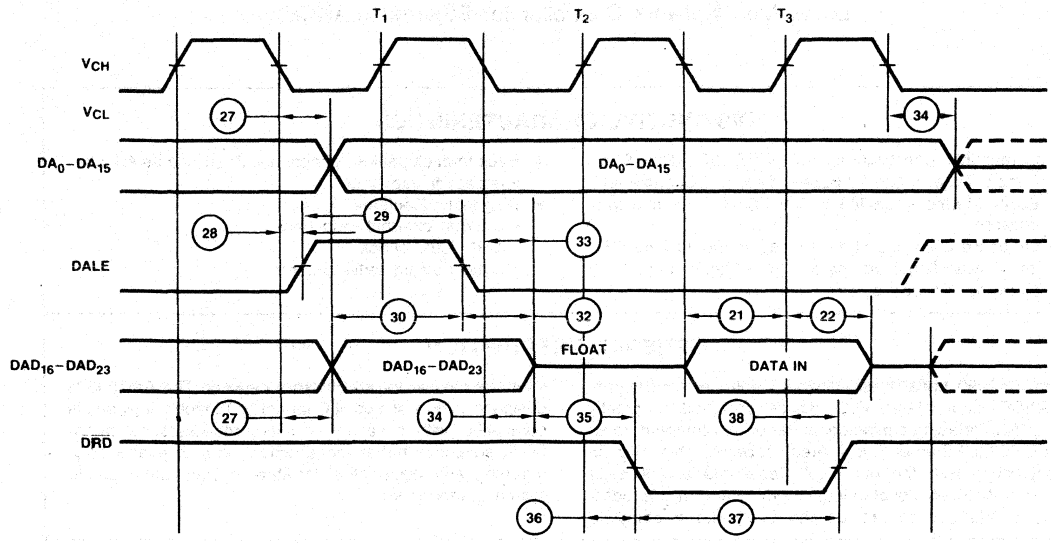


05777B

Timing Diagram 8. DMA Write Operation

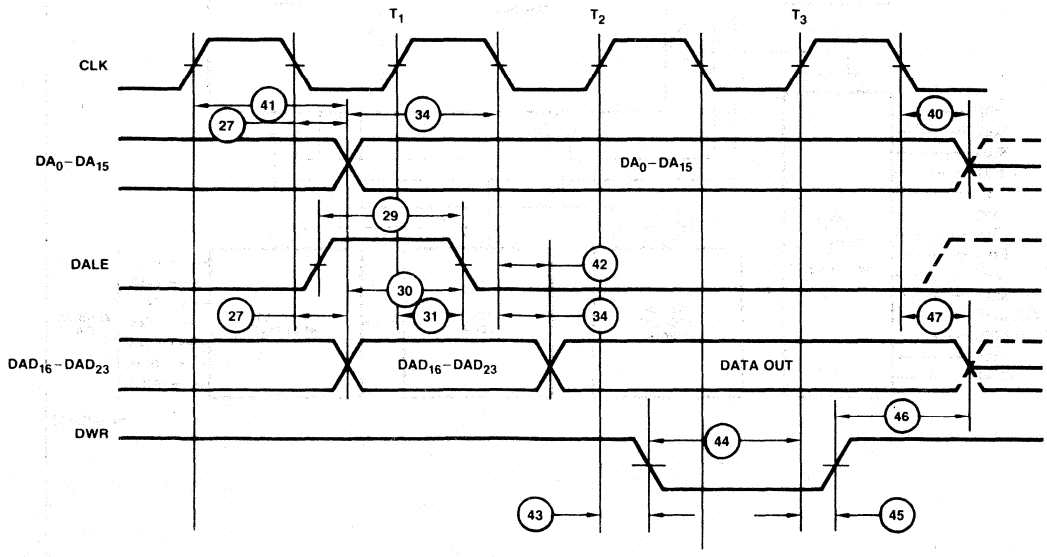
05777B

Refer to page 7-1 for Essential Information on Military Devices



05777B

Timing Diagram 9. Document Store Bus Read Operation



05777B

Timing Diagram 10. Document Store Bus Write Operation

05777B

Refer to page 7-1 for Essential Information on Military Devices

Am7990

Local Area Network Controller for Ethernet (LANCE)

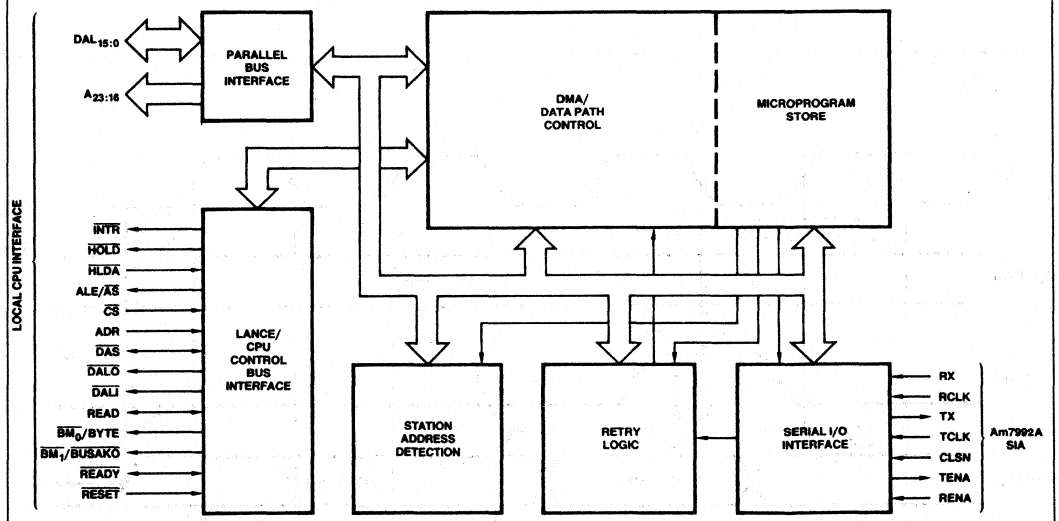
DISTINCTIVE CHARACTERISTICS

- Compatible with Ethernet and IEEE-802.3 Rev D (10 Base 5 Type A, and 10 Base 2 Type B, "Cheapernet")
- Easily interfaced to 8086, 68000, Z8000, LSI-II microprocessors
- On-board DMA and buffer management, 48 byte FIFO
- 24-bit wide linear addressing (Bus Master Mode)
- Network and packet error reporting
- Back-to-back packet reception with as little as 4.1 μsec interpacket gap time
- Diagnostic Routines
  - Internal/external loop back
  - CRC logic check
  - Time domain reflectometer

GENERAL DESCRIPTION

The Am7990 Local Area Network Controller for Ethernet (LANCE) is a 48-pin VLSI device designed to greatly simplify interfacing a microcomputer or minicomputer to an IEEE-802.3/Ethernet Local Area Network. This chip, in conjunction with the Am7992A Serial Interface Adapter (SIA) and closely coupled local memory and microprocessor, is intended to provide the user with a complete interface module for an Ethernet network. The Am7990 is designed using a scaled N-Channel MOS technology and is compatible with a variety of microprocessors. On-board DMA, advanced buffer management, and extensive error reporting and diagnostics facilitate design and improve system performance.

BLOCK DIAGRAM



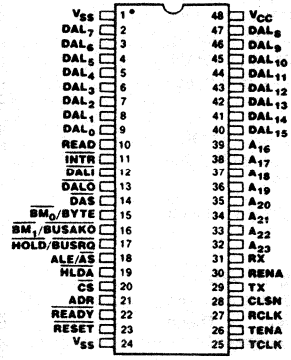
BD002062

05698A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

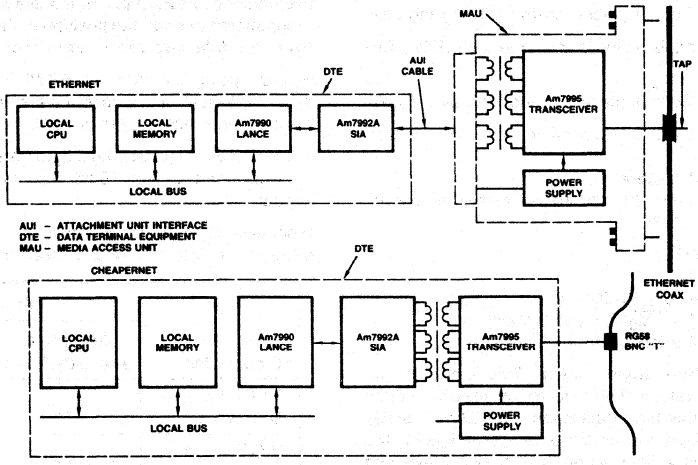
D-48-2



CD001450

Note: Pin 1 is marked for orientation

TYPICAL ETHERNET/CHAEPERNET NODE



AF000473

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am7990 **D** **C** **B**

Device Type

- Screening Option
  - B = Burn-in
  - Blank = Standard processing
- Temperature
  - C = Commercial 0°C to 70°C
- Package
  - D = SIDE BRAZED
  - P = plastic
  - L = leadless chip carrier

Valid Combinations	
Am7990	DC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

DAL_{00} – DAL_{15} **Data/Address Lines (Input/Output 3-State)**
The time multiplexed Address/Data bus. During the address portion of a memory transfer, DAL_{00} – DAL_{15} contains the lower 16 bits of the memory address. The upper 8 bits of address are contained in A_{16} – A_{23}.

During the data portion of a memory transfer, DAL_{00} – DAL_{15} contains the read or write data, depending on the type of transfer.

The LANCE drives these lines as a Bus Master and as a Bus Slave.

A_{16} – A_{23} **High Order Address Bus (Output 3-State)**
The additional address bits necessary to extend the DAL lines to access a 24-bit address. These lines are driven as a Bus Master only.

READ **(Input/Output 3-State)**
Indicates the type of operation to be performed in the current bus cycle. This signal is an output when the LANCE is a Bus Master.

High – Data is taken off the DAL by the chip.

Low – Data is placed on the DAL by the chip.

The signal is an input when the LANCE is a Bus Slave.

High – Data is placed on the DAL by the chip.

Low – Data is taken off the DAL by the chip.

$\overline{BM_0}$/ BYTE $\overline{BM_1}$/ $\overline{BUSAKO}$ **(Output 3-state)**
Pins 15 and 16 are programmable through bit (00) of CSR_3.

$\overline{BM_0}$, $\overline{BM_1}$

If CSR_3 (00) BCON = 0
PIN 15 = $\overline{BM_0}$ (Output 3-state)
PIN 16 = $\overline{BM_1}$ (Output 3-state)

$\overline{BM_0}$, $\overline{BM_1}$ (Byte Mask). This indicates the byte(s) on the DAL are to be read or written during this bus transaction. The LANCE drives these lines only as a Bus Master. It ignores the Byte Mask lines when it is a Bus Slave and assumes word transfers.

Byte selection using Byte Mask is done as described by the following table.

$\overline{BM_1}$	$\overline{BM_0}$	
LOW	LOW	Whole Word
LOW	HIGH	Upper Byte
HIGH	LOW	Lower Byte
HIGH	HIGH	None

BYTE, $\overline{BUSAKO}$

If CSR_3 (00) BCON = 1
PIN 15 = BYTE (Output 3-state)
PIN 16 = $\overline{BUSAKO}$ (Output)

Byte selection may also be done using the BYTE line and DAL_{00} line, latched during the address portion of the bus cycle. The LANCE drives BYTE only as a Bus Master and ignores it when a Bus Slave selection is done (similar to $\overline{BM_0}$, $\overline{BM_1}$).
Byte selection is done as outlined in the following table.

BYTE	DAL_{00}	
LOW	LOW	Whole Word
LOW	HIGH	Illegal Condition
HIGH	LOW	Lower Byte
HIGH	HIGH	Upper Byte

$\overline{BUSAKO}$ is a bus request daisy chain output. If the chip is not requesting the bus and it receives $\overline{HLDA}$, $\overline{BUSAKO}$ will be driven Low. If the LANCE is requesting the bus when it receives $\overline{HLDA}$, $\overline{BUSAKO}$ will remain High.

Byte Swapping

In an effort to be compatible with the variety of 16-bit microprocessors available to the designer, the LANCE may be programmed to swap the position of the upper and lower order bytes on data involved in transfers with the internal FIFO.

Byte swapping is done when BSWP = 1. The most significant byte of the word in this case will appear on DAL lines 7-0 and the least significant byte on DAL lines 15-8.

When BYTE = H (indicating a byte transfer) the table indicates on which part of the 16-bit data bus the actual data will appear.

Whenever byte swap is activated, the only data that is swapped is data traveling to and from the FIFO.

Signal Line	Mode Bits: BSWP = 0 and BCON = 1	Mode Bits: BSWP = 1 and BCON = 1
BYTE = L and DAL_{00} = L	Word	Word
BYTE = L and DAL_{00} = H	Illegal	Illegal
BYTE = H and DAL_{00} = H	Upper Byte	Lower Byte
BYTE = H and DAL_{00} = L	Lower Byte	Upper Byte

$\overline{CS}$ **Chip Select (Input)**
Indicates, when asserted, that the LANCE is the slave device of the data transfer. $\overline{CS}$ must be valid throughout the data portion of the bus cycle. $\overline{CS}$ must not be asserted when $\overline{HLDA}$ is Low.

ADR — **Register Address Port Select (Input)**
When LANCE is slave, ADR indicates which of the two register ports is selected. ADR LOW selects register data port; ADR HIGH selects register address port. ADR must be valid throughout the data portion of the bus cycle and is only used by the LANCE when CS is Low.

ALE/$\overline{AS}$ — **Address Latch Enable (Output 3-State)**
Used to demultiplex the DAL lines and define the address portion of the bus cycle. This I/O pin is programmable through bit (01) of CSR_3.

As ALE (CSR_3 (01), ACON = 0), the signal transitions from a HIGH to a LOW during the address portion of the transfer and remains Low during the data portion. ALE can be used by a Slave device to control a latch on the bus address lines. When ALE is High, the latch is open, and when ALE goes Low, the latch is closed.

As $\overline{AS}$ (CSR_3 (01), ACON = 1), the signal pulses Low during the address portion of the bus transaction. The Low-to-High transition of $\overline{AS}$ can be used by a Slave device to strobe the address into a register.

The LANCE drives the ALE/$\overline{AS}$ line only as a Bus Master.

$\overline{DAS}$ — **Data Strobe (Input/Output 3-State)**
Defines the data portion of the bus transaction. $\overline{DAS}$ is high during the address portion of a bus transaction and low during the data portion. The Low-to-High transition can be used by a Slave device to strobe bus data into a register. $\overline{DAS}$ is driven only as a Bus Master.

$\overline{DALO}$ — **Data/Address Line Out (Output 3-State)**
An external bus transceiver control line. $\overline{DALO}$ is asserted when the LANCE drives the DAL lines. $\overline{DALO}$ will be Low only during the address portion if the transfer is a READ. It will be Low for the entire transfer if the transfer is a WRITE. $\overline{DALO}$ is driven only when LANCE is a Bus Master.

$\overline{DALI}$ — **Data/Address Line In (Output 3-State)**
An external bus transceiver control line. $\overline{DALI}$ is asserted when the LANCE reads from the DAL lines. It will be Low during the data portion of a READ transfer and remain High for the entire transfer if it is a WRITE. $\overline{DALI}$ is driven only when LANCE is a Bus Master.

$\overline{HOLD}$/$\overline{BUSRQ}$ — **Bus Hold Request (Output Open Drain)**
Asserted by the LANCE when it requires access to memory. $\overline{HOLD}$ is held Low for the entire ensuing bus transaction. The function of this pin is programmed through bit (00) of CSR_3. Bit (00) of CSR_3 is cleared when $\overline{RESET}$ is asserted.

When CSR_3 (00) BCON = 0

PIN 17 = $\overline{HOLD}$ (Output Open Drain)

When CSR_3 (00) BCON = 1

PIN 17 = $\overline{BUSRQ}$ (Output Open Drain)

$\overline{BUSRQ}$ will be asserted only if pin 17 is High prior to assertion.

$\overline{HLDA}$ — **Bus Hold Acknowledge (Input)**
A response to $\overline{HOLD}$. When $\overline{HLDA}$ is Low in response to the chip's assertion of $\overline{HOLD}$, the chip is the Bus Master. $\overline{HLDA}$ deasserts upon the deassertion of $\overline{HOLD}$.

$\overline{INTR}$ — **Interrupt (Output Open Drain)**
An attention signal that indicates, when active, that one or more of the following CSR_0 status flags is set: BABL, MERR, MISS, RINT, TINT or IDON. $\overline{INTR}$ is enabled by bit 06 of CSR_0 (INEA = 1). $\overline{INTR}$ remains asserted until the source of Interrupt is removed.

RX — **Receive (Input)**
Receive Input Bit Stream.

TX — **Transmit (Output)**
Transmit Output Bit Stream.

TENA — **Transmit Enable (Output)**
Transmit Output Bit Stream enable. A level asserted with the Transmit Output Bit Stream, TX, to enable the external transmit logic.

RCLK — **Receive Clock (Input)**
A 10MHz square wave synchronized to the Receive data and only active while receiving an Input Bit Stream.

CLSN — **Collision (Input)**
A logical input that indicates that a collision is occurring on the channel.

RENA — **Receive Enable (Input)**
A logical input that indicates the presence of carrier on the channel.

TCLK — **Transmit Clock (Input)**
10MHz clock.

$\overline{READY}$ — **(Input/Output Open Drain)**
When the LANCE is a Bus Master, $\overline{READY}$ is an asynchronous acknowledgement from the bus memory that it will accept data in a WRITE cycle or that it has put data on the DAL lines in a READ cycle.

As a Bus Slave, the LANCE asserts $\overline{READY}$ when it has put data on the DAL lines during a READ cycle or is about to take data off the DAL lines during a write cycle. $\overline{READY}$ is a response to $\overline{DAS}$ and will return High after $\overline{DAS}$ has gone High. $\overline{READY}$ is an input when the LANCE is a Bus Master and an output when the LANCE is a Bus Slave.

$\overline{RESET}$ — **(Input)**
Bus Request Signal. Causes the LANCE to cease operation, clear its internal logic, and enter an Idle state with the stop bit of CSR_0 set.

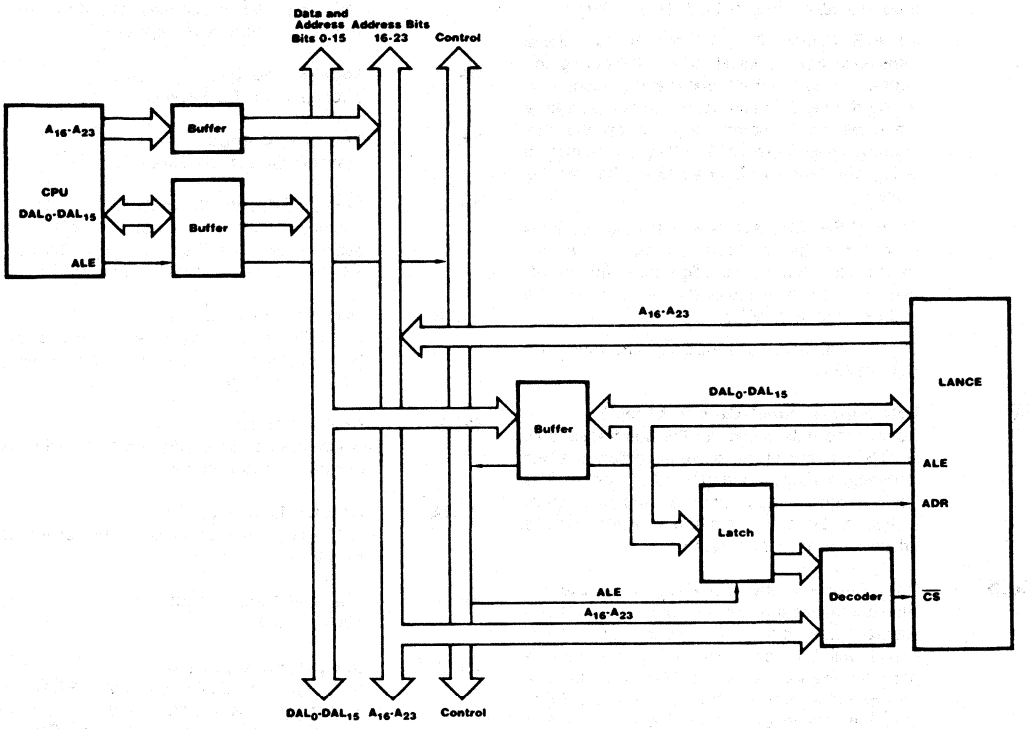
V_{CC} — Power supply pin +5 volts ±5%.

V_{SS} — Ground. Pins 1 and 24 should be connected together externally, as close to the chip as possible.

PRODUCT OVERVIEW

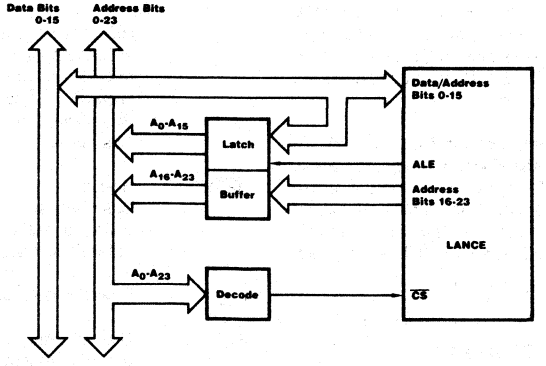
The parallel interface of the Local Area Network Controller for Ethernet (LANCE) has been designed to be "friendly" or easy to interface to a variety of popular 16-bit microprocessors. These microprocessors include the following: Z8000, 8086, 68000 and LSI-11. The LANCE has a 24-bit wide linear address space when it is in the Bus Master Mode, allowing it to DMA directly into the entire address space of the above microprocessors. A programmable mode of operation allows byte addressing in one of two ways: a Byte/Word control signal compatible with the 8086 and Z8000 or an Upper Data Strobe and Lower Data Strobe signal compatible with microprocessors such as the 68000. A programmable polarity on the Address Strobe signal eliminates the need for external logic. The LANCE interfaces with both multiplexed and demultiplexed data busses and features control signals for address/data bus transceivers.

a. Multiplexed Bus



DF000390

b. Demultiplexed Bus



DF000140

Figure 1. LANCE/CPU Interfacing

During initialization, the CPU loads the starting address of the initialization block into two internal control registers. The LANCE has four internal control and status registers ($CSR_{0, 1, 2, 3}$) which are used for various functions, such as the loading of the initialization block address, different programming modes and status conditions. The host processor communicates with the LANCE during the initialization phase for demand transmission and periodically to read the status bits following interrupts. All other transfers to and from the memory are handled as DMA under microword control.

Interrupts to the microprocessor are generated by the LANCE upon: 1) completion of its initialization routine, 2) the reception of a packet, 3) the transmission of a packet, 4) transmitter timeout error, 5) a missed packet and 6) memory error.

The cause of the interrupt is ascertained by reading CSR_0. Bit (06) of CSR_0, (INEA), enables or disables interrupts to the microprocessor. In systems where polling is used in place of interrupts, bit (07) of CSR_0, (INTR), indicates an interrupt condition.

The basic operation of the LANCE consists of two distinct modes: transmit and receive. In the transmit mode, the LANCE chip directly accesses data (in a transmit buffer) in memory. It prefaces the data with a preamble, sync pattern, and calculates and appends a 32-bit CRC. This packet is then ready for transmission to the Am7991A SIA. On transmission, the first byte of data loads into the 48-byte FIFO. The LANCE then begins to transmit preamble while simultaneously loading the rest of the packet into FIFO for transmission.

In the receive mode, packets are sent via the SIA to the LANCE. The packets are loaded into the 48-byte FIFO for preparation of automatic downloading into buffer memory. A CRC is calculated and compared with the CRC appended to the data packet. If the calculated CRC checksum doesn't agree with the packet CRC, an error bit is set.

ADDRESSING

Packets can be received using 3 different destination addressing schemes: physical, logical and promiscuous.

The first type is a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the LANCE during an initialization cycle. There are two types of logical address. One is group type mask where the 48-bit address in the packet is put through a hash filter to map the 48-bit physical addresses into 1 of 64 logical groups. If any of these 64 groups have been preselected as the logical address, then the 48-bit address is stored in main memory. At this time, a look up is performed comparing the 48-bit incoming address with the pre-stored 48-bit logical address. This mode can be useful if sending packets to all of a particular type of device simultaneously (i.e., send a packet to all file servers or all printer servers). Additional details on logical addressing can be found in the INITIALIZATION section under "Logical Address Filter." The second logical address is a broadcast address where all nodes on the network receive the packet. The last receive mode of operation is the so-called "promiscuous mode" in which a node will accept all packets on the coax regardless of their destination address.

COLLISION DETECTION AND IMPLEMENTATION

The Ethernet CSMA/CD network access algorithm is implemented completely within the LANCE. In addition to listening for a clear coax before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the coax at the same time, they will collide and the data on the coax will be garbled. The transmitting nodes listen while they transmit, detect the collision, then continue to transmit for a predetermined length of time to "jam" the network and ensure that all nodes have recognized the collision. The transmitting nodes then delay a random amount of time according to the Ethernet "truncated binary backoff" algorithm in order that the colliding nodes don't try to repeatedly access the network at the same time. Up to 16 attempts to access the network are made by the LANCE before reporting back an error due to excessive collisions.

ERROR REPORTING AND DIAGNOSTICS

Extensive error reporting is provided by the LANCE. Error conditions reported relate either to the network as a whole or to data packets. Network-related errors are recorded as flags in the CSRs and are examined by the CPU following interrupt. Packet-related errors are written into descriptor entries corresponding to the packet.

System errors include:

- Babbling Transmitter
  - Transmitter attempting to transmit more than 1518 data bytes.
- Collision
  - Collision detection circuitry nonfunctional
- Missed packet
  - Insufficient buffer space
- Memory timeout
  - Memory response failure

Packet-related errors:

- CRC
  - Invalid data
- Framing
  - Packet did not end on a byte boundary
- Overflow/Underflow
  - Indicates abnormal latency in servicing a DMA request
- Buffer
  - Insufficient buffer space available

The LANCE performs several diagnostic routines which enhance the reliability and integrity of the system. These include a CRC logic check and two loop back modes (internal/external). Errors may be introduced into the system to check error detection logic. A Time Domain Reflectometer is incorporated into the LANCE to aid system designers locate faults in the Ethernet cable. Shorts and opens manifest themselves in reflections which are sensed by the TDR.

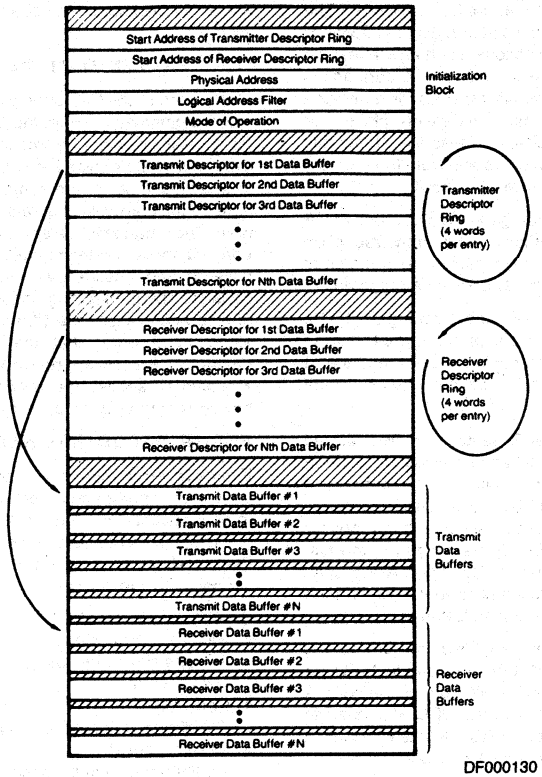


Figure 2a. LANCE/Processor Memory Interface

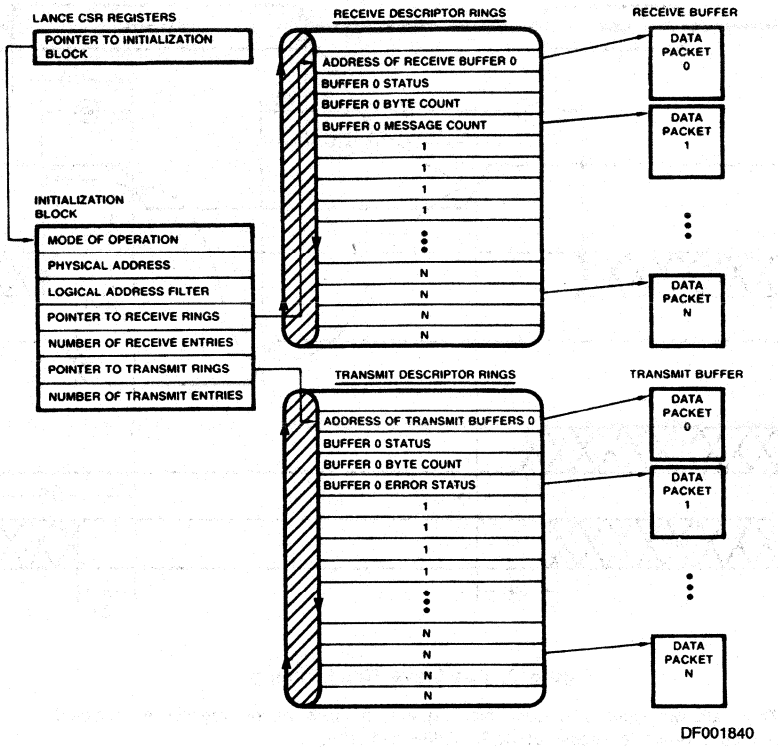


Figure 2b. LANCE Memory Management

BUFFER MANAGEMENT

A key feature of the LANCE and its on-board DMA channel is the flexibility and speed of communication between the LANCE and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings, as shown in Figure 2a. There are separate descriptor rings to describe transmit and receive operations. Up to 128 tasks may be queued up on a descriptor ring awaiting execution by the LANCE. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the length of the data buffer. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The LANCE searches the descriptor rings in a "lookahead manner" to determine the next empty buffer in order to chain buffers together or to handle back-to-back packets. As each buffer is filled, an "own" bit is reset, allowing the host processor to process the data in the buffer.

LANCE INTERFACE

CSR bits such as ACON, BCON and BSWP are used for programming the pin functions used for different interfacing schemes. For example, ACON is used to program the polarity of the Address Strobe signal (ALE/$\overline{AS}$).

BCON is used for programming the pins, for handling either the BYTE/$\overline{WORD}$ method for addressing word organized, byte addressable memories where the BYTE signal is decoded along with the least significant address bit to determine upper or lower byte, or an explicit scheme in which two signals labeled as BYTE MASK ($\overline{BM_0}$ and $\overline{BM_1}$) indicate which byte is addressed. When the BYTE scheme is chosen, the $\overline{BM_1}$ pin can be used for performing the function $\overline{BUSAKO}$.

BCON is also used to program pins for different DMA modes. In a daisy chain DMA scheme, 3 signals are used ($\overline{BUSRQ}$, $\overline{HLDA}$, $\overline{BUSAKO}$). In systems using a DMA controller for arbitration, only $\overline{HOLD}$ and $\overline{HLDA}$ are used.

LANCE IN BUS MASTER MODE

All data transfers from the LANCE in the Bus Master mode are timed by ALE, $\overline{DAS}$, and $\overline{READY}$. The automatic adjustment of the LANCE cycle by the $\overline{READY}$ signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600ns in length and can be increased in 100ns increments.

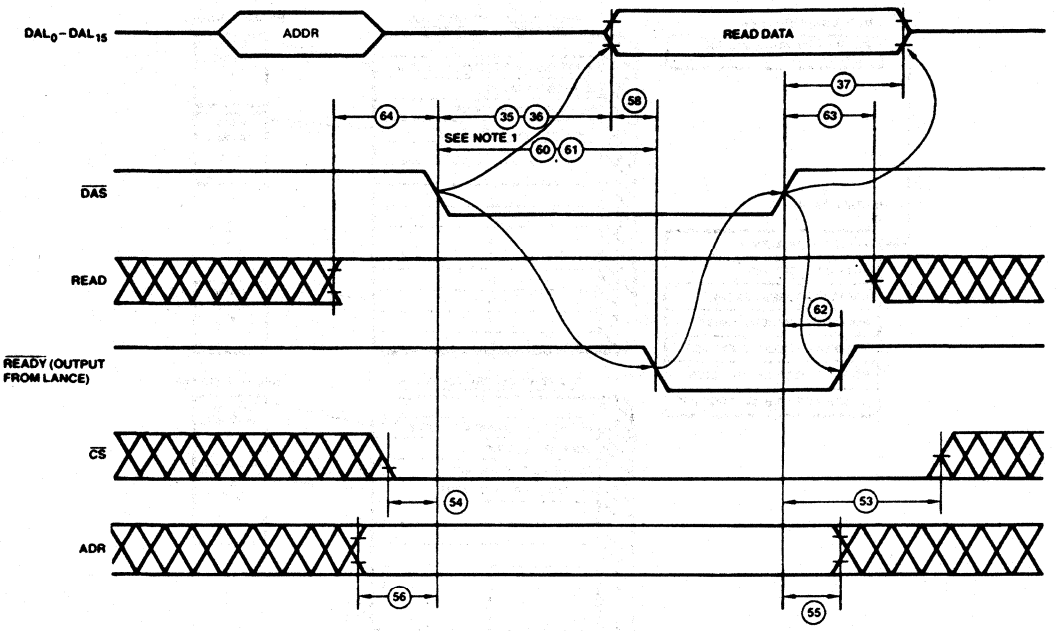


Figure 3. Bus Slave Read Timing

Note: 1. There are two types of delays which depend on which internal register is accessed.
Type 1 refers to access of CSR_0, CSR_3 and RAP.
Type 2 refers to access of CSR_1 and CSR_2 which are longer than Type 1 delay.

READ SEQUENCE

The read cycle is begun by valid addresses being placed on DAL_{00}–DAL_{15} and A_{16}–A_{23}. The BYTE MASK signals are placed valid to indicate a word, upper byte or lower byte memory reference. READ indicates the type of cycle. ALE or $\overline{AS}$ are pulsed, and the trailing edge of either can be used to latch addresses. DAL_{00}–DAL_{15} go into a 3-state mode, and $\overline{DAS}$ falls Low to signal the beginning of the memory access. The memory responds by placing $\overline{READY}$ Low to indicate that the DAL lines have valid data. The LANCE then latches memory data on the rising edge of $\overline{DAS}$, which in turn ends the memory cycle and $\overline{READY}$ returns High. Refer to Figure 5a.

The bus transceiver controls, $\overline{DALI}$ and $\overline{DALO}$, are used to control the bus transceivers. $\overline{DALI}$ signals to strobe data toward the LANCE, and $\overline{DALO}$ signals to strobe data or addresses away from the LANCE. During a read cycle, $\overline{DALO}$ goes inactive before $\overline{DALI}$ becomes active to avoid "spiking" of the bus transceivers.

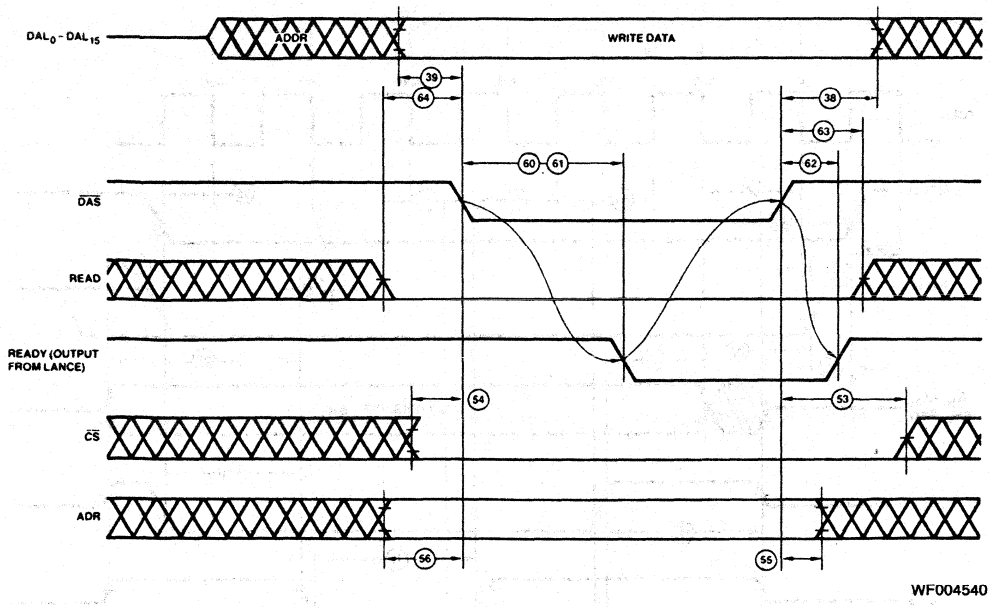


Figure 4. Bus Slave Write Timing

WRITE SEQUENCE

The write cycle is similar to the read cycle except that the DAL_{00}–DAL_{15} lines change from containing addresses to data after either ALE or $\overline{AS}$ goes inactive. After data is valid on the bus, $\overline{DAS}$ goes active. Data to memory is held valid after $\overline{DAS}$ goes inactive. Refer to Figure 5b.

LANCE IN BUS SLAVE MODE

The LANCE enters the Bus Slave Mode whenever $\overline{CS}$ becomes active. This mode must be entered whenever writing or reading the four status control registers (CSR_0, CSR_1, CSR_2, and CSR_3) and the Register Address Pointer (RAP). RAP and CSR_0 may be read or written to at anytime, but the LANCE must be stopped (by setting the stop bit in CSR_0) for CSR_1, CSR_2, and CSR_3 access.

READ SEQUENCE

At the beginning of a read cycle, $\overline{CS}$, READ, and $\overline{DAS}$ are asserted. ADR also must be valid at this time. (If ADR is a "1," the contents of RAP are placed on the DAL lines. Otherwise the contents of the CSR register addressed by RAP are placed on the DAL lines.) After the data on the DAL lines become valid, the LANCE asserts $\overline{READY}$, $\overline{CS}$, READ, $\overline{DAS}$, and ADR must remain stable throughout the cycle. Refer to Figure 3.

WRITE SEQUENCE

This cycle is similar to the read cycle, except that during this cycle, READ is not asserted (READ is Low). The DAL buffers are tristated which configures these lines as inputs. The assertion of $\overline{READY}$ by LANCE indicates to the memory device that the data on the DAL lines have been stored by LANCE in its appropriate CSR register. $\overline{CS}$, READ, $\overline{DAS}$, ADR, and DAL <15:00> must remain stable throughout the write cycle. Refer to Figure 4.

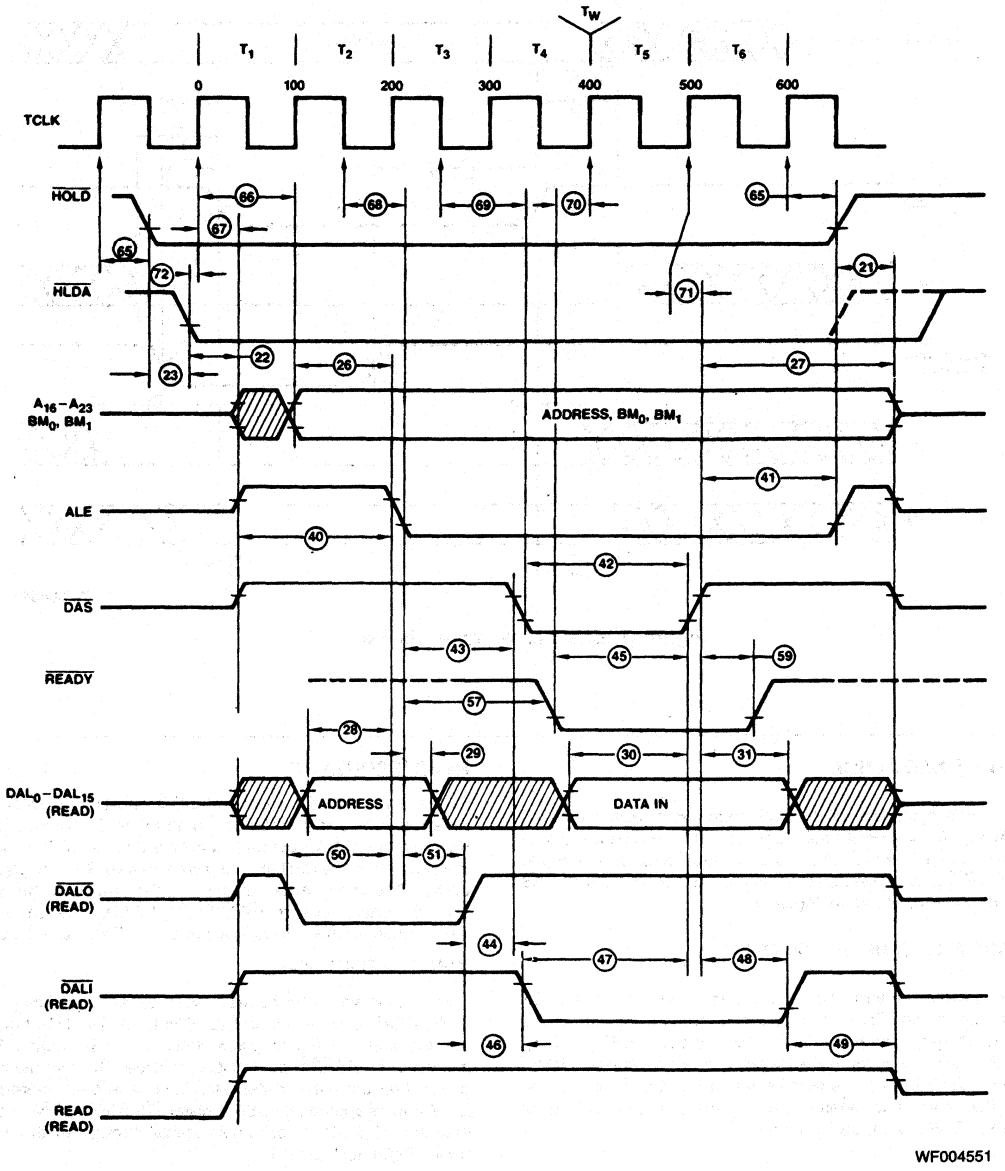


Figure 5a. Bus Master Read Timing

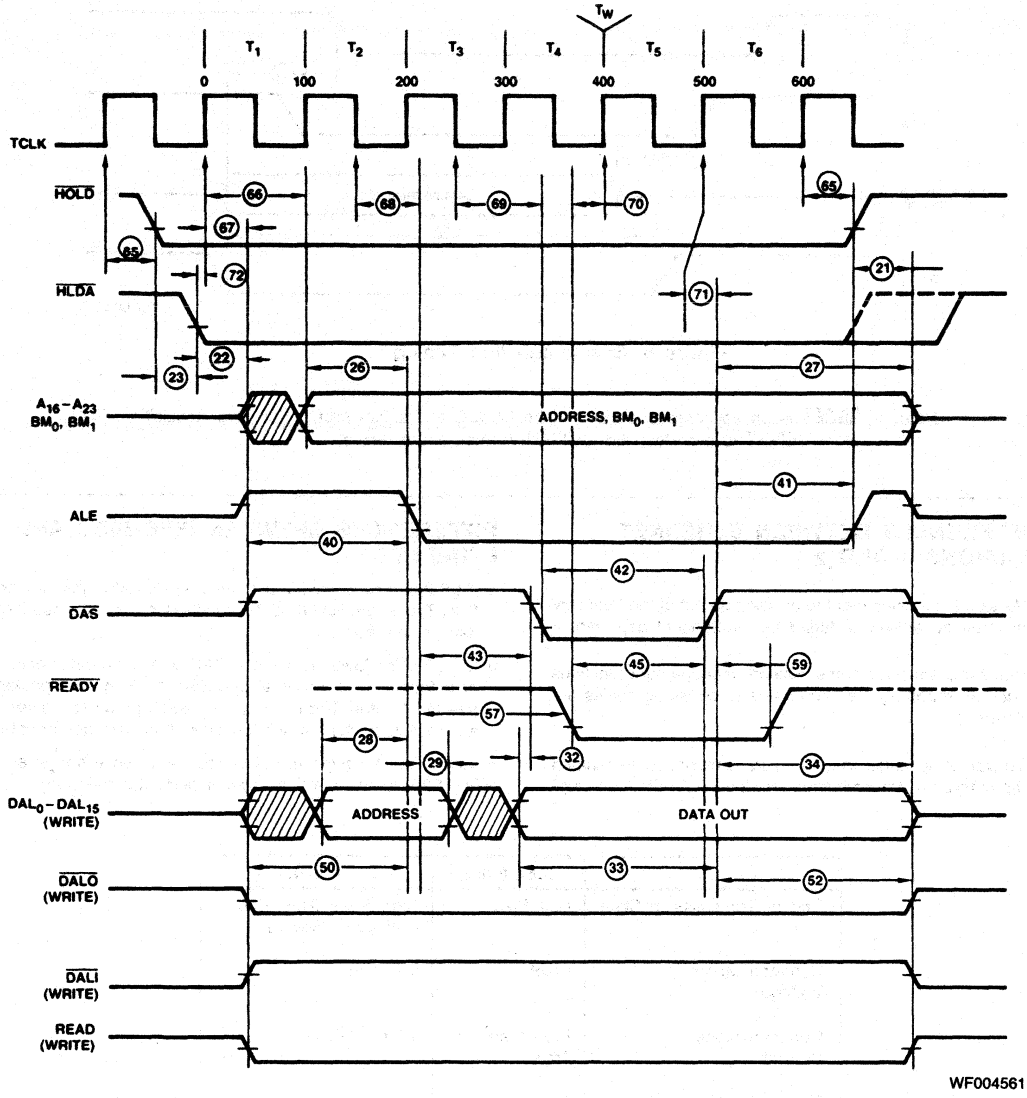


Figure 5b. Bus Master Write Timing

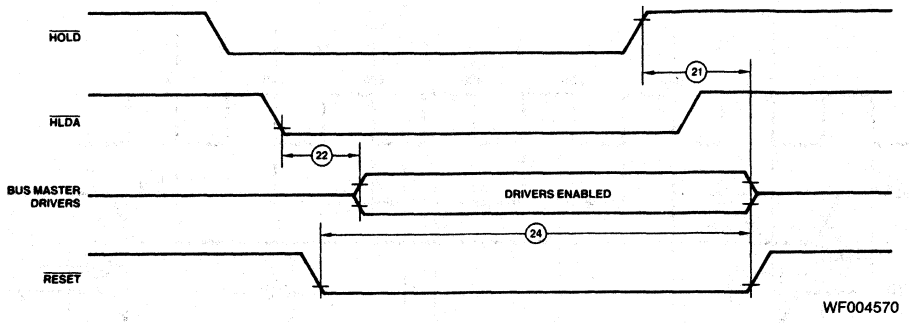


Figure 6. Bus Acquisition Timing

Note: 1. $\overline{\text{RESET}}$ is an asynchronous input to the LANCE and is not part of the Bus Acquisition timing. When $\overline{\text{RESET}}$ is asserted, the LANCE becomes a Bus Slave.

DIFFERENCES BETWEEN ETHERNET VERSIONS 1 AND 2

a. Version 2 specifies that the collision detect of the transceiver must be activated during the interpacket gap time.

b. Version 2 specifies some network management functions, such as reporting the occurrence of collisions, retries and deferrals.

c. Version 2 specifies that when transmission is terminated, the differential transmit lines are driven OV diff. (half step).

DIFFERENCES BETWEEN IEEE-802.3 AND ETHERNET

a. IEEE-802.3 specifies a 2-byte length field rather than a type field. The length field (802.3) described the actual amount of data in the frame.

b. IEEE-802.3 allows the use of a PAD field in the data section of a frame, while Ethernet specifies the minimum packet size at 64 bytes. The use of a PAD allows the user to send and receive packets which have less than 46 bytes of data.

A partial list of significant differences between Ethernet and IEEE-802.3 at the physical layer include the following:

	IEEE-802.3	Ethernet
End of Transmission State	Half Step	High State (Rev 1) or Half Step (Rev 2)
Common Mode Voltage	±5.5V	0 – +5V
Common Mode Current	Less than 1mA	1.6mA ±40%
Receive±, Collision±		
Input Threshold	±160mV	±175mV
Fault Protection	16V	0V

PROGRAMMING SPECIFICATION

This section defines the control and Status Registers and the memory data structures required to program the Am7990 (LANCE).

PROGRAMMING THE Am7990 (LANCE)

The Am7990 (LANCE) is designed to operate in an environment that includes close coupling with a local memory and a microprocessor (HOST). The Am7990 LANCE is programmed by a combination of registers and data structures resident within the chip and in memory. There are four Control and Status Registers (CSRs) within the chip which are programmed by the HOST device. Once enabled, the chip has the ability to access memory locations to acquire additional operating parameters.

The Am7990 has the ability to do independent buffer management as well as transfer data packets to and from the Ethernet. There are three memory structures accessed by the Chip:

1. Initialization Block – 12 words in contiguous memory starting on a word boundary. It also contains the operating parameters necessary for device operation. The initialization block is comprised of:
   - Mode of Operation
   - Physical Address
   - Logical Address Mask
   - Location to Receive and Transmit Descriptor Rings
   - Number of Entries in Receive and Transmit Descriptor Rings
2. Receive and Transmit Descriptor Rings – Two ring structures, one each for incoming and outgoing packets. Each entry in the rings is 4 words long and each entry must start on a quadword boundary. The Descriptor Rings are comprised of:
   - The address of a data buffer
   - The length of that data buffer
   - Status information associated with the buffer
3. Data Buffers – Contiguous portions of memory reserved for packet buffering. Data buffers may begin on arbitrary byte boundaries.

In general, the programming sequence of the chip may be summarized as:

1. Programming the chip's CSRs by a host device to locate an initialization block in memory. The byte control, byte addressing, and address latch enable modes are defined here also.
2. The chip loading itself with the information contained within the initialization block.
3. The chip accessing the descriptor rings for packet handling.

CONTROL AND STATUS REGISTERS

There are four Control and Status Registers (CSRs) resident within the chip. The CSRs are accessed through two bus addressable ports, an address port (RAP) and a data port (RDP).

ACCESSING THE CONTROL AND STATUS REGISTERS

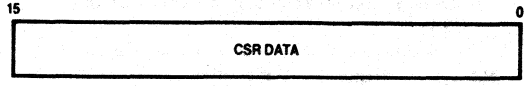
The CSRs are read (or written) in a two step operation. The address of the CSR to be accessed is written into the address port (RAP) during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the data port (RDP) is read from (or written into) the CSR selected in the RAP.

Once written, the address in RAP remains unchanged until rewritten.

To distinguish the data port from the address port, a discrete I/O pin is provided.

ADR I/O Pin	Port
L	Register Data Port (RDP)
H	Register Address Port (RAP)

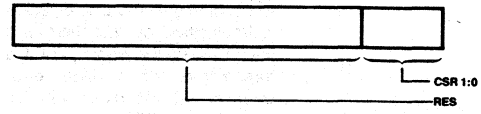
Register Data Port (RDP)



AF001450

Bit	Name	Description
15:00	CSR Data	Writing data into RDP writes the data into the CSR selected in RAP. Reading the data from the RDP reads the data from the CSR selected in RAP. CSR_1, CSR_2 and CSR_3 are accessible only when the STOP bit of CSR_0 is set. If the STOP bit is not set while attempting to access CSR_1, CSR_2 or CSR_3, the chip will return READY, but a READ operation will return undefined data. WRITE operation is ignored.

Register Address Port (RAP)



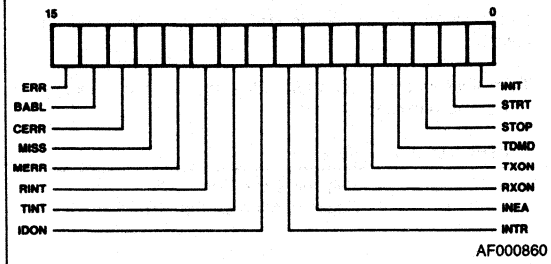
AF001490

Bit	Name	Description
15:02	RES	Reserved and read as zeroes.
01:00	CSR(1:0)	CSR address select. READ/WRITE. Selects the CSR to be accessed through the RDP. RAP is cleared by Bus RESET.

CSR(1:0)	CSR
00	CSR_0
01	CSR_1
10	CSR_2
11	CSR_3

CONTROL AND STATUS REGISTER DEFINITION

Control and Status Register 0 (CSR_0)



AF000860

The LANCE updates CSR_0 by logical "ORING" the previous and present value of CSR_0.

Bit	Name	Description
15	ERR	ERROR summary is set by the "OR" of BABL, CERR, MISS and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; writing it has no effect. It is cleared by Bus $\overline{\text{RESET}}$, by setting the STOP bit, or clearing the individual error flags.
14	BABL	BABBLE is a transmitter timeout error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length packet. BABL is a flag which indicates excessive length in the transmit buffer. It will be set after 1519 data bytes have been transmitted; the chip will continue to transmit until the whole packet is transmitted or until there is a failure before the whole packet is transmitted. When BABL error occurs, an interrupt will be generated if INEA = 1. BABL is READ/CLEAR ONLY and is set by the chip, and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
13	CERR	COLLISION ERROR indicates that the collision input to the chip failed to activate within 2μs after a chip-initiated transmission was completed. The collision after transmission is a transceiver test feature.This function is also known as heartbeat or SQE (Signal Quality Error) test. CERR is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit. CERR error will not cause an interrupt to occur (INTR = 0).
12	MISS	MISSED PACKET is set when the receiver loses a packet because it does not own any receive buffer, indicating loss of data. Silo overflow is not reported because there is no receive ring entry in which to write status. MISS is not valid in internal loopback mode. When MISS is set, an interrupt will be generated if INEA = 1. MISS is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
11	MERR	MEMORY ERROR is set when the chip is the Bus Master and has not received $\overline{\text{READY}}$ within 25.6μs after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off (CSR_0, TXON = 0, RXON = 0) and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
10	RINT	RECEIVER INTERRUPT is set when the chip updates an entry in the Receive Descriptor Ring for the last buffer received or reception is stopped due to a failure. When RINT is set, an interrupt is generated if INEA = 1. RINT is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
09	TINT	TRANSMITTER INTERRUPT is set when the chip updates an entry in the transmit descriptor ring for the last buffer sent or transmission is stopped due to a failure.

05698A

Refer to page 7-1 for Essential Information on Military Devices

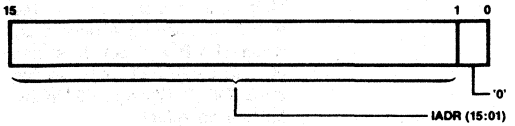
Bit	Name	Description
		When TINT is set, an interrupt is generated if INEA = 1. TINT is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
08	IDON	INITIALIZATION DONE indicates that the chip has completed the initialization procedure started by setting the INIT bit. When IDON is set, the chip has read the Initialization Block from memory and stored the new parameters. When IDON is set, an interrupt is generated if INEA = 1. IDON is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
07	INTR	INTERRUPT FLAG is set by the "OR" of BABL, MISS, MERR, RINT, TINT and IDON. If INEA = 1 and INTR = 1, the INTR I/O pin will be Low. INTR is READ ONLY; writing this bit has no effect. INTR is cleared by RESET, by setting the STOP bit, or by clearing the condition causing the interrupt.
06	INEA	INTERRUPT ENABLE allows the INTR I/O pin to be driven Low when the Interrupt Flag is set. If INEA = 1 and INTR = 1, the INTR I/O pin will be Low. If INEA = 0, the INTR I/O pin will be High, regardless of the state of the Interrupt Flag. INEA is READ/WRITE and cleared by RESET or by setting the STOP bit. INEA cannot be set while STOP bit is set. INEA can be set in parallel or after INIT and/or STRT bit are set.
05	RXON	RECEIVER ON indicates that the receiver is enabled. RXON is set when STRT is set if DRX = 0 in the MODE register in the initialization block and the initialization block has been read by the chip by setting the INIT bit. RXON is cleared when IDON is set from setting the INIT bit and DRX = 1 in the MODE register, or a memory error (MERR) has occurred. RXON is READ ONLY; writing this bit has no effect. RXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
04	TXON	TRANSMITTER ON indicates that the transmitter is enabled. TXON is set when STRT is set if DTX = 0 in the MODE register in the initialization block and the INIT bit has been set. TXON is cleared when IDON is set and DTX = 1 in the MODE register,or an error, such as MERR, UFLO or BUFF, has occurred during transmission. TXON is READ ONLY; writing this bit has no effect. TXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
03	TDMD	TRANSMIT DEMAND, when set, causes the chip to access the Transmit Descriptor Ring without waiting for the polltime interval to elapse. TDMD need not be set to transmit a packet; it merely hastens the chip's response to a Transmit Descriptor Ring entry insertion by the host. TDMD is WRITE WITH ONE ONLY and is cleared by the microcode after it is used. It may read as a "1" for a short time after it is written because the microcode may have been busy when TDMD was set. It is also cleared by $\overline{\text{RESET}}$ or by setting the STOP bit. Writing a "0" in this bit has no effect.
02	STOP	STOP disables the chip from all external activity when set and clears the internal logic. Setting STOP is the equivalent of asserting $\overline{\text{RESET}}$. The chip remains inactive and STOP remains set until the STRT or INIT bit is set. If STRT, INIT and STOP are all set together, STOP will override the other bits and only STOP will be set.

Bit	Name	Description
		STOP is READ/WRITE WITH ONE ONLY and set by RESET. Writing a "0" to this bit has no effect. STOP is cleared by setting either INIT or STRT. CSR_1, CSR_2, and CSR_3 must be reloaded when the STOP bit is set.
01	STRT	START enables the chip to send and receive packets, perform direct memory access, and do buffer management. The STOP bit must be set prior to setting the STRT bit. Setting STRT clears the STOP bit. If STRT and INIT are set together, the INIT function will be executed first. STRT is READ/WRITE and is set with one only. Writing a "0" into this bit has no effect. STRT is cleared by $\overline{RESET}$ or by setting the STOP bit.
00	INIT	INITIALIZE, when set, causes the chip to begin the initialization procedure and access the Initialization Block. The STOP bit must be set prior to setting the INIT bit. Setting INIT clears the STOP bit. If STRT and INIT are set together, the INIT function will be executed first. INIT is READ/WRITE WITH "1" ONLY. Writing a "0" into this bit has no effect. INIT is cleared by $\overline{RESET}$ or by setting the STOP bit.

Control and Status Register 1 (CSR_1)

RAP = 1

READ/WRITE : Accessible only when the STOP bit of CSR_0 is a ONE. CSR_1 is unaffected by $\overline{RESET}$.



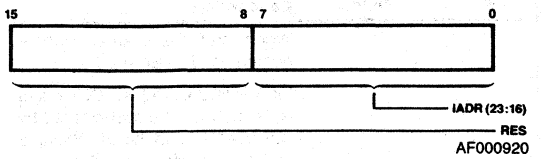
AF000970

Bit	Name	Description
15:01	IADR	The low order 16 bits of the address of the first word (lowest address) in the Initialization Block.
00		Must be zero.

Control and Status Register 2 (CSR_2)

RAP = 2

READ/WRITE : Accessible only when the STOP bit of CSR_0 is a ONE. CSR_2 is unaffected by $\overline{RESET}$.



AF000920

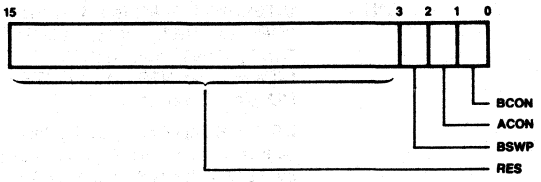
Bit	Name	Description
15:08	RES	Reserved.
07:00	IADR	The high order 8 bits of the address of the first word (lowest address) in the Initialization Block.

Control and Status Register 3 (CSR_3)

CSR_3 allows redefinition of the Bus Master interface.

RAP = 3

READ/WRITE : Accessible only when the STOP bit of CSR_0 is ONE. CSR_3 is cleared by $\overline{RESET}$ or by setting the STOP bit in CSR_0.



AF000900

Bit	Name	Description
15:03	RES	Reserved and read as "0."
02	BSWP	BYTE SWAP allows the chip to operate in systems that consider bits (15:08) of data to be pointed by an even address and bits (07:00) to be pointed by an odd address. When BSWP = 1, the chip will swap the high and low bytes on DMA data transfers between the silo and bus memory. Only data from silo transfers is swapped; the Initialization Block data and the Descriptor Ring entries are NOT swapped. BSWP is READ/WRITE and cleared by $\overline{RESET}$ or by setting the STOP bit in CSR_0.
01	ACON	ALE CONTROL defines the assertive state of ALE when the chip is a Bus Master. ACON is READ/WRITE and cleared by $\overline{RESET}$ and by setting the STOP bit in CSR_0. ACON = 0: ALE Asserted High ACON = 1: ALE Asserted Low

00	BCON	BYTE CONTROL redefines the Byte Mask and Hold I/O pins. BCON is READ/WRITE and cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR_0.

BCON	Pin16	Pin15	Pin17
0	$\overline{BM_1}$	$\overline{BM_0}$	$\overline{HOLD}$
1	$\overline{BUSAKO}$	BYTE	$\overline{BUSRQ}$

All data transfers from the LANCE in the Bus Master mode are in words. However, the LANCE can handle odd address boundaries and/or packets with an odd number of bytes.

INITIALIZATION

INITIALIZATION BLOCK

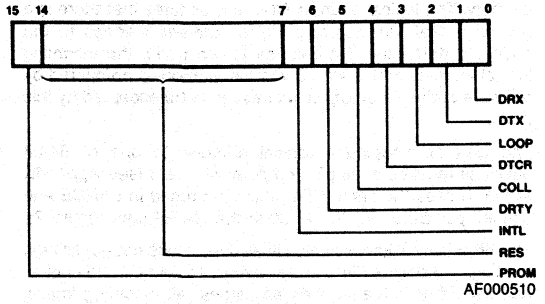
Chip initialization includes the reading of the initialization block in memory to obtain the operating parameters. The following is a definition of the Initialization Block.

The Initialization Block is read by the chip when the INIT bit in CSR_0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure proper parameter initialization and chip operation. After the chip has read the Initialization Block, IDON is set in CSR_0 and an interrupt is generated if INEA = 1.

Higher Addresses	TLEN-TDRA (23:16)	IADR +22
	TDRA (15:00)	IADR +20
	RLEN-RDRA (23:16)	IADR +18
	RDRA (15:00)	IADR +16
	LADRF (63:48)	IADR +14
	LADRF (47:32)	IADR +12
	LADRF (31:16)	IADR +10
	LADRF (15:00)	IADR +08
	PADR (47:32)	IADR +06
	PADR (31:16)	IADR +04
	PADR (15:00)	IADR +02
Base Address of Block	MODE	IADR +00

Mode

The Mode Register allows alteration of the chip's operating parameters. Normal operation is with the Mode Register clear.

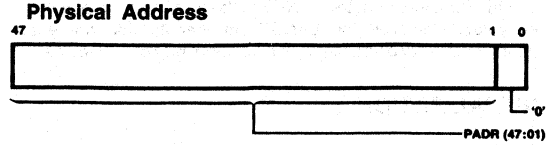


AF000510

Bit	Name	Description
15	PROM	PROMISCUOUS mode. When PROM = 1, all incoming packets are accepted.
14:07	RES	RESERVED
06	INTL	INTERNAL LOOPBACK is used with the LOOP bit to determine where the loopback is to be done. Internal loopback allows the chip to receive its own transmitted packet. Since this represents full duplex operation, the packet size is limited to 8-32 bytes. Internal loopback in the LANCE is operational only when the packets are addressed to the node itself. The Lance will not receive any packets externally when it is in internal loopback mode. EXTERNAL LOOPBACK allows the LANCE to transmit a packet through the SIA transceiver cable out to the Ethernet coax. It is used to determine the operability of all circuitry and connections between the LANCE and the coaxial cable. Multicast addressing in external loopback is valid only when DTCR = 1 (user needs to append the 4 bytes CRC). In external loopback, the LANCE also receives packets from other nodes. INTL is only valid if LOOP = 1; otherwise, it is ignored. **LOOP INTL LOOPBACK** 0 X No loopback, normal 1 0 External 1 1 Internal
05	DRTY	DISABLE RETRY. When DRTY = 1, the chip will attempt only one transmission of a packet. If there is a collision on the first transmission attempt, a Retry Error (RTRY) will be reported in Transmit Message Descriptor 3 (TMD_3).
04	COLL	FORCE COLLISION. This bit allows the collision logic to be tested. The chip must be in internal loopback mode for COLL to be valid. If COLL = 1, a collision will be forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a retry error reported in TMD_3.
03	DTCR	DISABLE TRANSMIT CRC. When DTCR = 0, the transmitter will generate and append a CRC to the transmitted packet. When DTCR = 1, the CRC logic is allocated to the receiver and no CRC is generated and sent with the transmitted packet.

Bit	Name	Description
		During loopback, DTCR = 0 will cause a CRC to be generated on the transmitted packet, but no CRC check will be done by the receiver since the CRC logic is shared and cannot generate and check CRC at the same time. The generated CRC will be written into memory with the data and can be checked by the host software. If DTCR = 1 during loopback, the host software must append a CRC value to the transmit data. The receiver will check the CRC on the received data and report any errors.
02	LOOP	LOOPBACK allows the chip to operate in full duplex mode for test purposes. The packet size is limited to 8–32 bytes.The received packet can be up to 36 bytes (32 + 4 bytes CRC) when DTCR=0. During loopback, the runt packet filter is disabled because the maximum packet is forced to be smaller than the minimum size Ethernet packet (64 bytes). LOOP = 1 allows simultaneous transmission and reception for a message constrained to fit within the silo. The chip waits until the entire message is in the silo before serial transmission begins. The incoming data stream fills the silo from behind as it is being emptied. Moving the received message out of the silo to memory does not begin until reception has ceased. In loopback mode, transmit data chaining is not possible. Receive data chaining is possible if receive buffers are 32 bytes long to allow time for lookahead.
01	DTX	DISABLE THE TRANSMITTER causes the chip to not access the Transmitter Descriptor Ring, and therefore, no transmissions are attempted. DTX = 1 will clear the TXON bit in CSR_0 when initialization is complete.
00	DRX	DISABLE THE RECEIVER causes the chip to reject all incoming packets and not access the Receive Descriptor Ring. DRX = 1 will clear the RXON bit in the CSR_0 when initialization is complete.

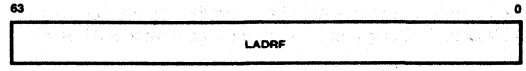
Physical Address



AF000520

Bit	Name	Description
47:00	PADR	PHYSICAL ADDRESS is the unique 48-bit physical address assigned to the chip. PADR (0) must be zero.

Logical Address Filter



AF000500

Bit	Name	Descriptor
63:00	LADRF	The 64-bit mask used by the chip to accept logical addresses.

If the first bit of an incoming address is a "1" [PADR (0) = 1], the address is deemed logical and is passed through the logical address filter.

The logical address filter is a 64-bit mask composed of four sixteen-bit registers, LADRF (63:00) in the initialization block, that is used to accept incoming Logical Addresses. The incoming address is sent through the CRC circuit. After all 48 bits of the address have gone through the CRC circuit, the high order 6 bits of the resultant CRC (32-bit CRC) are strobed into a register. This register is used to select one of the 64-bit positions in the Logical Address Filter. If the selected filter bit is a "1," the address is accepted and the packet will be put in memory. The logical address filter only assures that there is a possibility that the incoming logical address belongs to the node. To determine if it belongs to the node, the incoming logical address that is stored in main memory is compared by software to the list of logical addresses to be accepted by this node.

The task of mapping a logical address to one of 64-bit positions requires a simple computer program (see Appendix A) which uses the same CRC algorithm (used in LANCE and defined per Ethernet) to calculate the HASH (see Figure 7).

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is always enabled. If the Logical Address Filter is loaded with all zeroes, all incoming logical addresses except broadcast will be rejected. The multicast addressing in external loopback is operational only when DTCR in the mode register is set to 1.

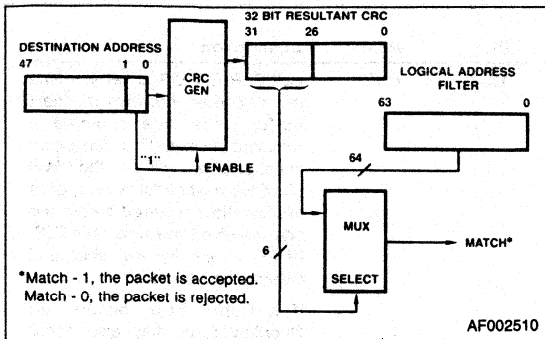
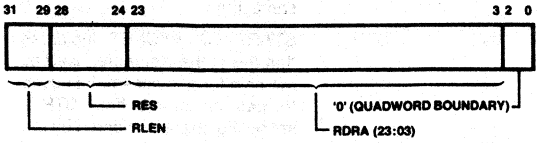


Figure 7. Logical Address Filter Operation

Receive Descriptor Ring Pointer

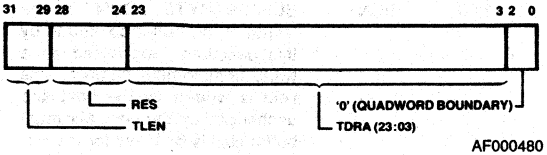


Bit	Name	Description
15:13	RLEN	RECEIVE RING LENGTH is the number of entries in the receive ring expressed as a power of two.

RLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

12:08	RES	RESERVED
07:00 15:03	RDRA	RECEIVE DESCRIPTOR RING ADDRESS is the base address (lowest address) of the Receive Descriptor Ring.
02:00		MUST BE ZEROES. These bits are RDRA (02:00) and must be zeroes because the Receive Rings are aligned on quadword boundaries.

Transmit Descriptor Ring Pointer



Bit	Name	Description
15:13	TLEN	TRANSMIT RING LENGTH is the number of entries in the Transmit Ring expressed as a power of two.

TLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

12:08	RES	RESERVED
07:00 15:03	TDRA	TRANSMIT DESCRIPTOR RING ADDRESS is the base address (lowest address) of the Transmit Descriptor Ring.
02:00		MUST BE ZEROES. These bits are TDRA (02:00) and must be zeroes because the Transmit Rings are aligned on quadword boundaries.

BUFFER MANAGEMENT

Buffer Management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are two rings allocated for the device: a Receive ring and a Transmit ring. The device is capable of polling each ring for buffers to either empty or fill with packets to or from the channel. The device is also capable of entering status information in the descriptor entry. Chip polling is limited to looking one ahead of the descriptor entry the chip is currently working with.

The location of the descriptor rings and their length are found in the initialization block, accessed during the initialization procedure by the chip. Writing a "ONE" into the STRT bit of CSR_0 will cause the chip to start accessing the descriptor rings and enable it to send and receive packets.

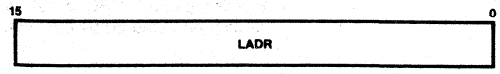
The chip communicates with a HOST device (probably a mircoprocessor) through the ring structures in memory. Each entry in the ring is either "owned" by the chip or the HOST. There is an ownership bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish ownership of the descriptor entry to the other device; it can never take ownership, and no device can change the state of any field in any entry after it has relinquished ownership.

DESCRIPTOR RINGS

Each descriptor in a ring in memory is a 4-word entry. The following is the format of the receive and the transmit descriptors.

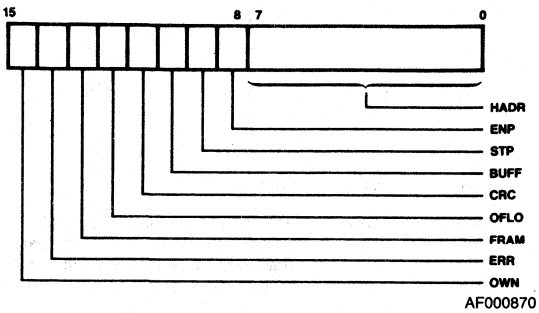
Receive Message Descriptor Entry

Receive Message Descriptor 0 (RMD_0)



Bit	Name	Description
15:00	LADR	The LOW ORDER 16 address bits of the buffer pointed to by this descriptor. LADR is written by the host and unchanged by the chip.

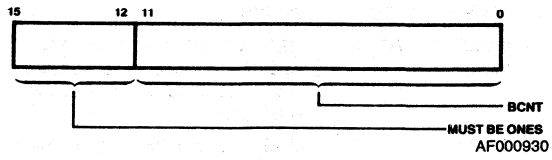
Receive Message Descriptor 1 (RMD_1)



AF000870

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the chip (OWN = 1). The chip clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the chip or host has relinquished ownership of a buffer, it must not change any field in the four words that comprise the descriptor entry.
14	ERR	ERROR summary is the "OR" of FRAM, OFLO, CRC or BUFF. ERR is set by the chip and cleared by the host.
13	FRAM	FRAMMING ERROR indicates that the incoming packet contained a noninteger multiple of eight bits and there was a CRC error. If there was not a CRC error on the incoming packet, then FRAM will not be set even if there was a noninteger multiple of eight bits in the packet. FRAM is not valid in internal loopback mode. FRAM is set by the chip and cleared by the host.
12	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming packet due to an inability to store the packet in a memory buffer before the internal silo overflowed. OFLO is set by the chip and cleared by the host.
11	CRC	CRC indicates that the receiver has detected a CRC error on the incoming packet. CRC is set by the chip and cleared by the host.
10	BUFF	BUFFER ERROR is set any time the chip does not own the next buffer while data chaining a received packet. This can occur in either of two ways: 1) the OWN bit of the next buffer is zero, or 2) silo overflow occurred before the chip received the next STATUS. BUFF is set by the chip and cleared by the host. If a Buffer Error occurs, an Overflow Error may also occur internally in the SILO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time.
09	STP	START OF PACKET indicates that this is the first buffer used by the chip for this packet. It is used for data chaining buffers. STP is set by the chip and cleared by the host.
08	ENP	END OF PACKET indicates that this is the last buffer used by the chip for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining. ENP is set by the chip and cleared by the host.
07:00	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and unchanged by the chip.

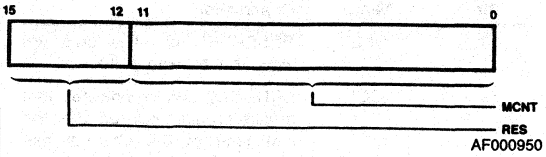
Receive Message Descriptor 2 (RMD_2)



AF000930

Bit	Name	Description
15:12		MUST BE ONES. This field is written by the host and unchanged by the chip.
11:00	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as a two's complement number. This field is written by the host and unchanged by the chip. Minimum buffer size is 64 bytes for the first buffer of packet.

Receive Message Descriptor 3 (RMD_3)

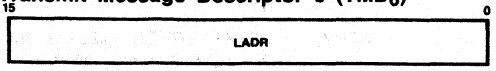


AF000950

Bit	Name	Description
15:12	RES	RESERVED and read as zeroes.
11:00	MCNT	MESSAGE BYTE COUNT is the length in bytes of the received message. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the chip and cleared by the host.

Transmit Message Descriptor Entry

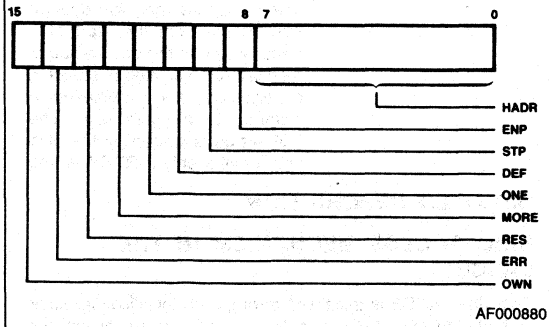
Transmit Message Descriptor 0 (TMD_0)



AF000940

Bit	Name	Description
15:00	LADR	The LOW ORDER 16 address bits of the buffer pointed to by this descriptor. LADR is written by the host and unchanged by the chip.

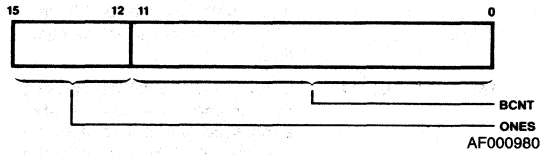
Transmit Message Descriptor 1 (TMD_1)



AF000880

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the chip (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The chip clears the OWN bit after transmitting the contents of the buffer. Both the host and the chip must not alter a descriptor entry after it has relinquished ownership.
14	ERR	ERROR summary is the "OR" of LCOL, LCAR, UFLO or RTRY. ERR is set by the chip and cleared by the host.
13	RES	RESERVED bit. The chip will write this bit with a "0."
12	MORE	MORE indicates that more than one retry was needed to transmit a packet. MORE is set by the chip and cleared by the host.
11	ONE	ONE indicates that exactly one retry was needed to transmit a packet. ONE is set by the chip and cleared by the host. One flag is not valid when LCOL is set.
10	DEF	DEFERRED indicates that the chip had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the chip is ready to transmit. DEFER is set by the chip and cleared by the host.
09	STP	START OF PACKET indicates that this is the first buffer to be used by the chip for this packet. It is used for data chaining buffers. STP is set by the host and unchanged by the chip. The STP bit must be set in the first buffer of the packet, or the LANCE will skip over this descriptor and poll the next descriptor(s) until the OWN and STP bit are set.
08	ENP	END OF PACKET indicates that this is the last buffer to be used by the chip for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining. ENP is set by the host and unchanged by the chip.
07:00	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and unchanged by the chip.
15:12	ONES	Must be ones. This field is set by the host and unchanged by the chip.

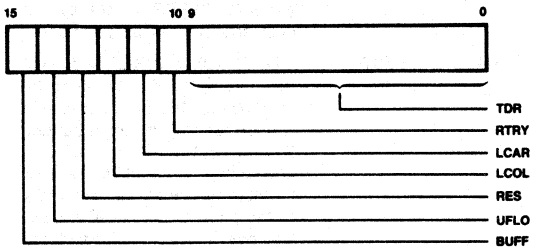
Transmit Message Descriptor 2 (TMD_2)



AF000980

Bit	Name	Description
11:00	BCNT	BUFFER BYTE COUNT is the usable length in bytes of the buffer pointed to by this descriptor expressed as a two's complement number. This is the number of bytes from this buffer that will be transmitted by the chip. This field is written by the host and unchanged by the chip. The first buffer of a packet has to be at least 100 bytes minimum when data chaining and 64 bytes (DTCR = 1) or 60 bytes (DCTR = 0) when not data chaining.

Transmit Message Descriptor 3 (TMD_3)



AF000890

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the chip during transmission when the chip does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: either the OWN bit of the next buffer is zero, or SILO underflow occurred before the chip received the next STATUS signal. BUFF is set by the chip and cleared by the host. BUFF error will turn off the transmitter (CSR_0, TXON = 0) If a Buffer Error occurs, an Underflow Error will also occur internally in the SILO. An Underflow Error will not be reported in the descriptor status entry unless both BUFF and UFLO errors occur at the same time.
14	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the SILO has emptied before the end of the packet was reached. Upon UFLO error, transmitter is turned off (CSR_0, TXON = 0). UFLO is set by the chip and cleared by the host.
13	RES	RESERVED bit. The chip will write this bit with a "0."
12	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The chip does not retry on late collisions. LCOL is set by the chip and cleared by the host.
11	LCAR	LOSS OF CARRIER is set when the carrier input (RENA) to the chip goes false during a chip-initiated transmission. The chip does not retry upon loss of carrier. It will continue to transmit the whole packet until done. LCAR is not valid in INTERNAL LOOPBACK MODE. LCAR is set by the chip and cleared by the host.
10	RTRY	RETRY ERROR indicates that the transmitter has failed in 16 attempts to successfully transmit a message due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after 1 failed transmission attempt. RTRY is set by the chip and cleared by the host.
09:00	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal chip counter that counts from the start of a transmission to the occurrence of a collision. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the chip and is valid only if RTRY is set.

DETAILED DESCRIPTION

RING ACCESS MECHANISM IN THE LANCE

Once the LANCE is initialized through the initialization block and started, the CPU and the LANCE communicate via transmit and receive rings, for packet transmission and reception.

There are 2 sets of RAM locations (four 16-bit register per set, corresponding to the 4 entries in each descriptor) in the LANCE. The first set points to the current buffer, and they are the working registers which are used for transferring the data for the packet. The second set contains the pointers to the next buffer in the ring which the LANCE obtained from the lookahead operation.

There are three types of ring access in the LANCE. The first type is when the LANCE polls the rings to own a buffer. The second type is when the buffers are data chained. The LANCE does a lookahead operation between the time that it is transferring data to/from the SILO; this lookahead is done only once. The third type is when the LANCE tries to own the next descriptor in the ring when it clears the OWN bit for the current buffer.

Transmit Ring Buffer Management

When there is no Ethernet activity, the LANCE will automatically poll the transmit ring in the memory once it has started (CSR_0, STRT = 1). This polling occurs every 1.6ms, (CSR_0 TDMD bit = 0) and consists of reading the status word of the transmit Ring, TMD_1, until the LANCE owns the descriptor. The LANCE will read TMD_0 and TMD_2 to get the rest of the buffer address and the buffer byte count when it owns the descriptor. Each of these memory reads is done separately with a new arbitration cycle for each transfer.

If the transmit buffers are data chained (current buffer ENP = 0), the LANCE will lookahead the next descriptor in the ring while transferring the current buffer into the SILO (see Figure 8a). The LANCE does this lookahead only once. If it does not own the next transmit Descriptor Table Entry (DTE) (2nd T_X ring for this packet) it will transmit the current buffer and updates the status of current Ring with the BUFF and UFLO error bits set. If the LANCE owns the 2nd DTE, it will also read the buffer address and the buffer byte count of this entry. Once the LANCE has finished emptying the current buffer, it clears the OWN bit for this buffer, and immediately starts loading the SILO from the next (2nd) buffer. Between DMA bursts, starting from the 2nd buffer, the LANCE does a lookahead again to check if it owns the next (3rd) buffer. This activity goes on until the last transmit DTE indicates the end of the packet (TMD_1, ENP = .1). Once the last part of the packet has been transmitted out from the SILO to the cable, the LANCE will update the status in TMD_1, TMD_3 (TMD_3 is updated only when there is an error) and relinquishes the last buffer to the CPU. The LANCE tries to own the next buffer (first buffer of the next packet), immediately after it relinquishes the last buffer of the current packet. This guarantees the back-to-back transmission of the packets. If the LANCE does not own the next buffer, it then polls the T_X ring every 1.6ms.

When an error occurs before all of the buffers get transmitted, the status, TMD_3, is updated in the current DTE, own bit is cleared in TMD_1, and TINT bit is set in CSR_0 which causes an interrupt if INEA = 1. The LANCE will then skip over the rest of the descriptors for this packet (clears the OWN bit and sets the TINT bit in CSR_0) until it finds a buffer with both the STP and OWN bit being set (it indicates the first buffer for the next packet).

When the transmit buffers are not data chained (current descriptor's ENP = 1), the LANCE will not perform any lookahead operation. It will transmit the current buffer, update the TMD_3 if any error, and then update the status and clear the OWN bit in TMD_1. The LANCE will then immediately check the next descriptor in the ring to see if it owns it. If it does, the LANCE will also read the rest of the entries from the descriptor table. If the LANCE does not own it, it will poll the ring once every 1.6ms until it owns it. User may set the TDMD bit in CSR_0 when it has relinquished a buffer to the LANCE. This will force the LANCE to check the OWN bit at this buffer without waiting for the polling time to elapse.

Receive Ring Buffer Management

Receive Ring access is similar to the transmit ring access. Once receiver is enabled, the LANCE will always try to have a receive buffer available, should there be a packet addressed to this node for reception. Therefore, when a packet has not arrived, the LANCE will poll the receive ring entry, once every 1.6ms, until it owns the current receive DTE. Once the LANCE owns the buffer, it will read RMD_0 and RMD_2 to get the rest of buffer address and buffer byte count. When the packet arrives from the cable, the LANCE will first check to see if it owns a buffer. If not, it will poll the receive ring once for a buffer. If it does not own the buffer, it will set the MISS error in CSR_0 and will not poll the receive ring until the packet ends.

Assuming the LANCE owns a receive buffer when the packet arrives, it will perform a lookahead operation on the next DTE between periods when it is dumping the received data from the SILO to the first receive buffer in case the current buffer requires data chaining. When the LANCE owns the buffer, the lookahead operation consists of 3 separate single word DMA reads: RMD_1, RMD_0, and RMD_2. When the LANCE does not own the next buffer, the lookahead operation consists of only one single DMA read, RMD_1. Either lookahead operation is done only once. Following the lookahead operation, whether LANCE owns the next buffer or not, the LANCE will transfer the data from SILO to the first receive buffer for this packet in burst mode (8 word transfer per one DMA cycle arbitration).

If the packet being received requires data chaining, and the LANCE does not own the 2nd DTE, the LANCE will update the current buffer status, RMD_1, with the BUFF and/or OVFL error bits set. If the LANCE does own the next buffer (2nd DTE) from previous lookahead, the LANCE will relinquish the current buffer and start filling up the 2nd buffer for this packet. Between the time that the LANCE is transferring data from the SILO to 2nd buffer, it does a lookahead operation again to see if it owns the next (3rd) buffer. If the LANCE does own the third DTE, it will also read RMD_0, and RMD_2 to get the rest of buffer pointer address and buffer byte count.

This activity continues on until the LANCE recognizes the end of the packet (cable is idle); it then updates the current buffer status with the end of packet bit (ENP) set. The LANCE will also update the message byte count (RMD_3) with the total number of bytes received for this packet in the current buffer (the last buffer for this packet).

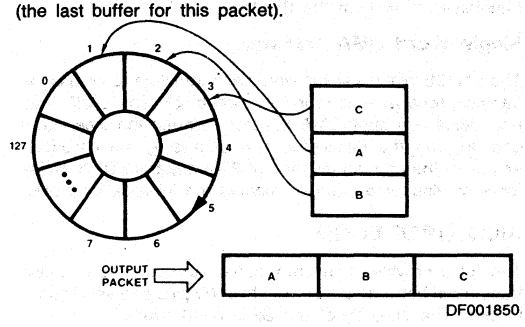


Figure 8a. Data Chaining (Transmit)

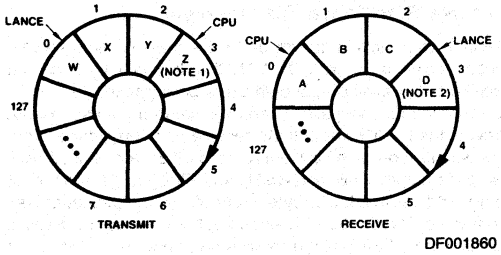


Figure 8b. Buffer Management Descriptor Rings

Notes: 1. W, X, Y, Z are the packets queued for transmission.
2. A, B, C, D are the packets received by the LANCE.

LANCE DMA TRANSFER (BUS MASTER MODE)

There are two types of DMA Transfers with the LANCE:

- Burst mode DMA
- Single word DMA

Burst Mode DMA

Burst DMA is used for Transmission or Reception of the Packets, (Read/Write from/to Memory).

The Burst Transfers are 8 consecutive word reads (transmit) or writes (receive) that are done on a single bus arbitration cycle. In other words, once the LANCE receives the bus acknowledge, ($\overline{HLDA}$ = Low), it will do 8 word transfers (8 DMA cycle, min. at 600ns per cycle) without releasing the bus request signal ($\overline{HOLD}$ = Low). If there are more than 16 bytes empty in the SILO, in transmit mode, or at least 16 bytes of data, in the SILO in receive mode, when the LANCE releases the bus ($\overline{HOLD}$ deasserted), the LANCE will request the bus again within 700ns. ($\overline{HOLD}$ dwell time). Burst DMAs are always 8 cycle transfers unless there are less than 8 words left to be transferred in to/from the SILO.

Single Word DMA Transfer

The LANCE initiates single word DMA transfers to access the transmit, receive rings or initialization block. The LANCE will not initiate any burst DMA transfer between the time that it gets to own the descriptor, and accessing the descriptor entries in the ring (an average of 3 - 4 separate DMA cycles for a multibuffer packet) or reading the initialization block.

SILO OPERATION

The SILO provides temporary buffer storage for data being transferred between the parallel bus I/O pins and serial bus I/O pins. The capacity of the SILO is 48 bytes.

Transmit

Data is loaded into the SILO under internal microprogram control. SILO has to be more than 16 bytes empty before the LANCE requests the bus ($\overline{HOLD}$ is asserted). The LANCE will start sending the preamble (if the line is idle) as soon as the first byte is loaded to the SILO from memory. Should transmitter be required to back off, there could be up to 32 bytes of data in the SILO ready for transmission. Reception has priority over transmission during the time that the transmitter is backing off.

Receive

Data is loaded into the SILO from the serial input shift register during reception. Data leaves the SILO under microprogram control. The LANCE microcode will wait until there are at least 16 bytes of data in the SILO before initiating a DMA burst transfer. Preamble (including the synch) is not loaded into the SILO.

Note: SILO is used as an alternative name for FIFO.

SILO - Memory Byte Alignment

Memory buffers may begin and end on arbitrary byte boundaries. Parallel data is byte aligned between the SILO and DAL lines (DAL_0–DAL_{15}). Byte alignment can be reversed by setting the Byte Swap (BSWP) bit in CSR_3.

TRANSMISSION - WORD READ FROM EVEN MEMORY ADDRESS

BSWP = 0: SILO BYTE n gets DAL <07:00>
SILO BYTE n + 1 gets DAL <15:08>

BSWP = 1: SILO BYTE n gets DAL <15:08>
SILO BYTE n + 1 gets DAL <07:00>

TRANSMISSION - BYTE READ FROM EVEN MEMORY ADDRESS

BSWP = 0: SILO BYTE n gets DAL <07:00>
BSWP = 1: SILO BYTE n gets DAL <15:08>

TRANSMISSION - BYTE READ FROM ODD MEMORY ADDRESS

BSWP = 0: SILO BYTE n gets DAL <15:08>
BSWP = 1: SILO BYTE n gets DAL <07:00>

RECEPTION - WORD WRITE TO EVEN MEMORY ADDRESS

BSWP = 0: DAL <07:00> gets SILO BYTE n
BSWP = 1: DAL <15:08> gets SILO BYTE n + 1

RECEPTION - BYTE WRITE TO EVEN MEMORY ADDRESS

BSWP = 0: DAL <07:00> gets SILO BYTE n
DAL <15:08> - don't care

BSWP = 1: DAL <15:08> gtes SILO BYTE n
DAL <07:00> - don't care

RECEPTION - BYTE WRITE TO ODD MEMORY ADDRESS

BSWP = 0: DAL <07:00> - don't care
DAL <15:08> gets SILO BYTE n

BSWP = 1: DAL <15:08> - don't care
DAL <07:00> gets SILO BYTE n

THE LANCE RECOVERY AND REINITIALIZATION

The transmitter and receiver section of the LANCE are turned on via the initialization block (MODE REG: DRX, DTX bits). The state of the transmitter and the receiver are monitored through the CSR_0 register (RXON, TXON bits). The LANCE must be reinitialized if the transmitter and/or the receiver has not been turned on during the original initialization, and later it is desired to have them turned on. Another reason why it may be desirable to reinitialize the LANCE, to turn the transmitter and/or receiver back on again, is when either section shuts off because of an error (MERR, UFLO, TX BUFF error). Care must be taken when the LANCE is reinitialized. The user should rearrange the descriptors in the transmit or receive ring prior to reinitialization. This is necessary since the transmit and receive descriptor pointers are reset to the beginning of the ring upon initialization.

Another way of starting the LANCE, once it has stopped (STOP = 0 in CSR_0), is by setting the STRT bit in CSR_0. The STRT puts the LANCE in operation in accordance with the

parameters set up in the mode register. If DTX and/or DRX are set to 0 in the mode register, the transmitter and/or receiver will be turned on again when STRT bit is set.

This approach may look like an easier task than the reinitialization mechanism, where the user is required to rearrange the descriptors in the ring. However, this approach is not recommended when the LANCE is stopped in the middle of a transmission or reception, or when the buffers are data chained.

To reinitialize the LANCE, the user must stop the LANCE by setting the stop bit in CSR_0 prior to reinitialization (setting INIT bit in CSR_0). The user needs to reprogram the CSR_3 because its content gets cleared when the stop bit gets set (soft reset). CSR_3 reprogramming is not needed when default values BCON, ACON, and BSWP are used. CSR_1 and CSR_2 are not affected by STOP bit; however, it is recommended that CSR_1 and CSR_2 be reloaded when the STOP bit is set.

FRAME FORMATTING

The LANCE performs the encapsulation/decapsulation function of the data link layer (2nd layer of ISO model) as follows:

Transmit

In transmit mode, the user must supply the destination address, source address, and Type Field (or Length Field) as a part of data in transmit data buffer memory. The LANCE will append the preamble, synch, and CRC (FCS) to the frame as is shown in Figures 9a and 9b.

Receive

In receive mode, the LANCE strips off the preamble and synch bits and transfers the rest of the frame, including the CRC bytes (4 bytes), to the memory. The LANCE will discard packets with less than 64 bytes (runt packet) and will reuse the receive buffer for the next packet. This is the only case where the packet is discarded. Runt packet is normally the result of a collision.

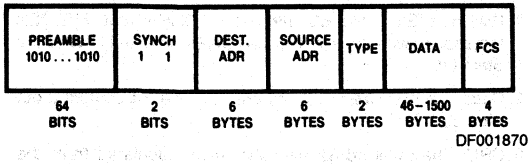


Figure 9a. Ethernet Frame Format

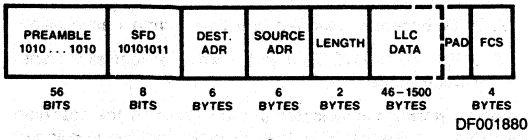


Figure 9b. IEEE 802.3 MAC Frame Format

FRAMING ERROR (DRIBBLING BITS)

The LANCE can handle up to 7 dribbling bits when a received packet terminates; the input to the LANCE, RCLK, stops, following the deassertion of RENA. During the reception, the CRC is generated on every serial bit (including the dribbling bits) coming from the cable, and it gets stored internally on byte boundary. The framing error is reported to the user as follows:

–If the number of the dribbling bits are 1 to 7 bits and there is no CRC error, then there is no Framing error (FRAM = 0).

–If the number of the dribbling bits are less than 8 and there is a CRC error, then there is also a Framing error (FRAM = 1).

–If the number of the dribbling bits = 0, then there is no Framing error. There may or may not be a CRC error.

INTERPACKET GAP TIME (IPG)

The interpacket gap time for back-to-back transmission is 9.6 to 10.6 microseconds, including synchronization. The interpacket delay interval begins immediately after the negation of the RENA signal. During the first 4.1μs of the IPG, RENA activity is masked off internally in the LANCE. If RENA is asserted and remains asserted during the first 4.1μs of IPG following a receive, the LANCE will defer to the packet (it will not receive it). If this condition occurs following a transmit, the LANCE will start to look for the synch bits (011) about 800ns (8 bit time) after the 4.1μs window has elapsed. Therefore, the packet may be received correctly if at least 8 bits of the preamble are left following the 4.1μs window, or the received packet may contain CRC error (not enough preamble bits left, LANCE may be locking to the synch bits in the middle of data), or the received packet may be discarded because of the runt packet (the data loss during the 4.1μs window).

If RENA is asserted after 4.1μs window, the LANCE will treat this as start of a new packet. It will start to look for the synch bits (011) after 8 bit time RENA becomes active. Whenever the LANCE is about to transmit and is waiting for the interpacket delay to elapse, it will begin transmission immediately after the interpacket delay interval, independent of the state of RENA. However, RENA must be asserted during the time that TENA is high. The LCAR (loss of carrier) error bit is otherwise set in TMD_3, after the packet has been transmitted.

COLLISION DETECTION AND COLLISION JAM

Collisions are detected by monitoring the CLSN I/O pin. If CLSN becomes asserted during a frame transmission, TENA will remain asserted for at least 32 (but not more than 40) additional bit times (including CLSN synchronization). This additional transmission after collision is referred to as COLLISION JAM. If collision occurs during the transmission of the preamble, the LANCE continues to send the preamble, and sends the JAM pattern following the preamble. If collision occurs after the preamble, the LANCE will send the JAM pattern following the transmission of the current byte. The JAM pattern is any pattern except the CRC bytes.

RECEIVE BASED COLLISION

If CLSN becomes asserted during the reception of a packet, this reception is immediately terminated. Depending on the timing of COLLISION DETECTION, the following will occur. A collision that occurs within 6 byte times (4.8ms) will result in the packet being rejected because of an address mismatch with the SILO write pointer being reset. A collision that occurs within 64 byte times (51.2ms) will result in the packet being rejected since it is a runt packet. A collision that occurs after 64 byte times (late collision) will result in a truncated packet being written to the memory buffer with the CRC error bit most likely being set in the Status Word of the Receive Ring. Late collision error is not recognized in receive mode.

TRANSMIT BASED COLLISION

When a transmission attempt has been terminated due to the assertion of CLSN, (a collision that occurs within 64 byte times), the LANCE will attempt to retrieve it 15 more times. The LANCE does not try to reread the descriptor entries from the T_X ring upon each collision. The descriptor entries for the current buffer are internally saved. The scheduling of the

retransmissions is determined by a controlled randomized process called "truncated binary exponential backoff." Upon the negation of the COLLISION JAM interval, the LANCE calculates a delay before retransmitting. The delay is an integral multiple of the SLOT TIME. The SLOT TIME is 512 bit times. The number of SLOT TIMES to delay before the nth retransmission is chosen as a uniformly distributed random integer in the range: $0 \leq r \leq 2^k$ where k = min (n, 10).

If all 16 attempts fail, the LANCE sets the RTRY bit in the current Transmit Message Descriptor 3, TMD_3, in memory, gives up ownership (sets the own bit to zero) for this packet, and processes the next packet in transmit ring for transmission. If there is a late collision (collision occurring after 64 byte times), the LANCE will not transmit again; it will terminate the transmission, note the LCOL error in TMD_3, and transmit the next packet in the ring.

COLLISION – MICROCODE INTERACTION

The microprogram uses the time provided by COLLISION JAM, INTERPACKET DELAY, and the backoff interval to restore the address and byte counts internally and starts loading the SILO in anticipation of retransmission. It is important that LANCE be ready to transmit when the backoff interval elapses to utilize the channel properly.

TIME DOMAIN REFLECTOMETRY

The LANCE contains a time domain reflectometry counter. The TDR counter is ten bits wide. It counts at a 10MHz rate. It is cleared by the microprogram and counts upon the assertion of RENA during transmission. Counting ceases if CLSN becomes true, or RENA goes inactive. The counter does not wrap around; once all ONEs are reached in the counter, that value is held until cleared. The value in the TDR is written into memory following the transmission of the packet. TDR is used to determine the location of suspected cable faults.

HEARTBEAT

During the INTERPACKET DELAY following the negation of TENA, the CLSN input is asserted by some transceivers as a self-test. If the CLSN input is not asserted within 2μs following the completion of transmission (after TENA goes low), then the LANCE will set the CERR bit in CSR_0. CERR error will not cause an interrupt to occur (INTR = 0).

CYCLIC REDUNDANCY CHECK (CRC)

The LANCE utilizes the 32 bit CRC function used in the Autodin-II network. Refer to the Ethernet specification (section 6.2.4 Frame Check Sequence Field and Appendix C; CRC Implementation) for more detail. The LANCE requirements for the CRC logic are the following:

1. TRANSMISSION – MODE <02> LOOP = 0, MODE <03> DTCR = 0. The LANCE calculates the CRC from the first bit following the Start bit to the last bit of the data field. The CRC value inverted is appended onto the transmission in one unbroken bit stream.

2. RECEPTION – MODE <02> LOOP = 0. The LANCE performs a check on the input bit stream from the first bit following the Start bit to the last bit in the frame. The LANCE continually samples the state of the CRC check on framed byte boundaries, and, when the incoming bit stream stops, the last sample determines the state of the CRC error. Framing error (FRAM) is not reported if there is no CRC error.

3. LOOPBACK – MODE <02> LOOP = 1, MODE <03> DTRC = 0. The LANCE generates and appends the CRC value to the outgoing bit stream as in Transmission but does not perform the CRC check of the incoming bit stream.

4. LOOPBACK – MODE <02> LOOP = 1 MODE <03> DTRC = 1. LANCE performs the CRC check on the incoming bit stream as in Reception, but does not generate or append the CRC value to the outgoing bit stream during transmission.

LOOPBACK

The normal operation of the LANCE is as a half-duplex device. However, to provide an on-line operational test of the LANCE, a pseudo-full duplex mode is provided. In this mode simultaneous transmission and reception of a loopback packet are enabled with the following constraints:

1. The packet length must be no longer than 32 bytes, and less than eight bytes, exclusive of the CRC.

2. Serial transmission does not begin until the SILO contains the entire output packet.

3. Moving the input packet from the SILO to the memory does not begin until the serial input bit stream terminates.

4. CRC may be generated and appended to the output serial bit stream or may be checked on the input serial bit stream, but not both in the same transaction.

5. In internal loopback, the packets should be addressed to the node itself.

6. In external loopback, multicast addressing can be used only when DTCR = 1 is in the mode register. In this case, the user needs to append the bytes CRC.

Loopback is controlled by bits <06, 03, 02> INTL, DTCR, and LOOP of the MODE register.

SERIAL TRANSMISSION

Serial transmission consists of sending an unbroken bit stream from the T_X I/O pin consisting of:

1. Preamble/Start bit: 62 alternating ONES and ZEROES terminating with the synch in two ONEs. The last ONE is the Start bit.

2. Data: The serialized byte stream from the SILO Shifted out with LSB first.

3. CRC: The inverted 32 bit polynomial calculated from the Data, address, and type field. CRC is not transmitted if:

   i. Transmission of the Data field is truncated for any reason.

   ii. CLSN becomes asserted any time during transmission.

   iii. MODE <03> DTCR = 1 in a normal or loopback transmission mode.

The Transmission is indicated at the I/O pin by the assertion of TENA with the first bit of the preamble and the negation of TENA after the last transmitted bit.

The LANCE starts transmitting the preamble when the following are satisfied:

1. There is at least one byte of data to be transmitted in the SILO.

2. The interpacket delay has elapsed.

3. The backoff interval has elapsed, if a retransmission.

SERIAL RECEPTION

Serial reception consists of receiving an unbroken bit stream on the R_X I/O pin consisting of:

1. Preamble/Start bit: Two ONES occurring a minimum of 8 bit times after the assertion of RENA. The last ONE is the Start bit.
2. Destination Address: The 48 bits (6 bytes) following the Start bit.
3. Data: The serialized byte stream following the Destination Address. The last 4 complete bytes of data are the CRC. The Destination Address and the Data are framed into bytes and enter the SILO. Source Address and Type field are part of the data which are transparent to the LANCE.

Reception is indicated at the I/O pin by the assertion of RENA and the presence of clock on RCLK while TENA is inactive. The LANCE does not not sample the received data until about 800ns after RENA goes high.

MICROPROGRAM OVERVIEW

The Ethernet protocol chip is controlled by a set of semi-independent hardware functions and a microprogram. The following are some of the routines associated with the operation of the LANCE.

Switch Routine

Upon power-up, the microprogram finds itself in a routine to evaluate the INIT, STRT, and STOP bits of CSR_0. INIT and STRT are cleared and STOP is set by the hardware by Bus $\overline{RESET}$. Setting either INIT or STRT through an I/O transfer to CSR_0 will clear STOP. Setting STOP through an I/O transfer will clear INIT and STRT. After seeing STOP cleared, the microprogram tests the state of INIT. If set, it branches to the initialization routine, returns, and tests the state of STRT. If INIT is clear and the STRT is set, the microprogram will go on to the Polling routine without going to the Initialization routine. If, while the STOP bit is set, an I/O transfer to CSR_1, CSR_2, or CSR_3 occurs, the microprogram traps to the CSR service routine.

Initialization Routine

This routine is entered only from the switch routine upon the setting of the INIT bit. Its function is to load the Chip with the data from the initialization block in memory. The routine accesses the initialization block through the address loaded into the LANCE by a trap to CSR_1 and CSR_2 that should have occurred prior to the INIT bit being set. This routine simply sequentially reads the initialization block, in separate single word DMA cycles, and stores the information away in the appropriate elements of the Chip.

When done, the microcode returns to the switch routine.

Polling Routine

This routine is entered from:

1. The switch routine upon the setting of the STRT bit.
2. The receive routine after a packet has been received.
3. The transmit routine after a packet has been transmitted.
4. The transmit routine if a TX Abort occurs.
5. The Memory Error Trap routine (MERR error) after the trap is serviced.

The routine begins by testing to see if the receiver is disabled, and, if not, tests the current receiver buffer ownership bit to see if it owns a buffer. If the Chip had not acquired a buffer previously, the microprogram goes to the receive polling routine to acquire one. Then the microprogram returns from the receive polling routine, or if the Chip had acquired a buffer previously, it tests to see if the transmitter is disabled, and if not, goes to the transmit polling routine to test if there is a buffer to be transmitted.

When the microprogram returns from the transmit polling routine, the microprogram enters a timing loop, and repeats the routine upon timeout. The timer is set around 1.6ms. The timing loop can be overridden by setting the TDMD bit in CSR_0. This will force the microprogram to fall through the wait loop. The TDMD bit is cleared immediately after leaving the wait loop. Therefore, to be effective, TDMD should be set after a buffer has been inserted on the transmit ring (own bit has been set).

During this routine, should the receiver become active, the microprogram traps to the receive routine.

Receive Polling Routine

The Receive Polling Routine is called by the main polling routine to check to see if the chip owns the receive buffer at the current pointer address. The microprogram first reads the status word from the current receive ring descriptor. If the chip does not own the buffer, the microprogram returns to the polling routine. If the chip does own the buffer, the microprogram reads in the rest of the descriptor entry, namely the rest of the buffer address and the buffer byte count. The chip only reads in 3 of the 4 words in the descriptor entry. The message byte count is not read because it is not used by the chip. The message byte count is written by the chip during the status update at the end of a reception. This routine will then return to the polling routine.

Receive Routine

The Receive Routine is entered when the receiver is enabled and the address of the incoming packet has passed address recognition. Once the Receive Routine is entered, the microprogram checks to see if the chip owns the current receive buffer. If it does not own the buffer, the microprogram will check the ownership bit in memory once for a buffer. If it does not own the buffer, the microprogram will set the miss error in CSR_0 and clear the SILO. The microprogram will then return to the polling routine once the current packet ends.

If the chip acquired buffer ownership while the receiver was still active, the microprogram will acquire the rest of the descriptor, namely the buffer address and buffer length. The microprogram will then back up the buffer address and byte count in case the packet is a runt. This is where the microprogram would have come if it had owned a receive buffer when it originally entered the receive routine.

Receive Buffer Lookahead

Receive lookahead is always done during the reception since the LANCE will not know the length of the receive packet. The lookahead is done during the time that SILO is being filled with data from the cable. The microprogram checks to see if there was only one receive buffer. If there is more than one receive buffer, the microprogram checks the ownership of the next buffer. If the chip owns the buffer, it reads the rest of the Descriptor into the internal RAM. If it does not own the buffer, it will continue with the receive routine, trapping to the RX DMA routine whenever there are 16 or more bytes available in the SILO. Lookahead is only done once whenever there is a trap to the receive routine.

When the LANCE does not own the next buffer and receive is still active after the current buffer is filled, the LANCE will update the status with BUFF error being set. OFLO (overflow) error may also get set if SILO overflows.

Receive Done

When receiver goes inactive (Done), the last byte of data has been read out of the SILO. The microprogram will check to see if the packet was a runt. If it is a runt, the receive buffer address pointer and byte count parameters are restored from

the previously loaded backup locations in the internal RAM. The microprogram then returns to the Polling routine. If the packet is not a runt, the receive status is updated in the ring descriptor.

Data Chain

If Byte Count of Current Buffer Equal 0 becomes true, it indicates that the receive buffer is full and the packet is not yet finished, which is the data chain case. The microprogram will update the receive status in the descriptor ring, and relinquish the buffer to the CPU. It will then check the next own bit. If the next own is false, which would be the case if there was only one buffer or if there was more than one buffer but the chip did not own the next one, the microprogram will wait for the receiver to go inactive. This indicates that no more data is arriving from the Ethernet. When the receiver goes inactive, the current RX status is updated, and the own bit is cleared.

If the chip owned the next buffer, the current receive buffer parameters in the internal RAM are updated from the next receive buffer parameters that had previously been loaded into the internal RAM. The microprogram will then check for end of the ring and update the address pointers accordingly. The microprogram will then go through the receive buffer lookahead section once, to try to acquire another receive buffer if one is available. The microprogram will finally get back to the wait loop until either receiver goes inactive, SILO overflow, or receive buffer overflow becomes true. There are two flags provided in the descriptor, STP (Start of Packet), and ENP (End of Packet), which allow the chip to mark the first and last buffers filled by the message. RMD_3 is not updated if its buffer is not the last buffer in the chain.

Receive DMA Routine

The Receive DMA routine is entered whenever there are 16 or more bytes of data in the SILO for transfer to memory during the reception. The routine is also entered when there are less than 16 bytes in the SILO and the receiver has gone inactive. This is to allow the SILO to empty at the end of the reception. Once entered, the Receive DMA routine will transfer 16 bytes of data to memory by doing 8 word transfers. These transfers are done on a single memory bus acquisition. This means that the chip will arbitrate through the $\overline{HOLD}$-$\overline{HLDA}$ sequence and then keep $\overline{HOLD}$ asserted for the duration of 8 transfers. The $\overline{READY}$ signal from the bus slave device is used to control the individual word transfers.

If the memory buffer starts on an odd address boundary, the first transfer will be 1 byte rather than 1 word (2 bytes). This routine is also used to transfer less than 16 bytes at the end of a reception depending upon the packet size, buffer addresses and data chaining.

Transmit Polling Routine

The transmit polling routine is entered from the polling routine to determine if a message has been scheduled on the transmit descriptor ring.

The routine begins by waiting for the TX Abort condition to finish if a TX Abort had occurred earlier. It then tests the status word of the ring descriptor entry. The routine tests the ownership of the ring buffer by reading the status word in the ring descriptor. If the Chip does not own the buffer, the microprogram returns to the polling routine. If it does own the buffer, this indicates a message is to be transmitted. The microprogram then tests the STP flag. If STP = 0, this buffer could be a fragment of a data chained packet that got an error in a previous buffer. The chip will release the buffer to the host by clearing the OWN bit. It will then update the ring address pointer and return to polling. In this manner, the chip skips over any bad transmit buffers on the ring, until it finds a buffer with both the OWN and STP bit being set.

If STP = 1, the microprogram performs memory transactions to acquire and store the address and byte count of the buffer in the Internal RAM. It then goes to the transmit routine to allow the transmission of the buffer.

The receive active trap is enabled during this routine to allow for processing of an incoming packet and termination of the transmit process.

Transmit Routine

The transmit routine is entered from the transmit polling routine when the microcode finds a buffer that it owns, indicating a message is scheduled to be transmitted. The routine is divided into three sections of microprogram, an initialization section, a buffer lookahead section, and a descriptor update section.

Upon entering the initialization section, the first thing the microprogram does is back up the buffer address and byte count in the event of a retry. It then enables the DMA engine to start filling the SILO and send the preamble. It then enters a wait loop until the transmitter is actually sending the bit stream. It then proceeds to the lookahead section. If the receiver became active while the microprogram was waiting for the transmitter to start, the transmission attempt is stopped and the microprogram goes to the receive routine via a TRAP.

Transmit Buffer Lookahead

Transmit lookahead occurs only when data chaining, and is done while the message is being transmitted from the SILO. In the lookahead section, the microprogram tests to determine if the current buffer it is transmitting has been marked with the end of the packet flag (ENP). If so, data chaining is not required. The microprogram enters a wait loop until either TX ERROR or TX DONE occurs. When DONE or ERROR or both finally set, the microprogram will report the error, if necessary, and then update the status word, update the ring address pointer and set the TINT bit in CSR_0. It will then return to the polling routine.

Transmit Data Chaining

There are two flags provided in transmit message descriptor 1 (TMD_1), STP (Start of Packet) and ENP (End of Packet) which mark the first and last buffers in the chain. The LANCE will, under microprogram control, continue to chain buffers pointed to by the sequential descriptors in the ring until the ENP flag is encountered. If the end of packet flag (ENP) is not set, data chaining is indicated. The microprogram first checks to see if it owns the next buffer. If not, the microprogram enters the descriptor update section and waits for TX DONE or TX ERROR. Eventually, an underflow error will occur because byte count overflow will occur without DONE having been set. Since there is no more data being written into the SILO and the transmitter is continuously reading data out of the SILO, the SILO will become empty and underflow will be set. This will cause the microprogram to branch out of the wait loop and update the descriptor with both BUFF and OVFL being set. When an underflow error occurs, the transmitter is disabled.

The LANCE owns the next buffer; the microprogram attempts to obtain the next buffer descriptor status, address, and byte count before entering a wait loop that looks for byte count overflow or TX ERROR. When byte count overflow does occur, the microprogram updates the descriptor and updates the internal current transmit buffer parameters. The microprogram will then return to the microcode that checks for the end of packet flag to sequence through the rest of the buffers in the data chain. If an error had occurred, the microprogram would report the error before updating the status word.

If an error needs to be reported, an error status word is written into the ring descriptor prior to writing the status word containing the "OWN" bit which releases the buffer. If no error is to be reported, the single word containing the "OWN" bit is written. The microprogram returns to the polling routine if the "ENP" flag is found or an error was reported. Otherwise, the microprogram returns to the lookahead sections.

Transmit DMA Routine

This routine is entered through a microtrap in the lookahead section of the transmit routine. The function of the routine is to move data out of local memory into the SILO. The trap is active when there are more than 16 free locations in the SILO and SILO underflow has not occurred.

Once entered, the transmit DMA routine will transfer 16 bytes of data from memory to the SILO by doing 8 word transfers. These transfers are done on a single memory bus acquisition. If the memory buffer starts on an odd addressing boundary, the first transfer will be 1 byte rather than 1 word (2 bytes). This routine is also used to transfer less than 16 bytes at the end of transmission depending upon the packet size, buffer addresses, and data chaining.

Retry Trap Routine

This routine is entered when a collision has been detected. The buffer address pointer is restored and the SILO is cleared to restore the read and write pointers. If there is a TX error, it indicates that 15 retransmissions have occurred (16 total attempts) or that the Disable Retry bit (DRTY) is set in the mode register. The microprogram then writes the status into the transmit descriptor ring, and returns to the polling routine to transmit the next packet. If there is no TX Error, the byte count is restored and the microprogram returns to the start of the transmit routine to attempt another transmission.

CSR Trap Routine

The CSR trap routine is entered only during the switch routine when the STOP bit of the CSR_0 is set. The function of the routine is to allow the access of CSR_1 and CSR_2 through an I/O transaction. The routine determines which CSR is being accessed, read or write, moves the data between the MDR and CDP RAM, and generates a Bus $\overline{\text{READY}}$ signal.

Memory Timeout Trap Routine

This trap is invoked whenever a memory transfer times out. That is, it does not receive $\overline{\text{READY}}$ within 25.6μsec after the assertion of the address on the bus.

The routine disables the receiver and transmitter by clearing the RXON and TXON bits in CSR_0.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Ambient Temperature with
Power Applied ..... −25°C to +125°C
Supply Voltage to Ground Potential
Continuous ..... −0.3V to +7V
Power Dissipation 2.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

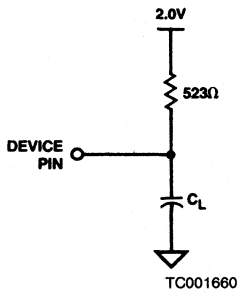
Commercial (C) Devices
Temperature ..... 0°C to +70°C
Supply Voltage ..... +4.75V to +5.25V
V_{SS} 0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

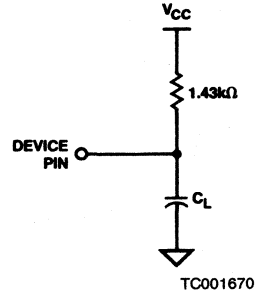
DC CHARACTERISTICS $T_A = 0$ to +70°C, $V_{CC} = +5V \pm 5\%$ unless otherwise specified

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input LOW Voltage		−0.5		0.8	Volts
V_{IH}	Input HIGH Voltage		2		V_{CC} + 0.5V	Volts
V_{OL}	Output LOW Voltage	I_{OL} = 3.2mA			0.5	Volts
V_{OH}	Output HIGH Voltage	I_{OH} = −0.4mA	2.4			Volts
I_{IL}	Input Leakage	V_{IN} = 0.4V to V_{CC}			±10	μA
C_{IN}	Input Capacitance				10	pF
C_{OUT}	Output Capacitance	F = 1MHz			10	pF
C_{IO}	Capacitance				20	pF

TEST LOAD DIAGRAMS FOR FUNCTIONAL AND AC TESTING



Test Load for All the Outputs and I/O Pins Except Pins 11, 17, 22



Test Load for Open Drain Outputs and Pins 11, 17, 22

C_L = 100pF for all pins except pins 26, 29.
C_L = 50pF for pins 26, 29.

SWITCHING CHARACTERISTICS $T_A = 0$ to 70°C, $V_{CC} = +5V \pm 5\%$ unless otherwise specified
BUS MASTER AND BUS SLAVE TIMING PARAMETERS

Number	Parameters	Description	Test Conditions	Min	Typ	Max	Units
1	t_{TCT}	TCLK Period		99		101	ns
2	t_{TCL}	TCLK LOW Time		45		55	ns
3	t_{TCH}	TCLK HIGH Time		45		55	ns
4	t_{TCR}	Rise Time of TCLK		0		8	ns
5	t_{TCF}	Fall Time of TCLK		0		8	ns
6	t_{TEP}	TENA Propagation Delay After the Rising Edge of TCLK	C_L = 50pF			95	ns
7	t_{TEH}	TENA Hold Time After the Rising Edge of TCLK	C_L = 50pF	5			ns
8	t_{TDP}	TX Data Propagation Delay After the Rising Edge of TCLK	C_L = 50pF			95	ns
9	t_{TDH}	TX Data Hold Time After the Rising Edge of TCLK	C_L = 50pF	5			ns
10	t_{RCT}	RCLK Period		85		118	ns
11	t_{RCH}	RCLK HIGH Time		38			ns
12	t_{RCL}	RCLK LOW Time		38			ns
13	t_{RCR}	Rise Time of RLCK		0		8	ns
14	t_{RCF}	Fall Time of RCLK		0		8	ns
15	t_{RDR}	RX Data Rise Time		0		8	ns
16	t_{RDF}	RX Data Fall Time		0		8	ns
17	t_{RDH}	RX Data Hold Time (RCLK to RX Data Change)		5			ns
18	t_{RDS}	RX Data Setup Time (RX Data Stable to the Rising Edge of RCLK)		60			ns
19	t_{DPL}	RENA LOW Time		120			ns
20	t_{CPH}	CLSN HIGH Time		80			ns
21	t_{DOFF}	Bus Master Driver Disable After Rising Edge of $\overline{HOLD}$		0		50	ns
22	t_{DON}	Bus Master Driver Enable After Falling Edge of $\overline{HLDA}$		0		250	ns
23	t_{HHA}	Delay to Falling Edge of $\overline{HLDA}$ from Falling Edge of $\overline{HOLD}$ (Bus Master)		0			ns
24	t_{RW}	$\overline{RESET}$ Pulse Width LOW		200			ns
25	t_{CYCLE}	Read/Write, Address/Data Cycle Time		600			ns
26	t_{XAS}	Address Setup Time to the Falling Edge of ALE		75			ns
27	t_{XAH}	Address Hold Time After the Rising Edge of $\overline{DAS}$		35			ns
28	t_{AS}	Address Setup Time to the Falling Edge of ALE		75			ns
29	t_{AH}	Address Hold Time After the Falling Edge of ALE		35			ns
30	t_{RDAS}	Data Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Read)		50			ns
31	t_{RDAH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Read)		0			ns
32	t_{DDAS}	Data Setup Time to the falling Edge of $\overline{DAS}$ (Bus Master Write)		0			ns
33	t_{WDS}	Data Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Write)		200			ns
34	t_{WDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Write)		35			ns
35	t_{SD01}	Data Driver Delay After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CRS 0, 3, RAP)		400		ns
36	t_{SD02}	Data Driver Delay Afte the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR 1, 2)		1200		ns
37	t_{SRDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave Read)		0		35	ns
38	t_{SWDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave Write)		0			ns
39	t_{SWDS}	Data Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave Write)		0			ns

SWITCHING CHARACTERISTICS (Cont.) $T_A = 0$ to 70°C, $V_{CC} = +5V \pm 5\%$ unless otherwise specified
BUS MASTER AND BUS SLAVE TIMING PARAMETERS

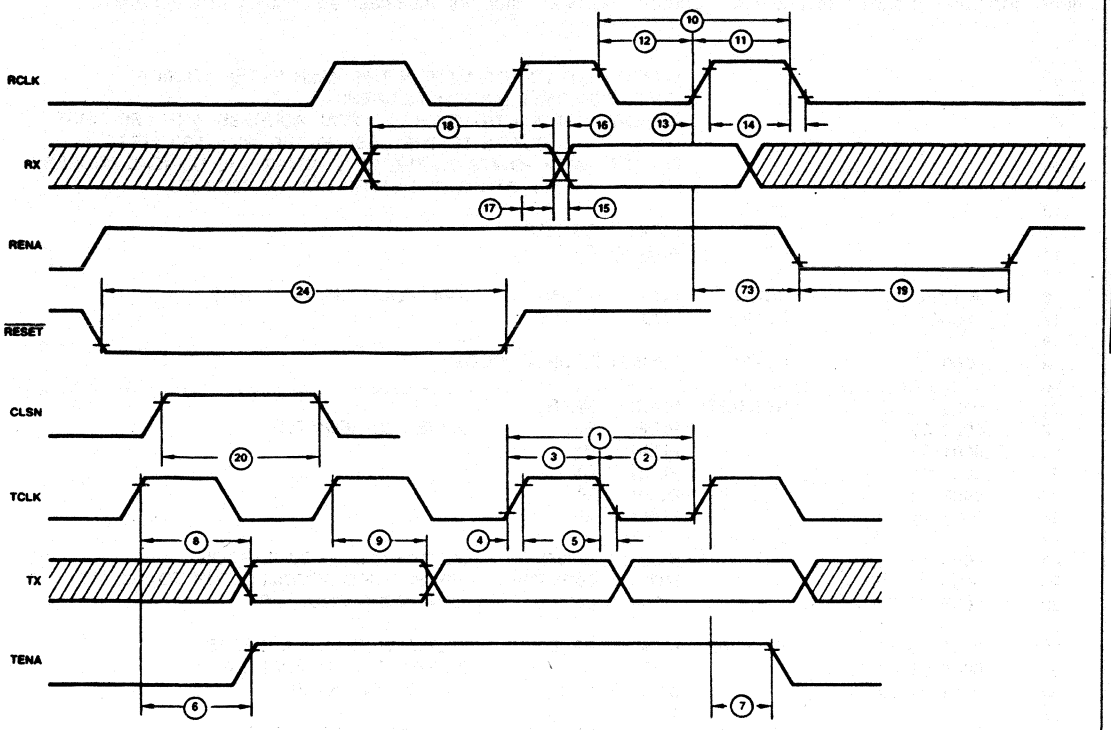
Number	Parameters	Description	Test Conditions	Min	Typ	Max	Units
40	t_{ALEW}	ALE Width HIGH		120			ns
41	t_{DALE}	Delay from Rising Edge of $\overline{DAS}$ to the Rising Edge of ALE		70			ns
42	t_{DSW}	$\overline{DAS}$ Width LOW		200			ns
43	t_{ADAS}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{DAS}$		80			ns
44	t_{RIDF}	Delay from the Rising Edge of DALO to the Falling Edge of $\overline{DAS}$ (Bus Master Read)		15			ns
45	t_{RDYS}	Delay from the Falling Edge of $\overline{READY}$ to the Rising Edge of $\overline{DAS}$	$t_{ARYD} = 300ns$	75		250	ns
46	t_{ROIF}	Delay from the Rising Edge of $\overline{DALO}$ to the Falling Edge of $\overline{DALI}$ (Bus Master Read)		15			ns
47	t_{RIS}	$\overline{DALI}$ Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Read)		135			ns
48	t_{RIH}	$\overline{DALI}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Read)		0			ns
49	t_{RIOF}	Delay from the Rising Edge of $\overline{DALI}$ to the Falling Edge of $\overline{DALO}$ (Bus Master Read)		55			ns
50	t_{OS}	$\overline{DALO}$ Setup Time to the Falling Edge of ALE (Bus Master Read)		110			ns
51	t_{ROH}	$\overline{DALO}$ Hold Time After the Falling Edge of ALE (Bus Master Read)		35			ns
52	t_{WDSI}	Delay from the Rising Edge of $\overline{DAS}$ to the Rising Edge of $\overline{DALO}$ (Bus Master Write)		35			ns
53	t_{CSH}	$\overline{CS}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
54	t_{CSS}	$\overline{CS}$ Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
55	t_{SAH}	ADR Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
56	t_{SAS}	ADR Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
57	t_{ARYD}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{READY}$ to Insure a Minimum Bus Cycle Time (600ns)				80	ns
58	t_{SRDS}	Data Setup Time to the Falling Edge of $\overline{Ready}$ (Bus Slave Read)		75			ns
59	t_{RDYH}	$\overline{READY}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master)		0			ns
60	t_{SR01}	$\overline{READY}$ Driver Turn On After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR 0, 3, RAP)		600		ns
61	t_{SR02}	$\overline{READY}$ Driver Turn On After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR 1, 2)		1400		ns
62	t_{SRYH}	$\overline{READY}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0		35	ns
63	t_{SRH}	READ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
64	t_{SRS}	READ Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
65	t_{CHL}	TCLK Rising Edge to Hold LOW or HIGH Delay				200	ns
66	t_{CAV}	TCLK to Address Valid				150	ns
67	t_{CCA}	TCLK Rising Edge to Control Signals Active				165	ns
68	t_{CALE}	TCLK Falling Edge to ALE LOW				150	ns
69	t_{CDL}	TCLK Falling Edge to DAS Falling Edge				150	ns
70	t_{RCS}	Ready Setup Time to TCLK		50			ns
71	t_{CDH}	TCLK Rising Edge to DAS HIGH				150	ns
72	t_{HCS}	HLDA Setup to TCLK		50			ns
73	t_{RENH}	RENA Hold Time After the Rising Edge of RCLK		40			ns

Notes: 1. Parameter #25 is not shown in the timing diagrams. It specifies the minimum bus cycle for a single DMA transfer.
2. The $\overline{READY}$ setup time before negation of $\overline{DAS}$ is a function of the synchronization time of $\overline{READY}$. The synchronization must occur within 100ns. Therefore, the setup time is 100ns plus any accumulated propagation delays. Ready slips occur on 100ns increments.

05698A

Refer to page 7-1 for Essential Information on Military Devices

SERIAL LINK TIMING



WF001531

Timing measurements are made at the following voltages, unless otherwise specified:

	High	Low
Output	2.0V	0.8V
Input	2.0V	0.8V
Float	V	0.5V

APPENDIX A

8086 computer program example to generate the hash filter, for multicast addressing in the LANCE.

```
                        ;       SUBROUTINE TO SET A BIT IN THE HASH FILTER FROM A
                        ;       GIVEN ETHERNET LOGICAL ADDRESS
                        ;       ON ENTRY SI POINTS TO THE LOGICAL ADDRESS WITH LSB FIRST
                        ;                DI POINTS TO THE HASH FILTER WITH LSB FIRST
                        ;       ON RETURN SI POINTS TO THE BYTE AFTER THE LOGICAL ADDRESS
                        ;                ALL OTHER REGISTERS ARE UNMODIFIED
                        ;
                                PUBLIC  SETHASH
                                ASSUME CS:CSE61
                        ;
= 1DB6          POLYL   EQU     1DB6H       ;CRC POLYNOMINAL TERMS
= 04C1          POLYH   EQU     04C1H
                        ;
0000            CSE61   SEGMENT PUBLIC 'CODE'
                        ;
0000            SETHASH PROC    NEAR
0000 50                 PUSH    AX          ;SAVE ALL REGISTERS
0001 53                 PUSH    BX
0002 51                 PUSH    CX
0003 52                 PUSH    DX
0004 55                 PUSH    BP
                        ;
0005 B8 FFFF            MOV     AX,0FFFFH   ;AX,DX = CRC ACCUMULATOR
0008 BA FFFF            MOV     DX,0FFFFH   ;PRESET CRC ACCUMULATOR TO ALL 1'S
000B B5 03              MOV     CH,3        ;CH = WORD COUNTER
                        ;
000D 8B 2C      SETH10: MOV     BP,[SI]     ;GET A WORD OF ADDRESS
000F 83 C6 02           ADD     SI,2        ;POINT TO NEXT ADDRESS
0012 B1 10              MOV     CL,16       ;CL = BIT COUNTER
                        ;
0014 8B DA      SETH20: MOV     BX,DX       ;GET HIGH WORD OF CRC
0016 D1 C3              ROL     BX,1        ;PUT CRC31 TO LSB
0018 33 DD              XOR     BX,BP       ;COMBINE CRC31 WITH INCOMING BIT
001A D1 E0              SAL     AX,1        ;LEFT SHIFT CRC ACCUMULATOR
001C D1 D2              RCL     DX,1
001E 81 E3 0001         AND     BX,0001H    ;BX = CONTROL BIT
0022 74 07              JZ      SETH30      ;DO NOT XOR IF CONTROL BIT = 0
                        ;
                        ;       PERFORM XOR OPERATION WHEN CONTROL BIT = 1
                        ;
0024 35 1D86            XOR     AX,POLYL
0027 81 F2 04C1         XOR     DX,POLYH
                        ;
002B 0B C3      SETH30: OR      AX,BX       ;PUT CONTROL BIT IN CRC0
002D D1 CD              ROR     BP,1        ;ROTATE ADDRESS WORD
002F FE C9              DEC     CL          ;DECREMENT BIT COUNTER
0031 75 E1              JNZ     SETH20
0033 FE CD              DEC     CH          ;DECREMENT WORD COUNTER
0035 75 D6              JNZ     SETH10
                        ;
                        ;       FORMATION OF CRC COMPLETE, AL CONTAINS THE REVERSED HASH
                        ;       CODE
                        ;
0037 B9 000A            MOV     CX,10
003A D0 E0      SETH40: SAL     AL,1        ;REVERSE THE ORDER OF BITS IN AL
003C D0 DC              RCR     AH,1        ;AND PUT IT IN AH
003E E2 FA              LOOP    SETH40
                        ;
                        ;       AH NOW CONTAINS THE HASH CODE
                        ;
0040 8A DC              MOV     BL,AH       ;BL = HASH CODE, BH IS ALREADY ZERO
0042 B1 03              MOV     CL,3        ;DIVIDE HASH CODE BY 8
0044 D2 EB              SHR     BL,CL       ;TO GET TO THE CORRECT BYTE
0046 B0 01              MOV     AL,01H      ;PRESET FILTER BIT
```

```
69      0048 80 E45 07              AND     AH,7H       ;EXTRACT BIT COUNT
70      004B 8A CC                  MOV     CL,AH
71      004D D2 E0                  SHL     AL,CL       ;SHIFT BIT TO CORRECT POSITION
72      004F 08 01                  OR      [DI + BX],AL ;SET IN HASH FILTER
73      0051 5D                     POP     BP
74      0052 5A                     POP     DX
75      0053 59                     POP     CX
76      0054 5B                     POP     BX
77      0055 58                     POP     AX
78      0056 C3                     RET
79                          ;
80      0057                SETHASH ENDP
81                          ;
82      0057                CSEG1   ENDS
83                          ;
84                                  END
```

Basic computer program example to generate the hash filter, for multicast addressing, in the LANCE.

```
100  REM
110  REM    PROGRAM TO GENERATE A HASH NUMBER GIVEN AN ETHERNET ADDRESS
120  REM
130  DEFINT A-Z
140  DIM A(47) : REM ETHERNET ADDRESS = 48 BITS
150  DIM C(32) : REM CRC REGISTER = 32 BITS
160  PRINT "ENTER STARTING ADDRESS'; : INPUT A$
170  IF LEN (A$) < > 12 THEN 160 : REM THE INPUT ADDRESS STARTING MUST BE 12 CHARS
180  REM
190  REM    UNPACK STARTING ADDRESS INTO ADDRESS ARRAY
200  REM
210  M = 0
220  FOR I = 0 TO 47 : A(I) = 0 : NEXT I
230  FOR N = 12 TO 1 STEP -1
240  Y$ = MID$ (A$,N,1)
250  IF Y$ = "0" THEN 420
260  IF Y$ = "1" THEN A(M) = 1 : GOTO 420
270  IF Y$ = "2" THEN A(M + 1) = 1 : GOTO 420
280  IF Y$ = "3" THEN A(M + 1) = 1 : A(M) = 1 : GOTO 420
290  IF Y$ = "4" THEN A(M + 2) = 1 : GOTO 420
300  IF Y$ = "5" THEN A(M + 2) = 1 : A(M) = 1 : GOTO 420
310  IF Y$ = "6" THEN A(M + 2) = 1 : A(M + 1) = 1 : GOTO 420
320  IF Y$ = "7" THEN A(M + 2) = 1 : A(M + 1) = 1 : A(M) = 1 : GOTO 420
330  A(M + 3) = 1
340  IF Y$ = "8" THEN 420
350  IF Y$ = "9" THEN A(M) = 1 : GOTO 420
360  IF Y$ = "A" THEN A(M + 1) = 1 : GOTO 420
370  IF Y$ = "B" THEN A(M + 1) = 1 : A(M) = 1 : GOTO 420
380  IF Y$ = "C" THEN A(M + 2) = 1 : GOTO 420
390  IF Y$ = "D" THEN A(M + 2) = 1 : A(M) = 1 : GOTO 420
400  IF Y$ = "E" THEN A(M + 2) = 1 : A(M + 1) = 1 : GOTO 420
410  IF Y$ = "F" THEN A(M + 2) = 1 : A(M + 1) = 1 : A(M) = 1
420  M = M + 4
430  NEXT N
440  REM
450  REM    PERFORM CRC ALGORITHM ON ARRAY A(0-47)
460  REM
470  FOR I = 0 TO 31 : C(I) = 1 : NEXT I
480  FOR N = 0 TO 47
490  REM LEFT CRC REGISTER BY 1
500  FOR I = 32 TO 1 STEP -1 : C(I) = C(I - 1) : NEXT I
510  C(0) = 0
520  T = C(32) XOR A(N) : REM T = CONTROL BIT
530  IF T < > THEN 600 : REM JUMP IF CONTROL BIT = 0
540  C(1) = C(1) XOR 1 : C(2) = C(2) XOR 1 : C(4) = C(4) XOR 1
550  C(5) = C(5) XOR 1 : C(7) = C(7) XOR 1 : C(8) = C(8) XOR 1
560  C(10) = C(10) XOR 1 : C(11) = C(11) XOR 1 : C(12) = C(12) XOR 1
```

```
570  C(16) = C(16) XOR 1 : C(22) = C(22) XOR 1 : C(23) = C(23) XOR 1
580  C(26) = C(26) XOR 1
590  C(0) = 1
600  NEXT N
610  REM
620  REM    CRC COMPUTATION COMPLETE, EXTRACT HASH NUMBER FROM C(0) TO C(5)
630  REM
640  HH = 32*C(0) + 16*C(1) + 8*C(2) + 4*C(3) + 2*C(4) + C(5)
650  PRINT "THE HASH NUMBER FOR ";A$;" IS ";HH
660  GOTO 160
```

MAPPING OF LOGICAL ADDRESS TO FILTER MASK

LAF Reg Bits Set	LAF Loc Dec	Destination Address Accepted (Hex)	LAF Reg Bits Set	LAF Loc Dec	Destination Address Accepted (Hex)
0	0	0000 0000 0085	0	32	0000 0000 0021
	1	0000 0000 00A5		33	0000 0000 0001
L	2	0000 0000 00E5	L	34	0000 0000 0041
A	3	0000 0000 00C5	A	35	0000 0000 0071
F	4	0000 0000 0045	F	36	0000 0000 00E1
	5	0000 0000 0065		37	0000 0000 00C1
0	6	0000 0000 0025	2	38	0000 0000 0081
	7	0000 0000 0005		39	0000 0000 00A1
	8	0000 0000 002B		40	0000 0000 008F
	9	0000 0000 000B		41	0000 0000 00BF
	10	0000 0000 004B		42	0000 0000 00EF
	11	0000 0000 006B		43	0000 0000 00CF
	12	0000 0000 00EB		44	0000 0000 004F
	13	0000 0000 00CB		45	0000 0000 006F
	14	0000 0000 008B		46	0000 0000 002F
15	15	0000 0000 00BB	15	47	0000 0000 000F
0	16	0000 0000 00C7	0	48	0000 0000 0063
	17	0000 0000 00E7		49	0000 0000 0043
	18	0000 0000 00A7		50	0000 0000 0003
	19	0000 0000 0087		51	0000 0000 0023
L	20	0000 0000 0007	L	52	0000 0000 00A3
A	21	0000 0000 0027	A	53	0000 0000 0083
F	22	0000 0000 0067	F	54	0000 0000 00C3
	23	0000 0000 0047		55	0000 0000 00E3
1	24	0000 0000 0069	3	56	0000 0000 00CD
	25	0000 0000 0049		57	0000 0000 00ED
	26	0000 0000 0009		58	0000 0000 00AD
	27	0000 0000 0029		59	0000 0000 008D
	28	0000 0000 00A9		60	0000 0000 000D
	29	0000 0000 0089		61	0000 0000 002D
	30	0000 0000 00C9		62	0000 0000 006D
15	31	0000 0000 00E9	15	63	0000 0000 004D

Am7992A

Serial Interface Adapter (SIA)

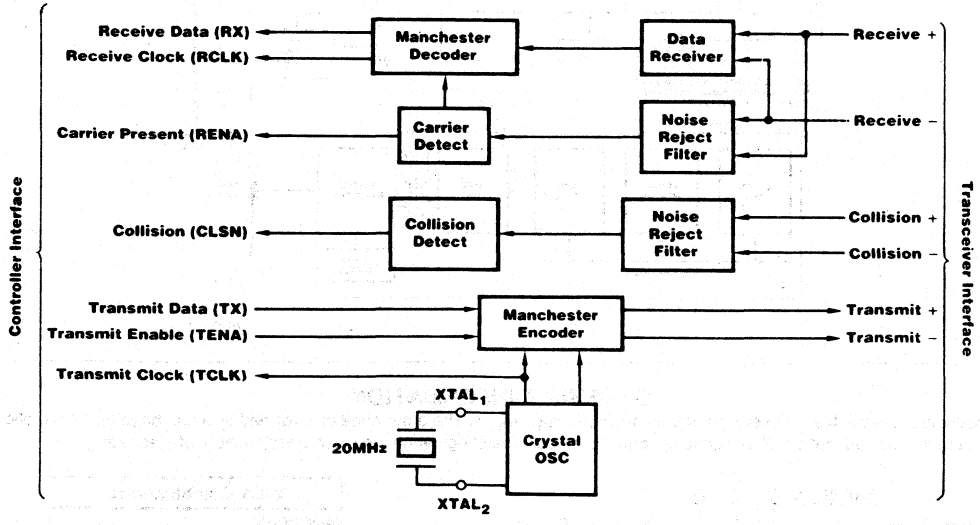
DISTINCTIVE CHARACTERISTICS

- Compatible with IEEE-802.3 Rev. D (10 Base 5 Type A, and 10 Base 2 Type B,"Cheapernet")/Ethernet specifications
- Crystal controlled Manchester Encoder
- Manchester Decoder acquires clock and data within six bit times with an accuracy of ±3ns
- Guaranteed carrier and collision detection squelch threshold limits
  - Carrier/collision detected for inputs greater than −275mV
  - No carrier/collision for inputs less than −175mV
- Input signal conditioning rejects transient noise
  - Transients < 10ns for collision detector inputs
  - Transients < 16ns for carrier detector inputs
- Receiver decodes Manchester data with up to ±20ns clock jitter (at 10MHz)
- TTL compatible host interface
- Transmit accuracy ±0.01% (without adjustments)

GENERAL DESCRIPTION

The Am7992A Serial Interface Adapter (SIA) is a Manchester Encoder/Decoder compatible with both Ethernet and IEEE–802.3 specifications. In an Ethernet/IEEE-802.3 application, the Am7992A interfaces the Am7990 Local Area Network Controller for Ethernet (LANCE) to the Ethernet transceiver cable, acquires clock and data within 6 bit-times, and decodes Manchester data with up to ±20ns phase jitter at 10MHz. SIA provides both guaranteed signal threshold limits and transient noise suppression circuitry in both data and collision paths to minimize false start conditions.

BLOCK DIAGRAM



BD002071

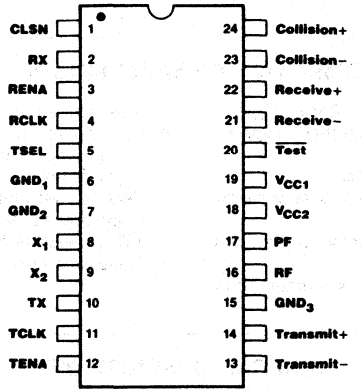
Serial Interface Adapter (SIA)

03378D

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

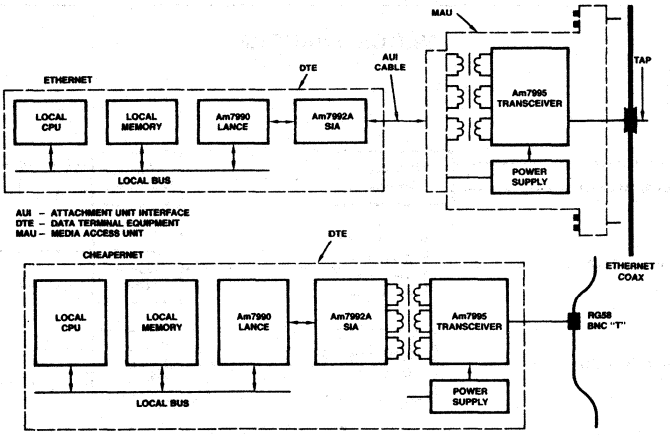
D-24-SLIM



CD001521

Note: Pin 1 is marked for orientation

TYPICAL ETHERNET NODE



AF000473

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am7992A D C B

Device Type

Screening Option
B = Burn-in
Blank = Standard processing

Temperature
C = Commercial 0°C to 70°C

Package
D = 24 pin SLIMDIP
P = plastic
L = leadless chip carrier

Valid Combinations	
Am7992A	

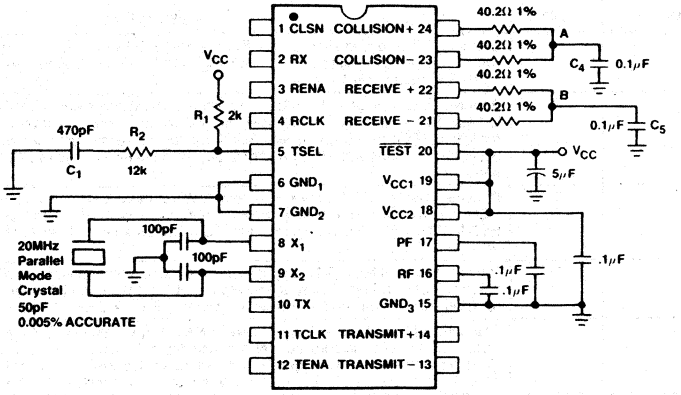
Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	CLSN	O	Collsion. A TTL active High output. Signals at the Collision ± terminals meeting threshold and pulse width requirements will produce a logic High at CLSN output. When no signal is present at Collision ±, CLSN output will be Low.
2	RX	O	Receive Data. A MOS/TTL output, recovered data. When there is no signal at Receive ± and $\overline{TEST}$ is high, RX is High. RX is actuated with RCLK and remains active until end of message. During reception, RX is synchronous with RCLK and remains active until end of message. During reception, RX is synchronous with RCLK and changes after the rising edge of RCLK. When $\overline{TEST}$ is Low, RX is enabled.
3	RENA	O	Receive Enable. A TTL active High output. When there is no signal at Receive ± and $\overline{TEST}$ is High, RENA is Low. Signals meeting threshold and pulse width requirements will produce a logic High at RENA. When Receive ± becomes idle, RENA returns to the Low state synchronous with the rising edge of RCLK.
4	RCLK	O	Receive Clock. A MOS/TTL output recovered clock. When there is no signal at Receive ± and $\overline{TEST}$ is High, RCLK is Low. RCLK is activated after the third negative data transition at Receive ±, and remains active until end of message. When $\overline{TEST}$ is Low, RCLK is enabled.
10	TX	I	Transmit. TTL compatible input. When TENA is High, signals at TX meeting setup and hold time to TCLK will be encoded as normal Manchester at Transmit + and Transmit –. TX High: Transmit + is negative with resept to Transmit – for first half of data bit cell. TX Low: Transmit + is positive with respect to Transmit – for first half of data bit cell.
12	TENA	I	Transmit Enable. TTL compatible input. Active high data encoder enable. Signals meeting setup and hold time to TCLK will allow encoding of Manchester data from TX to Transmit + and Transmit –.
11	TCLK	O	Transmit Clock. MOS/TTL output. TCLK provides symmetrical High and Low clock signals at data rate for reference timing of data to be encoded. It also provides clock signals for the controller chip (Am7990 – LANCE) and an internal timing reference for receive path voltage controlled oscillators.
14, 13	Transmit + Transmit –	O	Transmit. A differential line output. This line pair is intended to operate into terminated transmission lines. For signals meeting setup and hold time to TCLK at TENA and TX, Manchester clock and data are outputted at Transmit + /Transmit –. When operating into a 78Ω terminated transmission line, signaling meets the required output levels and skew for both Ethernet and IEEE 802.3 drop cables.
22, 21	Receive + Receive –	I	Receiver. A differential input. A pair of internally biased line receivers consisting of a carrier detect receiver with offset threshold and noise filtering to detect the signal, and a data recovery receiver with no offset for Manchester data decoding.
24, 23	Collision + Collision –	I	Collision. A differential input. An internally biased line receiver input with offset threshold and noise filtering. Signals at Collision± have no effect on data path functions.
5	TSEL	I/O	Transmit Mode Select. An open collector output and sense amplifier input. TSEL Low: Idle transmit state Transmit + is positive with respect to Transmit –. TSEL High: Idle transmit state Transmit + and Transmit – are equal, providing "zero" differential to operate transformer coupled loads. When connected with an RC network, TSEL is held Low during transmission. At the end of transmission the open collector output is disabled, allowing TSEL to rise and provide a smooth transmission from logic High to "zero" differential idle. Delay and output return to zero are externally controlled by the RC time constant TSEL.
8, 9	X_1, X_2	I	Biased Crystal Oscillator X_1 is the input and X_2 is the bypass port. When connected for crystal operation, the system clock which appears at TCLK is half the frequency of the crystal oscillator. X_1 may be driven from an external source of two times the data rate.
16	RF	O	Frequency Setting Voltage Controlled Oscillator (V_{CO}) Loop Filter. This loop filter output is a reference voltage for the receive path phase detector. It also is a reference for timing noise immunity circuits in the collision and receive enable path. Nominal reference V_{CO} gain is 1.25 TCLK frequency MHz/V.
17	PF	I	Receive Path V_{CO} Phase Lock Loop Filter. This loop filter input is the control for receive path loop damping. Frequency of the receive V_{CO} is internally limited to transmit frequency ±12%. Nominal receive V_{CO} gain is 0.25 reference V_{CO} gain MHz/V.
20	$\overline{TEST}$	I	Test Control. A static input that is connected to V_{CC} for normal Am7992A operation and to ground for testing of receive path function. When $\overline{TEST}$ is grounded RCLK and RX are enabled so that receive path loop may be functionally tested.
6	GND_1		High Current Ground
7	GND_2		Logic Ground
15	GND_3		Voltage Controlled Oscillator Ground
19	V_{CC1}		High Current and Logic Supply
18	V_{CC2}		Voltage Controlled Oscillator Supply

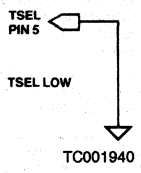
Am7992A
External Component Diagram



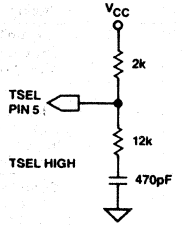
AF000450

Notes: 1. Connect R_1, C_1, for 0 differential nontransmit. Connect to ground for logic 1 differential nontransmit.
2. Pin 20 shown for normal device operation.
3. The inclusion of C_4 and C_5 is necessary to reduce the common-mode current draw of the terminating network when transformer coupling is not used at the SIA end. If transformer coupling is used at the SIA end, then C_4 and C_5 should be replaced by a short to ground.

TRANSMIT MODE SELECT (TSEL) CONNECTION



TC001940



TC001950

DETAILED DESCRIPTION

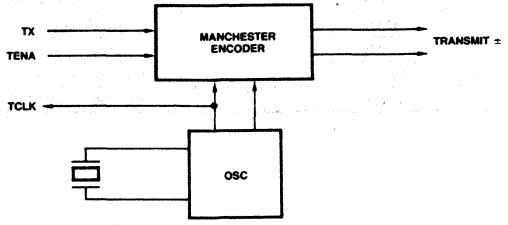
The Am7992A Serial Interface Adapter (SIA) has three basic functions. It is a Manchester Encoder/line driver in the transmit path, a Manchester Decoder with noise filtering and quick lock-on characteristics in the receive path, and a signal detect/converter (10MHz differential to TTL) in the collision path. In addition, the SIA provides the interface between the TTL logic environment of the Local Area Network Controller for Ethernet (LANCE) and the differential signaling environment in the transceiver cable.

TRANSMIT PATH

The transmit section encodes separate clock and NRZ data input signals meeting the set-up and hold time to TCLK at TENA and TX, into a standard Manchester II serial bit stream. The transmit outputs (transmit + /transmit –) are designed to operate into terminated transmission lines. When operating into a 78Ω terminated transmission line, signaling meets the required output levels and skew for both Ethernet and IEEE–802.3.

Transmitter Timing and Operation

A 20MHz fundamental mode crystal oscillator provides the basic timing reference in the SIA. It is divided by two to create the transmit clock reference (TCLK). Both clocks are fed into the Manchester Encoder to generate the 10MHz and 20MHz



AF003040

Figure 1. Transmit Section

transitions in the encoded data stream. The 10MHz clock, TCLK, is used by the SIA to internally synchronize transmit data (TX) and transmit enable (TENA). TCLK is also used as a stable bit rate clock by the receive section of the SIA and by other devices in the system (the Am7990 LANCE uses TCLK to drive its internal state machine). The oscillator may use an external .005% crystal or an external TTL level input as a reference. Transmit accuracy of .01% is achieved (no external adjustments are required).

TENA is activated when the first bit of data is made available on TX. As long as TENA remains High, signals at TX will be encoded as Manchester and will appear at Transmit + and Transmit –. When TENA goes Low, the differential transmit outputs go to one of two idle states:

- TSEL HIGH – The idle state of Transmit +/Transmit – yields "zero" differential to operate transformer coupled loads.(Figure 9A)
- TSEL LOW – In this idle state, Transmit + is positive to Transmit – (logical High). (Figure 9B)

RECEIVE PATH

The principle function of the receiver is the separation of the Manchester encoded data stream into clock and NRZ data.

Input Signal Conditioning

Before the data and clock can be separated it must be determined whether there is "real" data or unwanted noise at the transceiver interface. The Am7992A SIA carrier detection receiver provides a static noise margin of –175mV to –275mV for received carrier detection. These DC thresholds assure that no signal more positive than –175mV is ever decoded and that signals more negative than –275mV are always decoded. Transient noise of less than 10ns duration in the collision path and 16ns duration in the data path are also rejected.

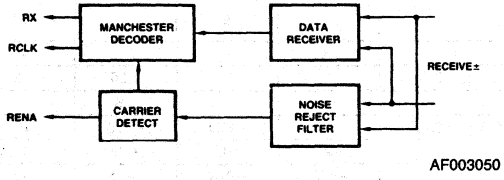


Figure 2. Receiver

The stage prevents unwanted idle state noise on the transceiver cable from causing "false starts" in the receiver. This helps assure a valid response to "real" data.

The receiver section (Figure 3) consists of two data paths. The receive data path is designed to be a zero threshold, high bandwidth receiver. The carrier detection receiver is similar, but with an additional bias generator. Only data amplitudes larger than the bias level are interpreted as valid data. The noise rejection filter prevents noise transients < 16ns from enabling the data receiver output. The collision detector similarly rejects noise transients < 10ns.

Receiver Section Timing

Receive Enable (RENA) is the "carrier present" indication established when a signal of sufficient amplitude (V_{IDC}) and duration (t_{RPWR}) is present at the receive inputs. Receive Clock (RCLK) and Receive Data (RX) become available after the third negative data transition at Receive+/Receive – inputs, and stay active until end of packet. During reception, RX is synchronous with RCLK changing after the rising edge of RCLK.

The receiver detects the end of a packet when the normal transition on the differential inputs cease. After the last Low-to-High transition, RENA goes Low and RCLK completes one last cycle, storing the last data bit. It then goes Low and remains Low. (See Receive End of Packet Timing diagrams.) When $\overline{TEST}$ is Low, RCLK does not go Low and stay Low but continues to run.

Receive Clock Control

To ensure quick capture of incoming data, the receiver phase-locked-loop is frequency locked to the transmit oscillator, and it phase locks to incoming data edges.

Clock and data are available within 6 bit times (accurate to within ±3ns). The SIA will decode jittered data of up to ±20ns (Figure 4).

Differential I/O Terminations

The differential input for the Manchester data (Receive ±) is externally terminated by two 40.2Ω ±1% resistors and one optional common mode bypass capacitor. The differential input impedance Z_{IDF} and the common mode input Z_{ICM} are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. The collision ± differential input is terminated in exactly the same way as the receive input (See External Component diagram).

Collision Detection

The Ethernet Transceiver detects collisions on the Ethernet and generates a 10MHz signal on the transceiver cable (Collision +/Collision–). This collision signal passes through an input stage which assures signal levels and pulse duration. When the signal is detected by the SIA, the SIA sets the CLSN line High. This condition continues for approximately 190ns after the last Low-to-High transition on Collision +/Collision–.

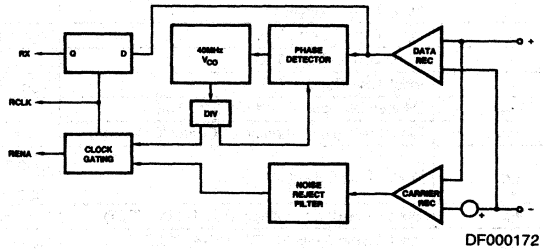


Figure 3. Receiver Section Detail

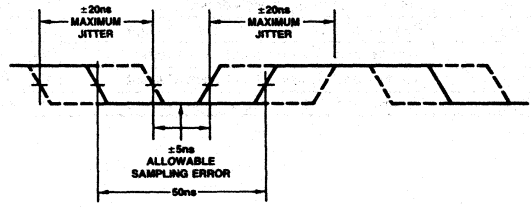


Figure 4. Maximum Jitter Impact on Sampling

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Ambient Temperature with
Power Applied ..... 0 to +70°C
Supply Voltage to Ground Potential
Continuous ..... +7.0V
DC Voltage Applied to Outputs For
High Output State ..... −0.5V to +V_{CC} max
DC Input Voltage (Logic Inputs) ..... +5.5V
DC Input Voltage (Receive/Collision) ..... −6 to +6V
Transmit ±Output Current ..... −50 to +5mA
DC Output Current, Into Outputs 100mA
DC Input Current (Logic Inputs) ..... ±30mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5.0V±5%	N/A

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS unless otherwise specified: COM'L T_A = 0 to 70°C, V_{CC} = 5.0V±5%, T_{OSC} = 50ns

Parameters	Description		Test Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage RX, RENA, CLSN,TCLK		I_{OH} = −1.0mA	2.4	3.4		V
V_{OL}	Output Low Voltage RCLK, TCLK, RENA RX, CLSN, TSEL		I_{OL} = 16mA		0.36	0.5	V
			I_{OL} = 1mA		0.25	0.4	
V_{OD}	Differential Output Voltage (Transmit+) − (Transmit−)	V_O	RL = 78Ω	550	670	770	mV
		$\overline{V_O}$		−550	−670	−770	
$V_{OD\ OFF}$	Transmit Differential Output Idle Voltage		RL = 78Ω, TSEL = High	−20	0.5	20	mV
$I_{OD\ OFF}$	Transmit Differential Output Idle Current			−0.5	±00.1	0.5	mA
V_{CMT}	Common Mode Output Transmit		RL = 78Ω	0	2.5	5	V
V_{ODI}	Differential Output Voltage Imbalance (Transmit±) $\|V_O - \overline{V_O}\|$				5	20	mV
V_{IH}	Input High Voltage TTL			2.0			V
I_{IH}	Input High Current TTL		V_{CC} = MAX, V_{IN} = 2.7V			+50	µA
V_{IL}	Input Low Voltage TTL					0.8	V
I_{IL}	Input Low Current TTL		V_{CC} = MAX, V_{IN} = 0.4V		−270	−400	µA
V_{IRD}	Differential Input Threshold (Rec Data)			−25	0	+25	mV
V_{IDC}	Differential Input Threshold (Carrier/Collision±)			−175	−225	−275	mV
I_{CC}	Power Supply Current		t_{OSC} = 50ns		125	180	mA
			t_{OSC} = 50ns, T_A = Max			160	
V_{IB}	Input Breakdown Voltage (Tx, TENA, TEST)		I_I = 1mA	5.5			V
V_{IC}	Input Clamp Voltage (TX, TENA, TEST)		I_{IN} = −18mA			−1.2	V
I_{SCO}	RX, TCLK, CLSN, RENA Short Circuit Current			−40	−80	−150	mA
R_{IDF}	Differential Input Resistance		V_{CC} = 0 to Max	6k	8.4k	13k	Ω
R_{ICM}	Common Mode Input Resistance		V_{CC} = 0 to Max	1.5k	2.1k	3.25k	Ω
V_{ICM}	Receive and Collision Input Bias Voltage		I_{IN} = 0	2.5	3.5	4.2	V
I_{ILD}	Receive and Collision Input Low Current		V_{IN} = −6V	−0.32	−1.06	−1.64	mA
I_{IHD}	Receive and Collision Input High Current		V_{IN} = 6V	+0.14	+0.6	+1.10	mA
I_{IHZ}	Receive and Collision Input High Current		V_{CC} = 0, V_{IN} = +6V	0.4	1.28	1.86	mA

SWITCHING CHARACTERISTICS unless otherwise specified:
COM'L T_A = 0 to +70°C, V_{CC} = 5.0V±5%, T_{OSC} = 50ns

Parameters	Description	Test Conditions	Min	Typ	Max	Units
RECEIVER SPECIFICATION						
t_{RCT}	RCLK Cycle Time		85	100	118	ns
t_{RCH}	RCLK High Time		38	50		ns
t_{RCL}	RCLK Low Time		38	50		ns
t_{RCR}	RCLK Rise Time	C_L = 50pF Figure 12a (Note 1)	0	2.5	8	ns
t_{RCF}	RCLK Fall Time		0	2.5	8	ns
t_{RDR}	RX Rise Time		0	2.5	8	ns
t_{RDF}	RX Fall Time		0	2.5	8	ns
t_{RDH}	RX Hold Time (RCLK ↑ to RX Change)		5	8		ns
t_{RDS}	RX Prop Delay (RCLK ↑ to RX Stable)			8	25	ns
t_{DPH}	RENA Turn-On Delay (V_{IDC} on Receive ± to $RENA_H$)	Figures 5, 11a and 15		50	180	ns
t_{DPO}	RENA Turn-Off Delay (V_{IDH} on Receive ± to $RENA_L$)	Figures 6 and 15		140	160	ns
t_{DPL}	RENA Low Time	Figure 6	120	200		ns
t_{RPWR}	Receive ± Input Pulse Width to Reject (Input < V_{IDC} Min)	Figures 11a and 15		30	16	ns
t_{RPWO}	Receive ± Input Pulse Width to Turn-On (Input < V_{IDC} Max)		40	30		ns
t_{RLT}	Decoder Acquisition Time	Figure 5		550	600	ns
COLLISION SPECIFICATION						
t_{CPWR}	Collision Input Pulse Width to Reject (Input < V_{IDC} Min)	Figures 11b and 15		18	10	ns
t_{CPWO}	Collision Input Pulse Width to Turn-on (Collision ± Exceeds V_{IDC} Max)		26	18		ns
t_{CPWE}	Collision Input to Turn-Off CSLN (Input > V_{IDC} Max)		80	117		ns
t_{CPWN}	Collision Input to Not Turn-Off CLSN (Input > V_{IDC} Min)			117	160	ns
t_{CPH}	CLSN Turn-On Delay (V_{IDC} Max on Collision ± to $CLSN_H$)	Figures 10, 11b and 15		33	180	ns
t_{CPO}	CLSN Turn-Off Delay (V_{IHD} Min on Collision ± to $CLSN_L$)			133	160	ns
TRANSMITTER SPECIFICATION						
t_{TCL}	TCLK Low Time	t_{OSC} = 50ns Figure 12b	45	50	55	ns
t_{TCH}	TCLK High Time		45	50	55	ns
t_{TCR}	TCLK Rise Time		0	2.5	8	ns
t_{TCF}	TCLK Fall Time		0	2.5	8	ns
t_{TDS}, t_{TES}	TXD and TENA Setup Time	Figure 12b	5	1.1		ns
t_{TDH}, t_{TEH}	TXD and TENA Hold Time		5	−1.1		ns
t_{TOCE}	Transmit ± Output, (Bit Cell Center to Edge)	Figures 9a, 9b and 14	49.5	50	50.5	ns
t_{OD}	TCLK High to Transmit ± Output			80	100	ns
t_{TOR}	Transmit ± Output Rise Time	20 – 80% Figure 12b and 14	1	2	5	ns
t_{TOF}	Transmit ± Output Fall Time		1	2	5	ns
V_{OD}	Undershoot Voltage at Zero Differential Point on Transmit Return to Zero (End of Message)				−100	mV

Note 1. Assumes equal capacitance loading on RCLK and RX.

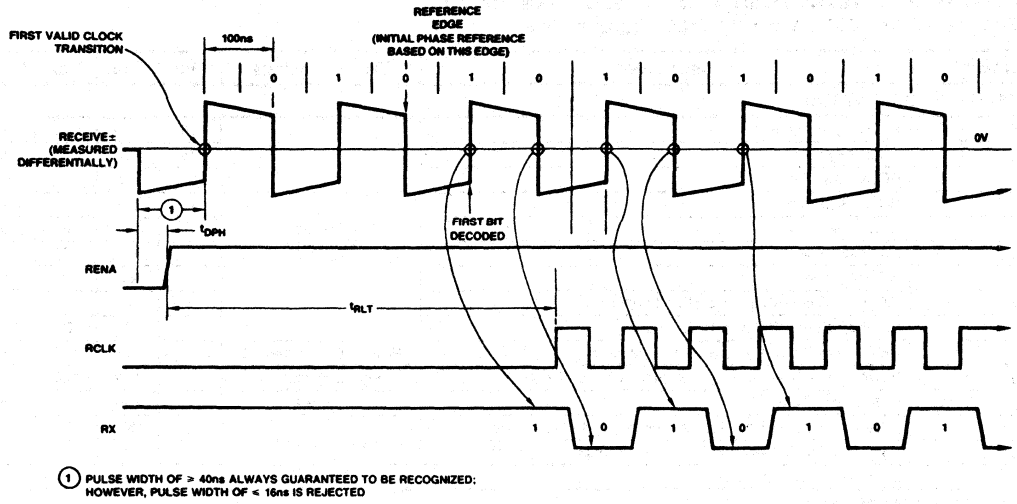


Figure 5. Receiver Timing – Start of Packet

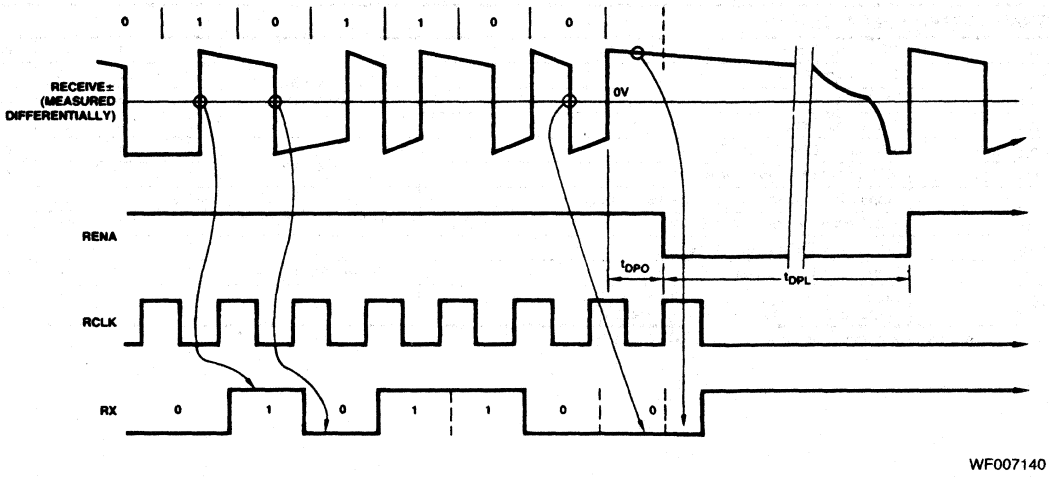


Figure 6. Receiver Timing – End of Packet (Last Bit = 0)

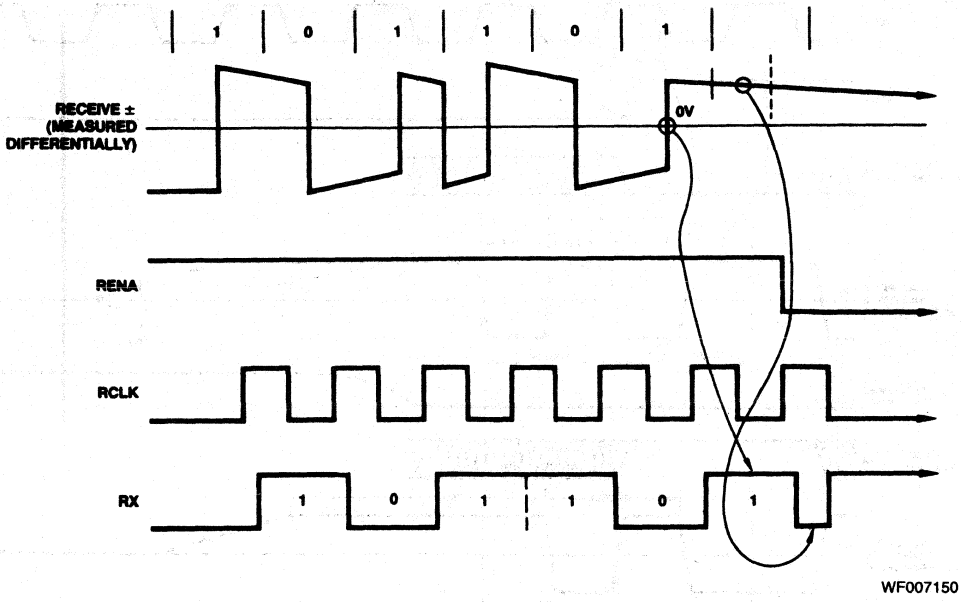


Figure 7. Receiver Timing – End of Packet (Last Bit = 1)

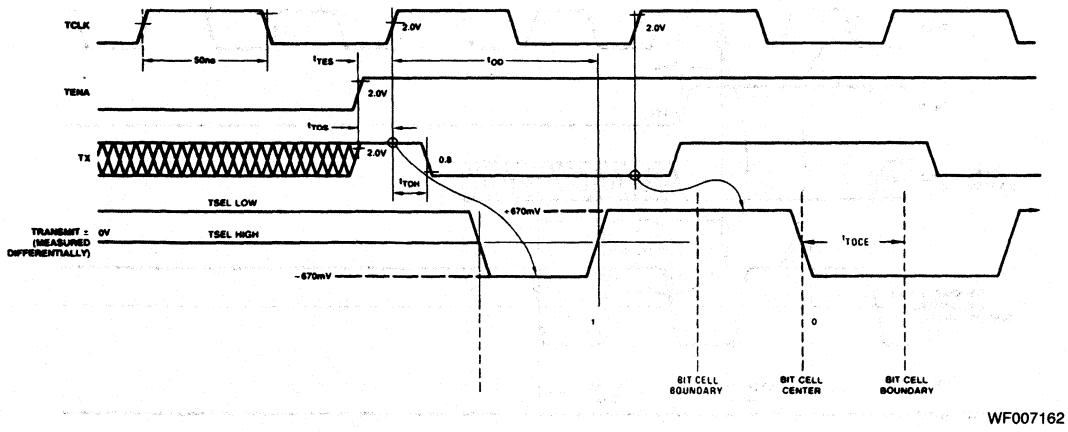


Figure 8. Transmitter Timing – Start of Transmission (TSEL Low, TSEL High)

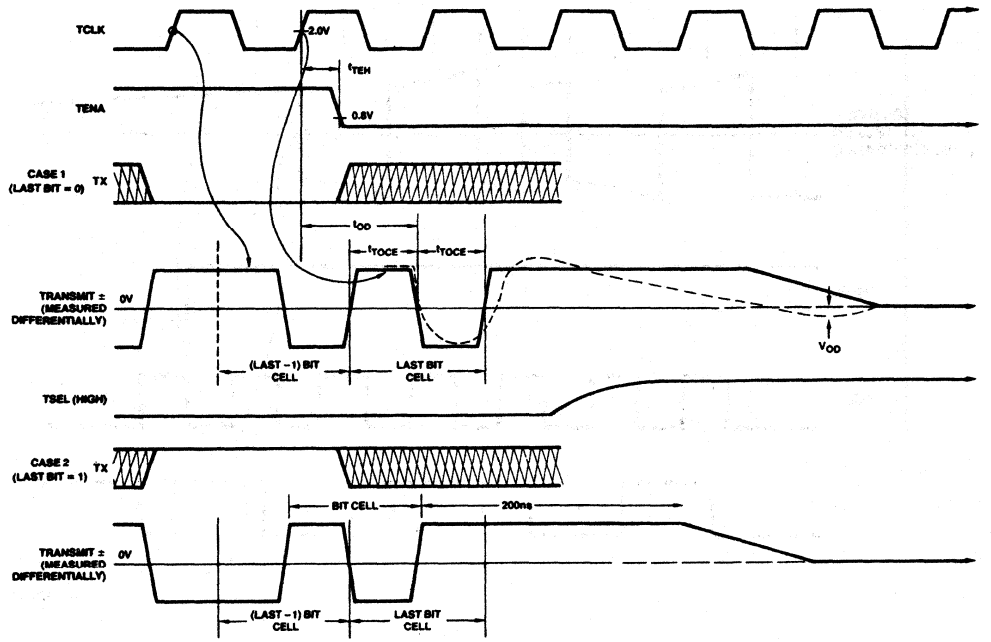


Figure 9a. Transmitter Timing – End of Transmission (TSEL High)

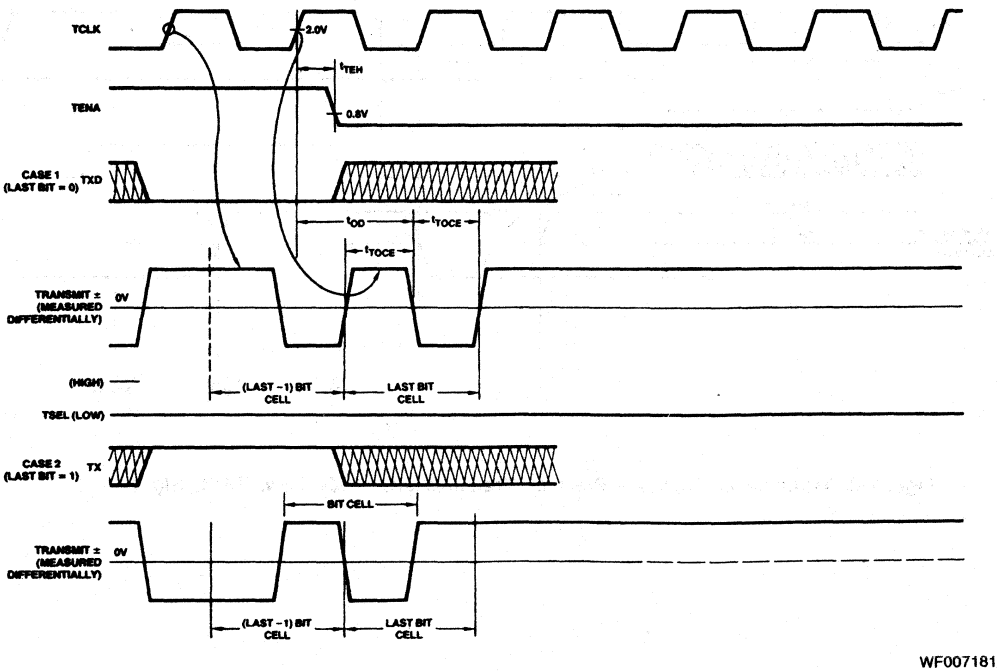


Figure 9b. Transmitter Timing – End of Transmission (TSEL Low)

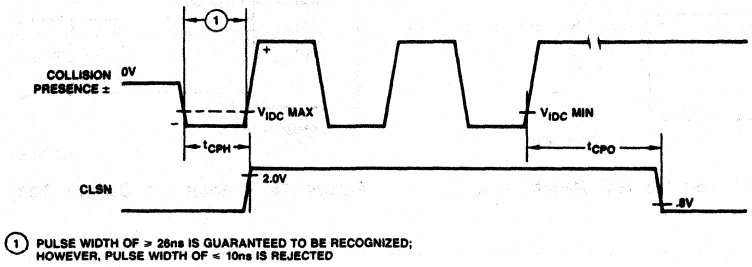


Figure 10. Collision Timing

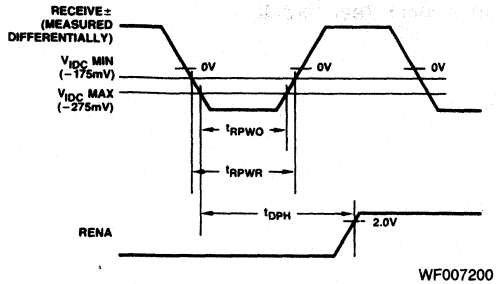


Figure 11a. Receive ±Input Pulse Width Timing

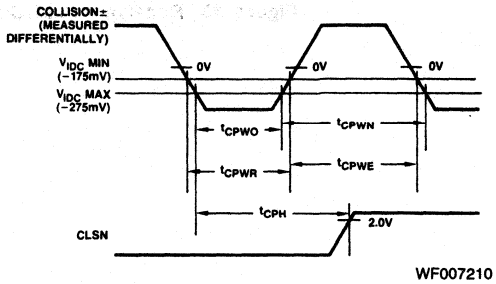


Figure 11b. Collision ±Input Pulse Width Timing

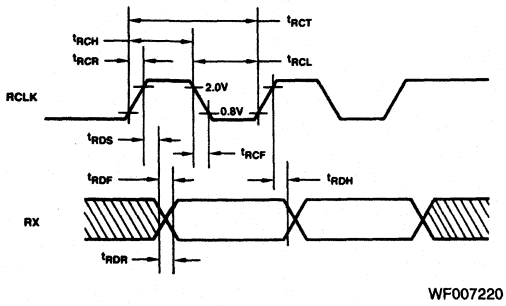


Figure 12a. RCLK and RX Timing

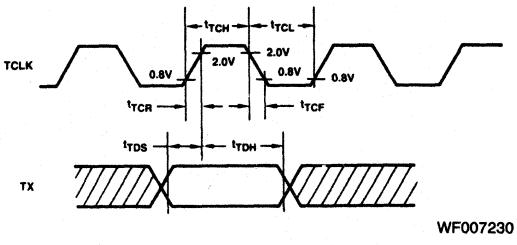
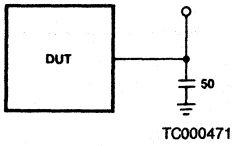
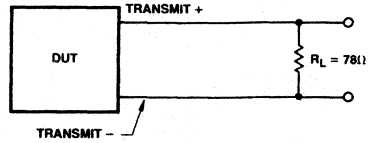


Figure 12b. TCLK and TX Timing



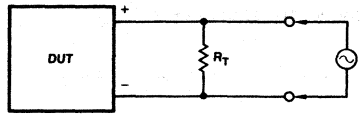
TC000471

Figure 13. Test Load for RX, RENA, RCLK, TCLK



TC000461

Figure 14. Transmit ± Output Test Circuit



TC000481

Figure 15. Receive ± and Collision ± Input Test Circuit

AmZ8016

DMA Transfer Controller

DISTINCTIVE CHARACTERISTICS

- Two independent multi-function channels
- Automatic loading/reloading of control parameters by each channel
- Optional automatic chaining of operations
- Channel interleave operations
- Masked data pattern matching for search operations
- Vectored interrupts on selected transfer conditions
- Base registers for repetitive operations

GENERAL DESCRIPTION

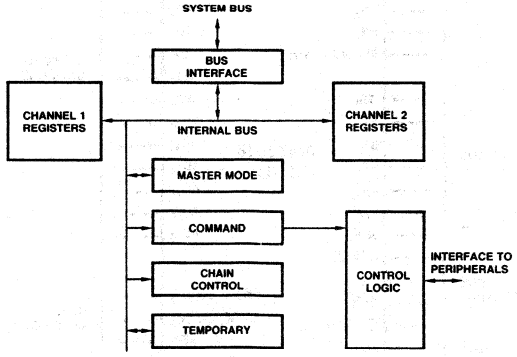
The AmZ8016* DMA Transfer Controller (DTC) is a high performance peripheral interface circuit for Z8000 processor systems. In addition to providing data block transfer capability between memory and peripherals, each of the DTC's two channels can perform peripheral-to-peripheral as well as memory-to-memory transfer. A special Search Mode of Operation compares data read from a memory or peripheral source to the content of a pattern register.

For all DMA operations (search, transfer, and transfer-and-search), the DTC can operate with either byte or word data sizes. In some system configurations, it may be necessary to transfer between word-organized memory and a byte-oriented peripheral. The DTC provides a byte packing/unpacking capability through its byte-word funneling transfer or transfer-and-search option. Some DMA applications may continuously transfer data between the same two locations. These applications may not require the flexibility inherent in reloading registers from memory tables. To service these repetitive DMA operations, base registers are provided on each channel which reinitialize the current source and destination Address and Operation Count registers. To change the data transfer direction under CPU control, provision is made for reassigning the source address as a destination and the destination as a source, eliminating the need for actual reloading of these address registers.

Frequently, DMA devices must interface to slow peripherals or slow memory. In addition to providing a hardware $\overline{\text{WAIT}}$ input, the AmZ8016 DTC allows the user to select independently, for both source and destination addresses, automatic insertion of 0, 1, 2 or 4 wait states. The user may even disable the $\overline{\text{WAIT}}$ input pin function altogether and use these software programmed wait states exclusively.

High throughput and powerful transfer options are of limited usefulness if a DMA requires frequent reloading by the host CPU. The AmZ8016 DTC minimizes CPU interactions by allowing each channel to load its control parameters from memory into the channel's control registers. The only action required of the CPU is to load the address of the control parameter table into the channel and issue an instruction to start this register loading operation.

BLOCK DIAGRAM



BD003460

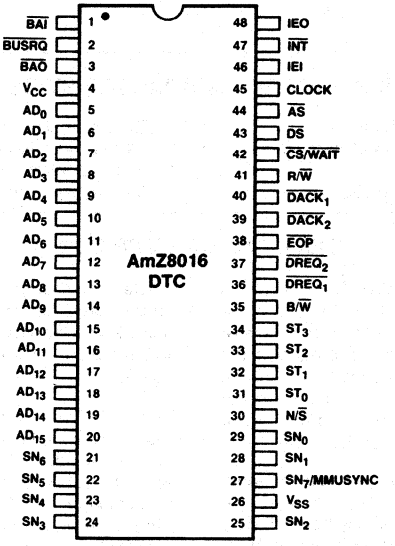
RELATED PRODUCTS

Part No.	Description
AmZ8016	Application Manual (contains detailed application configuration and software example).

*Z8000 is a trademark of Zilog, Inc.

Refer to page 7-1 for Essential Information on Military Devices

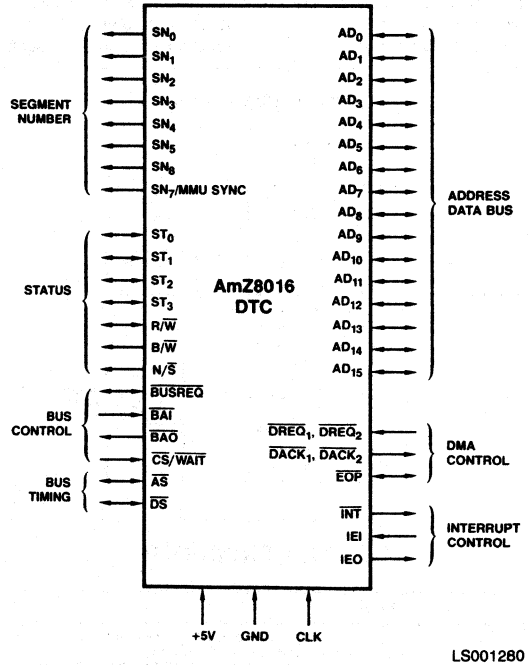
CONNECTION DIAGRAM
Top View
D-48, P-48



CD005300

Note: Pin 1 is marked for orientation

LOGIC SYMBOL

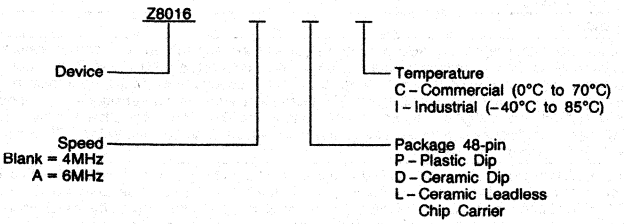


LS001280

Refer to page 7-1 for Essential Information on Military Devices

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Z8016	DC, PC ADC, APC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	VCC		+5 Power Supply.
26	VSS		Ground.
45	CLOCK	I	(Clock). The Clock signal controls the internal operations and the rates of data transfers. It is usually derived from a master system clock or the associated CPU clock. The Clock input requires a high voltage input signal. When the DTC is used with an MMU, they must both be driven from the same clock signal. Many DTC input signals can make transitions independent of the DTC clock; these signals can be asychronous to the DTC clock. On other signals, such as $\overline{WAIT}$ inputs, transitions must meet set-up and hold requirements relative to the DTC clock. See the timing diagrams for details.
5-20	AD_0-AD_{15}	I/O	(Address/Data Bus). The Address/Data Bus is a time-multiplexed, bidirectional, active High, three-state bus used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to O. AD_0 is the least significant bit position and AD_{15} is the most significant. The presence of addresses is defined by the timing edge of $\overline{AS}$, and the asserted or requested presence of data is defined by the $\overline{DS}$ signal. The status output lines ST_0-ST_3 indicate the type of transaction, either memory or I/O. When the DTC is in control of the system bus, it dominates the AD Bus; when the DTC is not in control of the system bus, the CPU or other external devices dominate the AD Bus. The presence of address or data on the AD_0-AD_{15} bus is defined only by $\overline{AS}$ and $\overline{DS}$. When the DTC is not in control of the bus; there is no required relation between the presence of address or data and the DTC clock. This allows the DTC to be used with a system bus which does not have a bussed clock signal.
44	$\overline{AS}$	I/O	(Address Strobe). Address Strobe is a bidirectional, active-low, three-state signal. A LOW-to-HIGH transition on this signal while $\overline{DS}$ is HIGH indicates that the AD_0-AD_{15} bus contains address information. During a DMA operation when the DTC is in control of the system, $\overline{AS}$ is an output generated by the DTC to indicate that a valid address is on AD_0-AD_{15}. The address information output by the DTC is stable prior to the LOW-to-HIGH $\overline{AS}$ transition. When the DTC is not in control of the system bus and the external system is transferring information to the DTC or from it, the DTC samples address information from the AD_0-AD_{15} bus on the LOW-to-HIGH $\overline{AS}$ transition. There are no timing requirements between $\overline{AS}$ as an input and the DTC clock; this allows use of the DTC with a system bus which does not have a bussed clock. If $\overline{AS}$ and $\overline{DS}$ are simultaneously LOW, the DTC is reset.
43	$\overline{DS}$	I/O	(Data Strobe). Data Strobe is a bidirectional, active-low, three-state signal. A LOW on this signal while $\overline{AS}$ is HIGH indicates that the AD_0-AD_{15} bus is being used for data transfer. When the DTC is not in control of the system bus and the external system is transferring information to or from the DTC, $\overline{DS}$ is a timing input used by the DTC to move data to or from the AD_0-AD_{15} bus. Data is written into the DTC by the external system on the LOW-to-HIGH $\overline{DS}$ transition. Data is read from DTC by the external system while $\overline{DS}$ is LOW. There are no timing requirements between $\overline{DS}$ as an input and the DTC clock; this allows use of the DTC with a system bus which does not have a bussed clock. During a DMA operation when the DTC is in control of the system, $\overline{DS}$ is an output generated by the DTC and used by the system to move data to or from the $AD_0 - AD_{15}$ bus. When the DTC has bus control, it writes to the external system by placing data on the $AD_0 - AD_{15}$ bus before the HIGH-to-LOW $\overline{DS}$ transition and holding the data stable until after the LOW-to-HIGH $\overline{DS}$ transition; while reading from the external system, the LOW-to-HIGH transition of $\overline{DS}$ latches data into the temporary register of the DTC (see timing diagram).
31-34	$ST_0 - ST_3$	I/O	(Status). The four Status lines are three-state, bidirectional signals containing coded information regarding the current bus transaction. When the DTC is not in control of the system bus, $ST_0 - ST_3$ are inputs and are used to detect interrupt and segement trap acknowledge cycles. There are not timing requirements between transitions on the $ST_0 - ST_3$ input and the DTC clock; input transitions on $ST_0 - ST_3$ are only defined relative to $\overline{AS}$ and $\overline{DS}$. When the DTC is in control of the system bus, the $ST_0 - ST_3$ lines are outputs which indicate the type of memory of I/O transition being performed. The status codes decoded and generated by the DTC are indicated in Figure 1 by the letters D and G, respectively.
41	R/$\overline{W}$	I/O	(Read/$\overline{Write}$).Read/Write is a bidirectional, three-state signal. Read polarity is HIGH and WRITE polarity is LOW. R/$\overline{W}$ indicates the data direction of the current bus transaction, and is stable starting when $\overline{AS}$ goes LOW until the bus transaction ends (see timing diagram). When the DTC is not in control of the system bus and the external system is transferring information to or from the DTC, R/$\overline{W}$ is a status input used by the DTC to determine if data is entering or leaving on the $SD_0 - AD_{15}$ bus during $\overline{DS}$ time. In such a case, Read (HIGH) indicates that the system is requesting data from the DTC and Write (LOW) indicates that the system is presenting data to the DTC. There are no timing requirements between R/$\overline{W}$ as an input and the DTC clock; transitions on R/$\overline{W}$ as an input are only defined relative to $\overline{AS}$ and $\overline{DS}$. When DTC is in control of the system bus, R/$\overline{W}$ is an output generated by the DTC, with Read indicating that data is being requested from the addressed location or devices and Write indicating that data is being presented to the addressed location or device. Flyby DMA operations are a special case where R/$\overline{W}$ is valid for the normally addressed memory or peripheral locations and must be interpreted in reverse by the "Flyby" peripheral that uses it.
30	N/$\overline{S}$	O	(Normal/$\overline{System}$, 3-State). Normal/System is a three-state output activated only when the DTC is in control of the system bus. This signal is used to indicate which memory space is being accessed. The N/$\overline{S}$ pin is HIGH for normal memory and LOW for system memory. System space is always indicated for I/O cycles.
35	B/$\overline{W}$	O	(Byte/$\overline{Word}$, 3-State). This output indicates the type of data transferred on the AD bus. HIGH indicates a byte (8-byte) and LOW indicates a word (16-bit) transfer. This output is activated when $\overline{AS}$ goes LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the DTC is always a byte address, even though the memory organized as 16-bit words. All word-sized data are word aligned and must be addressed by even addresses ($A_0 = 0$). When addressing byte transactions, the least significant address bit determines which byte is needed; an even address specifies the most significant byte ($AD_8 - AD_{15}$), and an odd address specifies the least significant byte ($AD_0 - AD_7$). (Note that the higher address specifies the less significant byte!) This addressing mechanism applies to memory accesses as well as I/O and special I/O accesses. When the DTC is a slave, it ignores the B/$\overline{W}$ signal.

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
42	$\overline{CS}/\overline{WAIT}$	I	(Chip Select/Wait). When the DTC is not in control of the system bus, this pin serves as an active-low Chip Select ($\overline{CS}$) input. A CPU or other external device uses $\overline{CS}$ to activate the DTC for reading and writing of its internal registers. $\overline{CS}$ may be held LOW for multiple transfers to and/or from the DTC, provided $\overline{AS}$ and $\overline{DS}$ are toggled for each transfer. There are no timing requirements between the $\overline{CS}$ input and the DTC clock; the $\overline{CS}$ input timing requirements are only defined relative to $\overline{AS}$. When the DTC is in control of the system bus, this pin serves as an active-low $\overline{WAIT}$ input. Slow memories and peripheral devices may use $\overline{WAIT}$ to extend $\overline{DS}$ during bus transfers. Unlike the $\overline{CS}$ input, transitions on the $\overline{WAIT}$ input must meet certain timing requirements relative to the DTC clock. See the timing diagram for details. The Wait function may be disabled using a control bit in the Master Mode register; in which case, the input is treated as an active-low Chip Select only and is ignored when the DTC is in control of the system bus.
2	$\overline{BUSRQ}$	I/O	(Bus Request). Bus Request is an active-low, open-drain, bidirectional signal used by the DTC to obtain control of the bus from the CPU. Before driving $\overline{BUSRQ}$ active, the DTC samples this line to insure that another request is not already being made by another device. $\overline{BUSRQ}$ lines from multiple devices are wire-ORed together externally with a common pull-up resistor of 1.8 kΩ or more. Since the DTC internally synchronizes the sampled $\overline{BUSRQ}$ signal, transitions on $\overline{BUSRQ}$ may be asynchronous to the DTC clock.
1	$\overline{BAI}$	I	(Bus Acknowledge In). $\overline{BAI}$ is an active-low asynchronous input indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since $\overline{BAI}$ is internally synchronized by the DTC before being used, transitions on $\overline{BAI}$ do not have to be synchronous with the DTC clock. The $\overline{BAI}$ input is usually connected to the $\overline{BUSAK}$ line from the CPU or to the $\overline{BAO}$ output from a higher-priority device in the Bus Request daisy chain. $\overline{AS}$ and $\overline{DS}$ must both be HIGH during the HIGH-to-LOW transition of $\overline{BAI}$.
3	$\overline{BAO}$	O	(Bus Acknowledge Out). $\overline{BAO}$ is an active-low output which indicates that $\overline{BAI}$ is active and that the DTC is not currently in control of the bus. This signal is intended for use by lower priority devices on the Bus Request daisy chain.
47	$\overline{INT}$	O	(Interrupt). Interrupt is an active-low, open-drain output used to interrupt the CPU. It may be connected to any of the CPU interrupt inputs and may be wire-ORed with other sources of interrupts. An external pull-up resistor of 1.8 kΩ or greater is required.
46	IEI	I	(Interrupt Enable In). IEI is an active-high input which allows the DTC to activate the $\overline{INT}$ output and to respond to interrupt acknowledge operations. It is used with other signals to implement the interrupt daisy chain. Transitions on IEI do not have to be synchronous with the DTC clock.
48	IEO	O	(Interrupt Enable Out). IEO is an active-high output that enables devices lower in the chain when higher priority interrupts are not pending or under service. It is used in conjunction with other signals to implement the Interrupt daisy chain. See the Interrupt section of this document for further details on $\overline{INT}$, IEI and IEO.
36, 37	$\overline{DREQ_1}$, $\overline{DREQ_2}$	I	(DMA Request). The DMA Request lines are two active-low inputs, one per channel. They may make transitions independent of the DTC clock and are used by external logic to indicate and control DMA operations performed by the DTC.
40, 39	$\overline{DACK_1}$, $\overline{DACK_2}$	O	(DMA Acknowledge). The DMA Acknowledge lines are active-low outputs, one per channel, which indicate that the channel is performing a DMA operation. $\overline{DACK}$ is pulsed, held active or held inactive during DMA transfers, as programmed in the Channel Mode register. For Flowthru operations, the peripheral is fully addressed using the conventional I/O addressing protocols and therefore may choose to ignore $\overline{DACK}$. $\overline{DACK}$ is always output as programmed in the Channel Mode register for a DMA operation, even when the operation is initiated by a CPU software request command or as a result of chaining. $\overline{DACK}$ is not output during the actual chaining operations.
38	$\overline{EOP}$	I/O	(End of Process). $\overline{EOP}$ is an active-low, open-drain, bidirectional signal. It must be pulled up with an external resistor of 1.8 kΩ or more. The DTC emits an output pulse on $\overline{EOP}$ when a TC or MC termination occurs, as defined later. An external source may terminate a DMA operation in progress by driving $\overline{EOP}$ LOW. $\overline{EOP}$ always applies to the active channel; if no channel is active, $\overline{EOP}$ is ignored. The Suppress output of the MMU may be connected to $\overline{EOP}$ to terminate DMA accesses which violate the MMU protection settings. To provide full access protection, an external $\overline{EOP}$ is accepted even during chaining.
29, 28, 25, 24-21	SN_0-SN_6	O	(Segment Number). The segment lines are three-state outputs activated only when the DTC is controlling the system bus. SN_0 is the least significant bit of the segment number and SN_6 is the most significant. The Z8001 and Z8002 CPUs access I/O by outputting a 16-bit I/O address on AD_0-AD_{15}. When the AmZ8016 DTC is operated in Logical Address space, the I/O address space is increased to 23 bits. The lower 16 bits of I/O address appear on AD_0-AD_{15}. An additional 7 bits of I/O addresses appear on SN_0-SN_6. Users of the DTC in the Logical Address space configuration may choose to disregard the SN_0-SN_6 I/O address information or may use it to increase the DTC's I/O address space beyond that of the CPU. When the AmZ8016 DTC is configured for Physical Address space, signals SN_0-SN_6 specify the 17th (SN_0) through 23rd (SN_6) bits of a 24-bit linear address. The lower 16 address bits appear on AD_0 through AD_{15} respectively; the 24th address bit is output on SN_7/MMUSync. This 24-bit linear address allows the DTC to access anywhere within 16 Megabytes of memory. Users of the DTC in the physical address space configuration may choose to disregard the extended I/O addressing capability of the DTC by disregarding SN_0-SN_6 and SN_7/MMUSync during I/O operations, or may use the extended addressing to increase the number of I/O ports accessible by the DTC beyond the number of I/O ports accessible by the CPU.

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
27	SN_7/MMUSync	O	(Segment Number 7/MMUSync). When DTC is programmed in Logical Address space, this line outputs an active-HIGH MMUSYNC pulse prior to each machine cycle. The MMU uses this signal to synchronize access to its translation table and to differentiate between CPU and DTC control. The MMU ignores MMUSYNC if ST_0-ST_3 indicate I/O. This output is LOW when DTC is a bus slave and the MM_1 bit is set. In Physical address space, this line outputs SN_7 which becomes the 24th address bit in a linear address space. This bit can be used to address both memory and I/O – see the SN_0-SN_6 pin description for details. With this output SN_7, a HIGH represents 1 and a LOW represents 0. This pin floats to high impedance state when DTC is a bus slave and the MM_1 bit is cleared.

ST3	ST2	ST1	ST0	Transaction/Operation	DTC Action (Note)
L	L	L	L	Internal Operation	
L	L	L	H	Memory Refresh	
L	L	H	L	I/O Transaction	G
L	L	H	H	Special I/O Transaction	G
L	H	L	L	Segment Trap Acknowledge	D
L	H	L	H	Non-Maskable Interrupt Acknowledge	D
L	H	H	L	Non-Vectored Interrupt Acknowledge	D
L	H	H	H	Vectored Interrupt Acknowledge	D
H	L	L	L	Memory Transaction for Data/DTC Chaining	G
H	L	L	H	Memory Transaction for Stack	G
H	L	H	L	Reserved	
H	L	H	H	Reserved	
H	H	L	L	Memory Transaction for Program Fetch (Subsequent Word)	G
H	H	L	H	Memory Transaction for Program Fetch (First Word)	
H	H	H	L	Reserved	
H	H	H	H	Reserved	

Notes: D = Status code is decoded by DTC when not in control of system bus.
G = Status code is generated by DTC when in control of system bus.

Figure 1. Status Code

Refer to page 7-1 for Essential Information on Military Devices

DETAILED DESCRIPTION

Any given DMA operation, be it a transfer, a search or a transfer-and-search operation, consists of three phases. In the first phase, the channel's registers are initialized to specify and control the desired DMA operation. In the second phase, the DMA operation itself is started and performed. The final phase involves terminating the DMA operation and performing any actions selected to occur on termination. Each of these different phases is described in detail in the following sections.

Reset

The DTC can be reset either by hardware or software. The software reset command is described in the "Commands" section. Hardware resets are applied by pulling both $\overline{AS}$ and $\overline{DS}$ LOW. Because the DTC may be in control of the bus when a reset is applied, it is important that $\overline{BAI}$ be driven HIGH when applying a reset to avoid possible bus contention between the applied LOW signals on $\overline{AS}$ and $\overline{DS}$ and the DTC's driving of these pins. As soon as $\overline{BAI}$ goes inactive, the DTC places the $AD_0 - AD_{15}$, $SN_0 - SN_6$, $ST_0 - ST_3$, $R/\overline{W}$, $N/\overline{S}$, $B/\overline{W}$, $\overline{AS}$ and $\overline{DS}$ signals in the high impedance state. If the DTC is programmed for Physical Address Space, SN_7/MMUSync will also be driven into the high impedance state when $\overline{BAI}$ goes HIGH. Figure 21 shows the suggested method of generating hardware resets for the DTC.

Both software and hardware resets clear the Master Mode register, clear CIE, IP, SIP, and WFB and set the CA and NAC in each Channel's Status register. The contents of all other DTC registers will be unchanged for a software reset. Since a hardware reset may have been applied part-way through a DMA operation being performed by a DTC channel, the channel's registers should be assumed to contain indeterminate data following a hardware reset.

Because the CA and NAC bits in the Status register are set by reset, the channel will be prevented from starting a DMA operation until its Chain Address register's Segment, Tag and offset fields are programmed and the channel is issued a "Start Chain Command".

Channel Initialization

The philosophy behind the AmZ8016 DTC design is that the DTC should be able to operate with a minimum of interaction with the host CPU. This goal is achieved by having the DTC load its own control parameters from memory into each channel. The CPU has to program only the Master Mode register and each Channel's Chain Address register. All other registers are loaded by the channels themselves from a table located in System Data memory and pointed to by the Chain Address register. This reloading operation is called chaining and the table is called the Chain Control Table.

The offset and segment fields of the Chain Address Register form a 24-bit address or a 23-bit address, which points to a location in system data memory space. Chaining is performed by repetitively reading words from memory. Note that the Chain Address register should always be loaded with an even offset; loading an odd offset will cause unpredictable results. The 2-bit Tag field facilitates interfacing to slow memory by allowing the user to select 0, 1, 2 or 4 programmable wait states. The DTC will automatically insert the programmed number of wait states in each memory access during chaining.

The Chain Address register points to the first word in the Chain Control Table. This word is called the Reload Word. See Figure 2. The purpose of the Reload Word is to specify which registers in the channel are to be reloaded. Reload Word bits 10 – 15 are undefined and may be 0 or 1. Each of bits 0 through 9 in the Reload Word correspond to either one or two registers in the channel (see Figure 3). When a Reload Word bit is '1', it means that the register or registers corresponding to that bit are to be reloaded. If a Reload Word bit is '0', the register or registers corresponding to that bit are not to be reloaded. The data to be loaded into the selected register(s) follow(s) the Reload Word in memory (i.e., the data are at successively larger memory addresses). The Chain Control Table is a variable length table. Only the data to be loaded are in the table and the data are packed together.

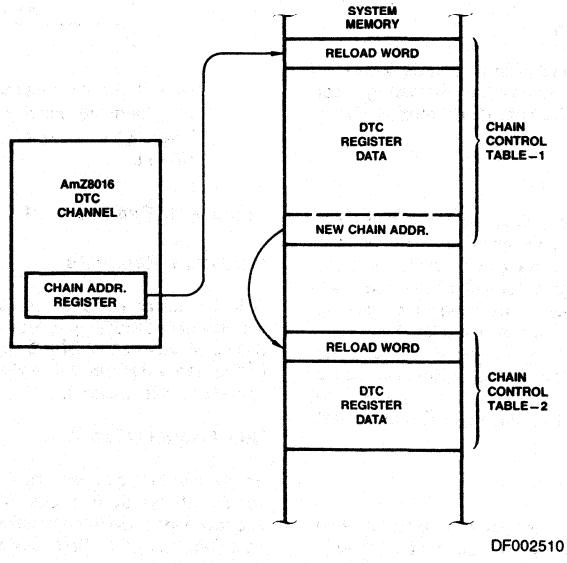


Figure 2. Chaining and Chain Control Tables

Refer to page 7-1 for Essential Information on Military Devices

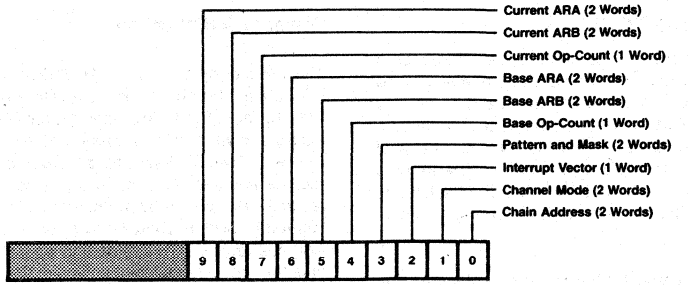


Figure 3. Reload Word

When the channel is to reload itself, it first uses the Chain Address register contents to load the Reload Word into the DTC's Chain Control Register. Next, the Chain Address register contents are incremented by two to point to the next word in memory. The channel then scans to Reload Word register from bit 9 down to bit 0 to see which registers are to be reloaded. If no registers are specified (bits 9 – 0 are all 0), no registers will be reloaded. If at least one of bits 9 – 0 are set to '1', the register(s) corresponding to the most significant set bit are reloaded, the bit is cleared and the Chain Address register is incremented by 2. The channel continues this operation of scanning the bits from the most significant to least significant bit position clearing each set bit after reloading its associated registers and incrementing the Chain Address register by 2. If all of bits 9 to 0 are set, all the registers will be reloaded in the order: Current ARA, Current ARB, Current Operation Count, . . . Channel Mode and Chain Address. Figure 4 shows two examples of Chain Control Tables. Example 1 shows the ordering of data when all register are to be reloaded. In example 2 only some registers are reloaded. Once the channel is reloaded, it is ready to perform a DMA operation. Note when loading Address Registers the Segment and Tag Word are loaded first, then the Offset Word.

Initiating DMA Operations

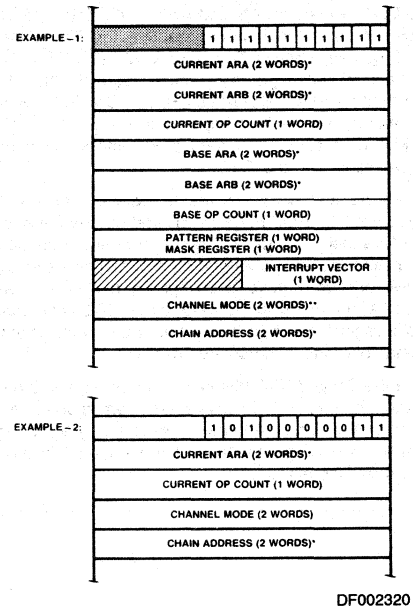
DMA Operations can be initiated in one of three ways – by software request, by hardware request and by loading a set software request bit into the Channel Mode register during Chaining.

Starting After Chaining

If the software request bit of the Channel Mode register is loaded with a '1' during chaining, the channel will perform the programmed DMA operation at the end of chaining. If the channel is programmed for Single Operation or Demand, it will perform the operation immediately. The channel will give up the bus after chaining and before the operation if the CPU Interleave bit in the Master Mode register is set. See the "Channel Response" section for details. Note that once a channel starts a chaining operation by fetching a Reload Word, it retains bus control at least until chaining of the last register's data is performed.

Software Requests

The CPU can issue Software Request commands to start DMA Operations on a channel. This will cause the channel to request the bus and perform transfers. See the description of the software request command for details.



*Note: Load the segment and tag word first, then the offset word.

**Most significant word first, then least significant.

Figure 4. Examples of Chain Control Table

Hardware Requests

DMA operations will often be started by applying a LOW on the channel's $\overline{\text{DREQ}}$ input. The "Channel Response" section describes when LOW $\overline{\text{DREQ}}$ signals are sampled and when $\overline{\text{DREQ}}$ requests can be applied to start the next DMA operation after chaining.

Bus Request/Grant

Before the DTC can perform a DMA Operation, it must gain control of the system bus. The $\overline{\text{BUSRQ}}$, $\overline{\text{BAI}}$ and $\overline{\text{BAO}}$ interface pins provide connections between the DTC and the host CPU and other DMA devices, to arbitrate which device has control of the system bus. When the DTC wants to gain bus control, it drives $\overline{\text{BUSRQ}}$ LOW.

Refer to page 7-1 for Essential Information on Military Devices

Some period of time after the DTC drives $\overline{BUSRQ}$ LOW, the CPU will relinquish bus control and drive its $\overline{BUSAK}$ signal LOW. This passes down the $\overline{BAI}$, $\overline{BAO}$ daisy chain. When the DTC's $\overline{BAI}$ input goes LOW, it may begin performing operations on the system bus. When the DTC finishes its operation, it stops driving $\overline{BUSRQ}$ LOW and allows $\overline{BAO}$ to follow $\overline{BAI}$.

Listed below are the rules followed by the DTC to request, acquire, and release the system bus. A description of the significance of the steps follows.

1. The DTC requests control of the system bus by driving $\overline{BUSRQ}$ LOW. The DTC may only drive $\overline{BUSRQ}$ LOW if $\overline{BUSRQ}$ is HIGH and $\overline{BAI}$ is HIGH.
2. After driving $\overline{BUSRQ}$ LOW, the DTC waits for its request to be acknowledged on the $\overline{BAI}$ input. When $\overline{BAI}$ goes LOW, the DTC has bus control, performs its operations and continues to drive $\overline{BUSRQ}$ until it completes.
3. When the DTC is finished with the system bus, it stops driving $\overline{BUSRQ}$ LOW and passes the LOW on $\overline{BAI}$ through to $\overline{BAO}$.
4. If the DTC is not requesting use of the bus, $\overline{BAO}$ always follows $\overline{BAI}$. If the DTC receives a LOW on $\overline{BAI}$ and $\overline{BUSRQ}$ is LOW and the DTC is not requesting use of the bus, the DTC drives $\overline{BAO}$ LOW. This situation would occur if some lower priority device was pulling $\overline{BUSRQ}$ LOW. The DTC simply passes the LOW $\overline{BAI}$ grant signal through to the lower priority device.

Note that $\overline{BAO}$ will always be LOW if $\overline{BAI}$ is LOW providing the DTC is not driving $\overline{BUSRQ}$ LOW. If the DTC is driving $\overline{BUSRQ}$ LOW, $\overline{BAO}$ will go LOW when the DTC finishes using the bus and stops applying a LOW to $\overline{BUSRQ}$. Note also that $\overline{BUSRQ}$ is a bidirectional signal. Since the DTC can only drive $\overline{BUSRQ}$ LOW if $\overline{BUSRQ}$ was previously HIGH, the DTC is able to sample $\overline{BUSRQ}$. Because the DTC may be on a different card than other DTCs and the CPU, some means must be provided to bidirectionally buffer $\overline{BUSRQ}$. Figure 21 shows a representative system with two DTC chips and a CPU. Figure 22 shows the logic used to bidirectionally buffer $\overline{BUSRQ}$. Note that the buffer and gates in the logic are both open collector (o.c.) devices.

It is necessary to ensure that all DTCs will behave identically, regardless of whether they are on the same card or different cards. Also, it is undesirable to require users to provide to Detail A logic on all DTCs, except where the logic is needed to provide buffering to drive a backplane. For this reason, each DTC incorporates identical logic to Detail A inside the chip. Thus, even if no external logic is used, the bus request-grant protocol will follow the above description. Note that when external buffering is used, the design of Detail A is such that when it is placed in series with the replicated Detail A logic inside the chip, the operation of the bus request protocol remains unchanged.

DMA Operations

There are three types of DMA operations: Transfer, Search and Transfer-and-Search. Transfers move data from a source location to a destination location. Two types of transfers are provided: Flowthru and Flyby. Searches read data from a source and compare the read data to the contents of the Pattern register. A Mask register allows the user to declare "don't care" bits.

The user can program that the search is to stop either when the read data matches the masked pattern or when the read data fails to match the masked pattern. This capability is called Stop-on-Match and Stop-on-no-Match. Transfer-and-Search combines the two functions to facilitate the transferring of variable length data blocks. Like transfer, Transfer-and-Search can be performed in either Flowthru or Flyby mode.

Transfers

Transfers use four of the Channel registers to control the transfer operation: the Current ARA and ARB register; the Current Operation Count register; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select whether ARA is to point to the source and ARB is to point to the destination or vice-versa. The Current Operation Count register specifies the number of words or bytes to be transferred.

Bits $CM_3 - CM_0$ in the Channel Mode register program whether Flowthru or Flyby transfer is to be performed. Flowthru transfers are performed in either two or three steps. First, the channel outputs the address of the source and reads the source data into the DTC's Temporary register. In two-step Flowthru Transfer, the channel will then address the destination and write the Temporary register data to the destination location. The three-step Flowthru operation is described later in this section. The source and destination for Flowthru Transfers can both be memory locations or both peripheral devices or one may be a memory location and the other a peripheral device. The $\overline{DACK}$ output for the transferring channel may be programmed to be inactive throughout the transfer or active during the transfer. This is controlled by bit CM_{18} in the Channel Mode register.

Flyby transfers provide improved transfer throughput over Flowthru but are restricted to transfers between memory and peripherals or between two peripherals. Flyby operations are described in detail in the "Flyby Transactions" section.

Transfers can use both byte- and word-sized data. Flowthru byte-to-byte transfers are performed by reading a byte from the source and writing a byte to the destination. The Current Operation Count register must be loaded with the number of bytes to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±1 if the register points to memory space and by ±2 if the register points to I/O space.

Flowthru word-to-word transfers require that the Current Operation Count specify the number of words to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±2 regardless of whether the register points to memory or I/O space.

Byte-word funneling provides packing and unpacking of byte data to facilitate high speed transfers between byte and word peripherals and/or memory. This funneling option can only be used in Flowthru mode. Funneled Flowthru transfers are performed in three steps. For transfers from a byte source to a word destination, two consecutive byte reads are performed from the source address. The data read is assembled into the DTC's Temporary register. In the third step, the Temporary register data is written to the destination address in a word transfer. Funneled transfers from a word source to a byte destination are performed by first loading a word from the source into the DTC's Temporary register. The word is then written out to the destination in two byte writes. For funnel operations, the byte-oriented address must be in the Current ARA register and the word-oriented address must be in the Current ARB register. The Flip bit (CM_4) in the Channel Mode register is used to specify which address is the source and which is the destination. When the byte address is to be incremented or decremented, the increment/decrement operation occurs after each of the two reads or writes. The increment/decrement is by ±1.

Refer to page 7-1 for Essential Information on Military Devices

In byte-to-word funneling operations it is necessary to specify which half of the Temporary register (upper or lower byte) is loaded with the first byte of data. Similarly, for word-to-byte funneling operations it is necessary to define which half of the Temporary register is written out first. Figure 5 summarizes these characteristics for both byte-to-word and word-to-byte funneling operations. The criteria used to determine the packing/unpacking order is based on whether the Current ARB register is programmed for incrementing or decrementing of the address. Note that if the address is to remain unchanged (i.e., if bit TG_4 in the Tag Field of the Current ARB register is 1), the increment/decrement bit (bit TG_3) still specifies the packing order.

Search

Searches use five of the Channel registers to control the transfer operation: either the Current ARA or ARB; the Operation Count; the Pattern and Mask registers; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select either Current ARA or ARB as the register specifying the source for the search. Only one of the Current Address registers is used for search operations since there is no destination address required. Channel mode register bit CM_2 is an enable for the output of the comparator and allows the MC (match condition) signal to be generated. The Current Operation Count register specifies the maximum number of words or bytes to be searched.

Search operations involve repetitive reads from the peripheral or memory until the specified match condition is met. The search then stops. This is called a Match Condition or MC termination. Each time a read is performed, the Source address, if so programmed, is incremented or decremented and the Operation Count is decremented by 1. If the match condition has not been met by the time the Operation Count reaches zero, the zero value will force a TC termination, ending the search. Searches can also stop due to a LOW being applied to the $\overline{EOP}$ interface pin. During a search operation, the channel's $\overline{DACK}$ output will be either inactive or active throughout the search. This is controlled by bit CM_{18} in the Channel Mode register. The reads from the peripheral or memory performed during search follow the timing sequences described in the "Flowthru Memory Transactions" and "Flowthru I/O Transactions" sections.

On each read during a Search operation, the DTC's Temporary register is loaded with data and compared to the Pattern register. The user can select that the search is to stop when the Pattern and Temporary register contents match or when they don't match. This Stop-On-Match/Stop-On-No-Match feature is programmed in bit CM_{17} of the Channel Mode register. A Mask register allows the user to exclude or mask selected Temporary register bits from the comparison by setting the corresponding Mask register bit to "1." The masked bits are defined to always match. Thus, in Stop-On-Match, successful matching of the unmasked bits, in conjunction with the always-matched masked bits, will cause the search to stop. For Stop-On-No-Match, the always-matched masked bits are by definition excluded from not matching and therefore excluded from stopping the search.

For word reads the user may select either 8-bit or 16-bit compares through Channel Mode register bit CM_{16}. In an 8-bit, Stop-On-Match, word-read operation, successful matching of either the upper or lower byte of unmasked Pattern and Temporary registers bits will stop the search. Both bytes do not have to match. In 16-bit Stop-On-Match with word reads, all unmasked Pattern and Temporary register bits must match to stop the search. In an 8-bit or 16-bit, Stop-On-No-Match, word-read Search operation, failure of any bit to match will terminate the Search operation.

In an 8-bit Stop-On-Match the byte-reads, the Search will Stop if either the upper or lower byte of unmasked Pattern and Temporary register bits match. For an 8-bit Stop-On-No-Match with byte reads, failure of matching in any unmasked Pattern and Temporary register bit will cause the search to stop.

For 8-bit searches, the upper and lower bytes of the Pattern and Mask register should usually be programmed with the same data. Failure to set the upper and lower bytes of the Pattern and Mask registers to identical values will result in different comparison criteria being used for the upper an lower bytes of the Temporary register. Users failing to program identical values for the upper and lower bytes can predict the results by recognizing that in 8-bit Stop-On-Match, the search will end if all the unmasked bits in either the upper or lower byte matches, and for 8-bit Stop-On-No-Match, the failure of any unmasked bit to match will end the search. For accurate predictions, it is also necessary to know that for word reads the Temporary register high and low bytes are loaded from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. In byte reads, the read byte is duplicated in both halves of the Temporary register except in funneling.

Funneling Direction	Current ARB Tag Field		Increment/Decrement and Packing/Unpacking Rules
	TG4	TG3	
Word-to-Byte (Flip-bit = 1)	0	0	Increment ARB, Write High Byte First
	0	1	Decrement ARB, Write Low Byte First
	1	0	Hold ARB, Write High Byte First
	1	1	Hold ARB, Write Low Byte First
Byte-to-Word (Flip-bit = 0)	0	0	Increment ARB, Read High Half of Word Written First
	0	1	Decrement ARB, Read Low Half of Word Written First
	1	0	Hold ARB, Read High Half of Word Written First
	1	1	Hold ARB, Read Low Half of Word Written First

Figure 5. Byte/Word Funneling

Refer to page 7-1 for Essential Information on Military Devices

Transfer-and-Search

Transfer-and-Search combines the operations of Transfer and Search functions. The registers used to control Transfer-and-Searches are the Current ARA and ARB register, the Operation Count register, the Pattern and Mask register, and the Channel Mode register.

A Transfer-and-Search operation will end when the data transferred meets the match condition specified in Channel Mode register bits $CM_{17} - CM_{16}$. The Mask and Pattern registers indicate those bits being compared with the Temporary register contents. Like Transfers and Searches, Transfers-and-Searches will also be terminated if the operation count goes to zero or if a LOW is applied to the $\overline{EOP}$ pin. Regardless of whether Transfer-and-Search stops because of a TC, MC or EOP, it will always complete the iteration by writing to the destination address before ending (writing twice for word-to-byte funneling).

In Flowthru mode, Transfer-and-Search the timing is identical to Flowthru Transfer. While the data is in the Temporary register, it is masked by the Mask register and compared to the Pattern register. For word Transfer and Transfer-and-Search, the high and low bytes of the Temporary register are always written to and read from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. For byte Transfer and Transfer-and-Search, the byte read is always loaded into both halves of the Temporary register and the entire register is driven directly out onto the $AD_{15} - AD_0$ bus. Transfer-and-Search can also be used with byte word funneling. In funneling, the match is an 8-bit match or 16-bit match as determined by the setting of bit CM_{16} and CM_{17}.

Flyby Transfer-and-Search can be used to increase throughput for transfer between two peripherals or between memory and a peripheral. In this operation, the operand sizes of the source and destination must be the same. A complete discussion of Flyby timing is given in the "Flyby Transactions" section. During a Flyby Transfer-and-Search, data is loaded into the Temporary register to facilitate the comparison operation and at the same time data is transferred from the source to the destination. When byte operands are used, data is loaded into both bytes of the Temporary register, from the $AD_{15} - AD_8$ bus if the Current ARA register is even and from $AD_7 - AD_0$ line if the Current ARA register is odd. This will alternate for memory bytes so the user must drive both halves of the bus to use the search. When word operands are used, data is loaded directly from $AD_{15} - AD_8$ and $AD_7 - AD_0$ into the Temporary register's high and low bytes respectively.

Channel Response

Channel Mode register bits $CM_6 - CM_5$ select the channel's response to the request to start a DMA operation. The response falls into either of two types: Single Operation or Demand. There are three subtypes for Demand operations: Demand Dedicated with Bus Hold, Demand Dedicated with Bus Release, and Demand Interleave. To make the discussions clear, it is necessary to define the term "single iteration of a DMA operation". For Search operations, one iteration consists of a single read operation and a comparison of the read data to the unmasked Pattern register bits. The Operation Count will be decremented by 1 and the Current Address register used incremented or decremented if so programmed. For Transfer and Transfer-and-Search operations, a single iteration comprises reading a datum from the source, writing it to the destination, comparing the read datum to the unmasked Pattern register bits (Transfer-and-Search only), decrementing the Operation Count by 1 and incrementing/decrementing the Current ARA and ARB registers if so programmed. In byte-word funneling, a single iteration consists of two reads followed by a write (Byte-to-Word funneling) or one read followed by two writes (Word-to-Byte funneling). In all Transfer and Transfer-and-Search cases the iteration will not stop until the data in the Temporary register is written to the destination.

Single Operation

The Single Operation response is intended for use with peripherals which transfer single bytes or words at irregular intervals. Each application of a Software request command will cause the channel to perform a single iteration of the DMA operation. Similarly, if the Software request bit is set by chaining, at the end of chaining the channel will perform a single iteration of the DMA operation. Each application of a HIGH-to-LOW transition on the $\overline{DREQ}$ input will also cause a single iteration of the DMA operation. If the Hardware mask bit is set when the transition is made, the iteration will be performed when the mask is cleared, providing the DMA operation has not terminated. See the Set/Clear Hardware mask bit command for details. Each time a Single Operation ends, the channel will give up control of the bus unless a new transition has occurred on $\overline{DREQ}$. The new transition can occur anytime after the LOW-to-HIGH $\overline{AS}$ transition on the first memory or I/O access of the DMA iteration. Timing Diagram 1 shows the times after which a new transition can be applied and recognized to avoid giving up the bus at the end of the current iteration.

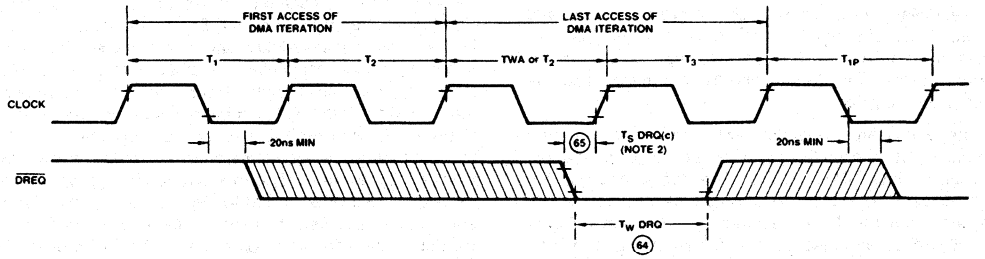
Demand Dedicated with Bus Hold

In Demand Dedicated with Bus Hold (abbreviated Bus Hold), the application of a Software request command or the setting of the software request bit during chaining or applying a LOW level on the $\overline{DREQ}$ input will cause the channel to acquire bus control.

If $\overline{DACK}$ is programmed as a level output ($CM_{18} = 0$), $\overline{DACK}$ will be active from when the channel acquires bus control to when it relinquishes control.

Once the channel gains bus control due to a LOW $\overline{DREQ}$ level, it samples $\overline{DREQ}$ as shown in Timing Diagram 2. If $\overline{DREQ}$ is LOW, an iteration of the DMA operation is performed. If $\overline{DREQ}$ is HIGH, the channel retains bus control and continues to drive all bus control signals active or inactive, but performs no DMA operation. This the user can start or stop execution of DMA operations by modulating $\overline{DREQ}$. Once TC, MC or EOP occurs, the channel will either release the bus or, if chaining or Base-to-Current reloading is to occur, perform the desired operation. After chaining or Base-to-Current reloading, if the channel is still in Bus Hold mode and does not have a set software request bit (set either by chaining or command), the channel will relinquish bus control unless a LOW $\overline{DREQ}$ level occurs within the time limits.

Refer to page 7-1 for Essential Information on Military Devices



WF005310

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of the first access of the DMA iteration.
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur T_SDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before T_SDRQ(c) if it has met the T_WDRQ parameter.
3. Flyby and Search transactions have only a single access; parameter T_SDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

Timing Diagram 1. Sampling $\overline{DREQ}$ During Single Transfer DMA Operations

Demand Dedicated with Bus Release

In Demand Dedicated with Bus Release (abbreviated Bus Release), the application of a Software Request command will cause the channel to request the bus and perform the programmed DMA operation until TC, MC or EOP. If the channel was programmed for Bus Release, and the software request bit was set during chaining, the channel will start the DMA operation as soon as chaining ends, without releasing the bus, and will continue performing the operation until TC, MC or EOP.

When an active $\overline{DREQ}$ is applied to a channel programmed for Bus Release, the channel will acquire the bus and perform DMA operations until (a) TC, MC or EOP or (b) until $\overline{DREQ}$ goes inactive. Timing Diagram 2 (b) shows when $\overline{DREQ}$ is sampled to determine if the channel should perform another cycle or release the bus. Note that this sampling also occurs on the last cycle of a chaining operation. If a channel has an active $\overline{DREQ}$ at the end of chaining, it will begin performing DMA operations immediately, without releasing the bus. When a TC, MC or EOP occurs, terminating a Bus Release mode operation, the channel, if enabled for chaining and/or Base-to-Current reloading, will perform chaining and/or reloading (assuming the Status register's SIP bit is clear) without releasing the bus.

If an active request is not applied and the channel is in Demand Dedicated with Bus Hold, the channel will go into state THLD (see Timing Diagram 2 (a)). If an active request is not applied and the channel is in Demand Dedicated with Bus Release or Demand Interleave mode, it will release the bus. Note that even if an active request is applied in Demand Interleave, the channel may still release the bus. The request for Demand Interleave should continue to be applied to ensure that the channel eventually responds to the request by acquiring the bus (i.e., the request is not latched by the channel).

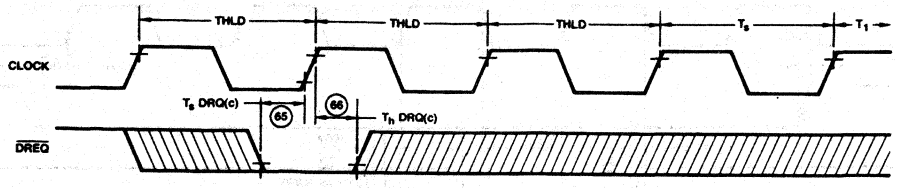
Demand Interleave

Demand Interleave behaves in different ways depending on the setting of Master Mode register bit MM_2. If MM_2 is set, the DTC will always relinquish bus control and then re-request it after each DMA iteration. This permits the CPU and other devices to gain bus control. For instance, if MM_2 is clear, control can pass from one DTC channel to the other without requiring the DTC to release bus control. If both channels have active requests, control will pass to the channel which did not just have control. If MM2 is clear and both channels have active requests and are in Demand Interleave mode, control will toggle between the channels after each DMA operation iteration and the DTC will retain bus control until both channels are finished with the bus. If MM2 is set and both channels have active requests and are in Demand Interleave mode, each channel will relinquish control to the CPU after each iteration resulting in the following control sequence: channel 1, CPU, channel 2, CPU, etc. Note that if there are other devices on the bus request daisy chain, they may gain control during the part of the sequence labeled CPU.

A software or hardware request will cause a channel programmed for Demand Interleave to perform interleaved DMA operations until TC, MC or EOP. If the Software request bit is set during chaining, the channel will retain the bus after chaining and will immediately start performing DMA iteration and will interleave all DMA iterations after the first. If $\overline{DREQ}$ is LOW on the last cycle during chaining, the channel will perform a single iteration immediately after chaining and interleave thereafter until (a) TC, MC or EOP or (b) $\overline{DREQ}$ goes HIGH. If (b) occurs, the channel will relinquish the bus until $\overline{DREQ}$ goes LOW again and the channel again starts performing interleaved operations. If (a) occurs, the channel will not interleave before first performing chaining and/or Base-to-Current reloading (assuming SIP is cleared).

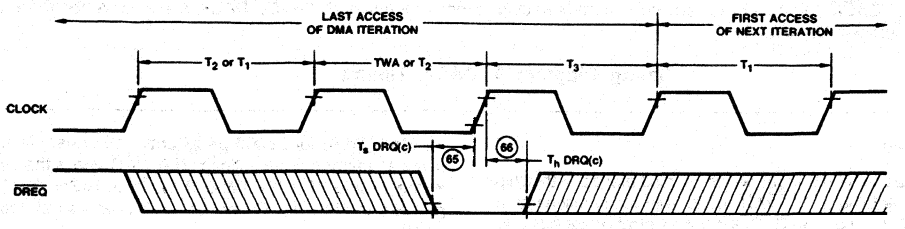
The waveform of $\overline{DACK}$ is programmed in Channel Mode Register (CM_{18}). The Pulsed $\overline{DACK}$ is for flyby transaction only. See Timing Diagram 3. Note: This figure shows a single Search or Flyby iteration. State TWA is optionally inserted if programmed. For more than one iteration, the level $\overline{DACK}$ output would stay active during the time the channel had bus control. When CM_{18} is set, the $\overline{DACK}$ output will be inactive for all non-flyby modes.

Refer to page 7-1 for Essential Information on Military Devices



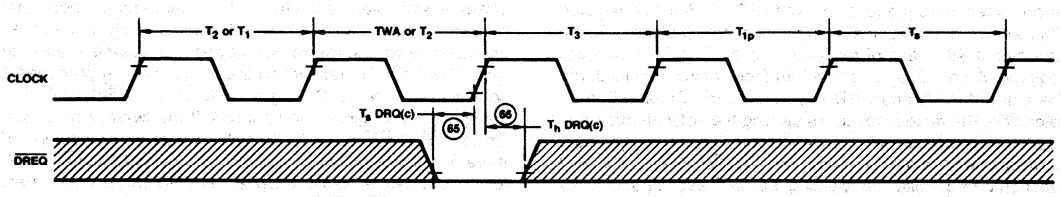
WF005320

a) Sampling $\overline{DREQ}$ while in Bus Hold Mode



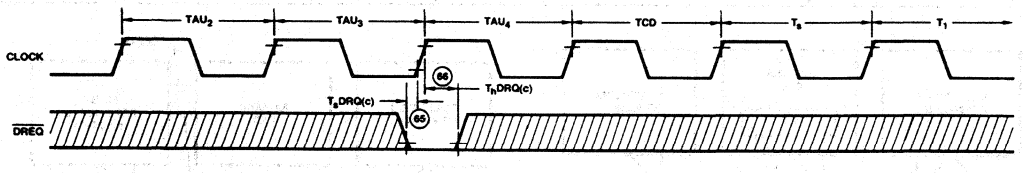
WF005330

b) $\overline{DREQ}$ Sampling in Demand Mode During DMA Operations



WF005340

c) Sampling $\overline{DREQ}$ at the End of Chaining



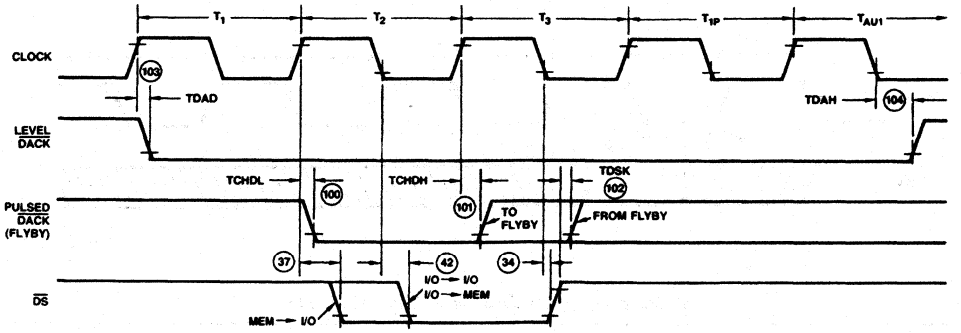
WF005350

d) Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading

Notes: 1. $\overline{DREQ}$ must be LOW from the start of T_SDRQ(c) to the end of T_hDRQ(c) to ensure that the request is recognized.
2. Failure to meet this set-up time will result in the channel releasing the bus.
3. T_S is a set-up state, generated before entering DMA operation cycle.
4. TAU_2 through TAU_4 are auto-reloading states, followed by TCD (chain decision) state.

Timing Diagram 2. $\overline{DREQ}$ Sampling in Demand Mode

Refer to page 7-1 for Essential Information on Military Devices



WF005360

Note: LEVEL $\overline{\text{DACK}}$ RE occurs as shown if auto-reloading is not programmed. LEVEL $\overline{\text{DACK}}$ stays LOW for three additional clocks for reloading.

Timing Diagram 3. $\overline{\text{DACK}}$ Timing

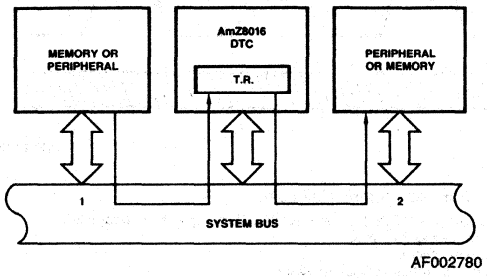
Wait States

The DTC has a $\overline{\text{WAIT}}$ input which is multiplexed with Chip Select ($\overline{\text{CS}}$) yielding a $\overline{\text{CS}}/\overline{\text{WAIT}}$ input. This pin functions as a $\overline{\text{CS}}$ for the DTC when the DTC is not in control of the bus and as a $\overline{\text{WAIT}}$ input when the DTC is bus master. Because multiplexing $\overline{\text{CS}}$ and $\overline{\text{WAIT}}$ requires external logic, (see Figure 23), the user can select that wait states are automatically inserted when the DTC, as bus master, accesses I/O or memory addresses. The number of wait states to be added to the memory or I/O transfer can be programmed by the user as 0, 1, 2 or 4 and can be separately programmed for the Current Address registers A and B and for the Chain Address register. This allows different speed memories and peripheral to be associated with each of these addresses. The Base Address registers A and B also have a Tag Field which is loaded into the Current ARA and ARB registers during Base-to-Current reloading. Because many users utilizing the software programmable wait states will not need the ability to generate hardware wait states through the $\overline{\text{CS}}/\overline{\text{WAIT}}$ pin, the wait function can be disabled, yielding a Chip Select input only, by clearing the Wait Line Enable bit (MM_3) in the Master Mode register.

During memory transactions, the $\overline{\text{WAIT}}$ input is sampled in the middle of the T_2 state. If $\overline{\text{WAIT}}$ is HIGH, and if no programmable wait states are selected, the DTC will proceed to state T_3. Otherwise, at least one wait state will be inserted. The flowthru I/O transaction should be programmed to have one wait state inserted (TWA) for Z8000 peripherals, otherwise timing is the same as memory transactions. The $\overline{\text{WAIT}}$ line is then sampled in the middle of state TWA. If $\overline{\text{WAIT}}$ is HIGH the DTC will proceed to state T_3. Otherwise additional wait states will be inserted.

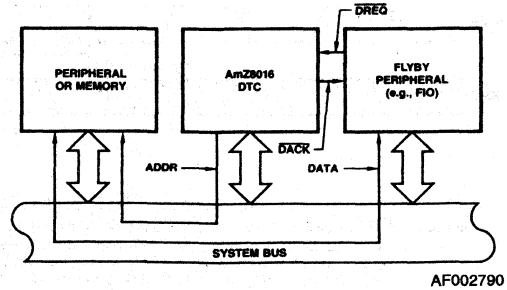
Consider what happens in a transaction when both hardware and software wait states are inserted. Each time the $\overline{\text{CS}}/\overline{\text{WAIT}}$ line is sampled, if it is LOW, a hardware wait state will be inserted in the next cycle. The software wait state insertion will be suspended until $\overline{\text{CS}}/\overline{\text{WAIT}}$ is sampled and is HIGH. The hardware wait states may be inserted anytime during the software wait state sequence. It is important to note that hardware wait states are served consecutively rather than concurrently with software wait states. For example, assume for a Flowthru I/O Transaction that a user has programmed 4 software wait states. Driving a LOW on the $\overline{\text{CS}}/\overline{\text{WAIT}}$ input during T_2 for 2 cycles would insert 2 hardware wait states. Driving $\overline{\text{CS}}/\overline{\text{WAIT}}$ HIGH for 3 cycles would allow insertion of three of the four software wait states. Driving $\overline{\text{CS}}/\overline{\text{WAIT}}$ LOW for 2 more cycles would insert 2 more hardware wait states. Finally, driving $\overline{\text{CS}}/\overline{\text{WAIT}}$ HIGH would allow the final software wait state to be inserted. During this last software wait state, the $\overline{\text{CS}}/\overline{\text{WAIT}}$ pin would be sampled for the last time. If it is HIGH, the channel will proceed to state T_3.

If the pin is LOW, the channel will insert hardware wait states until the pin goes HIGH and the channel would then enter state T_3 to complete the I/O transaction.



AF002780

Figure 6. Configuration of Flowthru Transaction



AF002790

Figure 7. Configuration of Flyby Transaction

Refer to page 7-1 for Essential Information on Military Devices

DMA Transactions

There are three types of transactions performed by the AmZ8016 DTC: Flowthru, Flyby, and Search. Figures 6 and 7 show the configurations of Flowthru and Flyby Transactions.

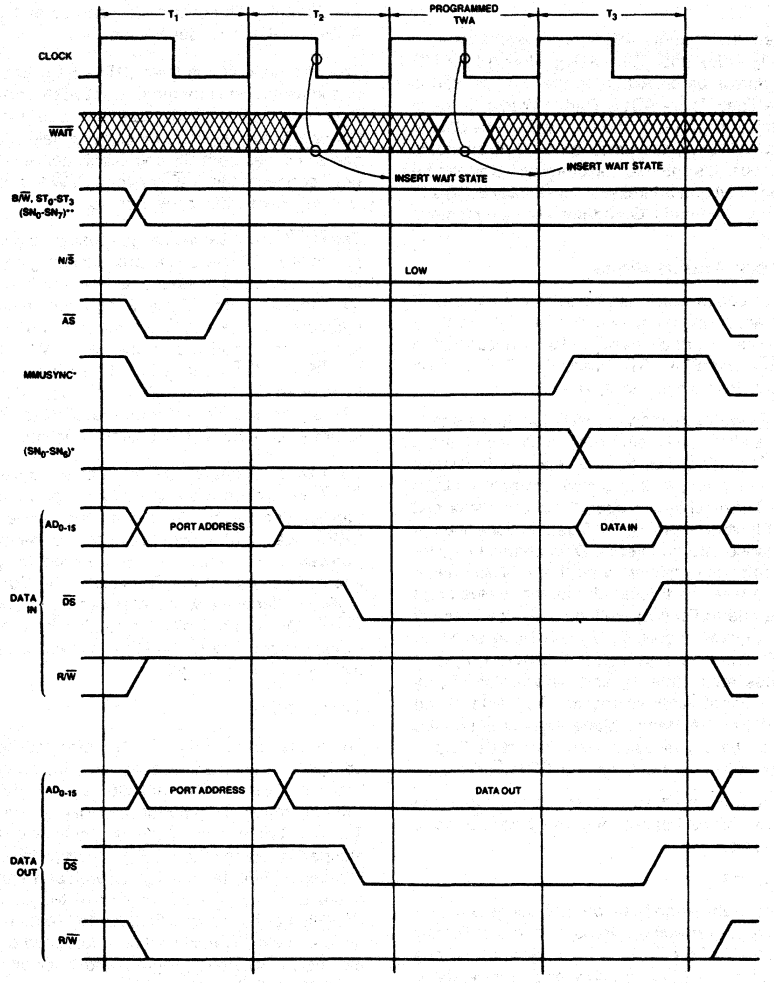
Flowthru I/O Transactions

There are Two types of I/O space on the AmZ8016: I/O and Special I/O. Status lines $ST_0 - ST_3$ specify when an I/O operation is being performed and which of the two I/O spaces are being accessed, as shown in Figure 1. During an I/O transaction, status signal $N/\overline{S}$ will be LOW to indicate a System Level operation.

Each I/O space is addressed by the host CPU by a 16-bit address. The DTC allows an extended I/O address of 24- or 23-bits to be used at the option of the user. When the Master Mode register bit MM_1 is cleared, the DTC is configured for Physical Address space. If MM_1 is set, the DTC is configured for Logical Address space.

The timing for I/O and Special I/O operations are identical. An I/O cycle consists of three states: T_1, T_2, and T_3 as shown in Timing Diagram 4. The TWA state is a wait state programmed to be inserted by the user. The user may select to insert additional software wait states through the Tag fields of the Current ARA and ARB registers. In addition, if Master Mode register bit $MM_3 = 1$, hardware wait states may be inserted by driving a LOW signal on the $\overline{CS}/\overline{WAIT}$ pin.

The $ST_0 - ST_3$ lines will reflect the appropriate code for the current cycle (I/O or Special I/O) early in T_1 and the AS output will be pulsed LOW to mark the beginning of the cycle. The



WF005650

*For logical addressing only
**For physical addressing only

Timing Diagram 4. Flowthru I/O Transactions

Refer to page 7-1 for Essential Information on Military Devices

offset portion of the address for the peripheral being accessed will appear on $AD_0 - AD_{15}$ during T_1. The $N/\overline{S}$ line will be set LOW (system) and the $R/\overline{W}$ line and $B/\overline{W}$ line will select a read or write operation for bytes or words. The $N/\overline{S}$, $R/\overline{W}$ and $B/\overline{W}$ lines will become stable during T_1 and will remain stable until after T_3. I/O address space is byte-address but both 8- and 16-bit data sizes are supported. During I/O transactions the $B/\overline{W}$ output signal will be HIGH for byte transactions and LOW for word transactions. For I/O transactions, both even and odd addresses can be output, hence the address bit output on AD_0 may be 0 or 1.

The channel can perform both I/O read and I/O write operations. During an I/O read, the $R/\overline{W}$ output will be HIGH. The $AD_0 - AD_{15}$ bus will be placed in the high impedance state by the DTC during T_2. The DTC drive the $\overline{DS}$ output LOW to signal the peripherals that data can be gated onto the bus. The DTC will strobe the data into it's Temporary register during T_3. $\overline{DS}$ will be driven HIGH to signal the end of the I/O transaction.

For byte I/O writes, the channel will drive the same data on data bus lines $AD_0 - AD_7$ and $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD0 is 0, the DTC will strobe data in from data lines $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 1, the DTC will strobe data in from data lines $AD_0 - AD_7$. Thus, when an 8-bit peripheral is connected to the bus, it's internal registers will typically be mapped at all even or all odd addresses. To simplify access to 8-bit peripherals, byte oriented I/O address are incremented/decremented by 2.

Flowthru Memory Transactions

There are six status codes which can be generated by a DTC channel while it is accessing memory. See Figure 2. Thus, if a user segregates memory into different banks by decoding the NORMAL/$\overline{SYSTEM}$ and $ST_0 - ST_3$ lines, the DTC can be used to move data from space to space.

The timing for all Flowthru memory transactions is the same, regardless of the status code being output. During chaining operations the DTC reads words from an address in System Data memory pointed to by the active channel's Chain Address register. Those chaining operations are performed identically to the Flowthru memory read transactions, except that the data is loaded into an internal DTC channel register rather than the Temporary register. Note that chaining never causes a write or a byte read; thus, all memory writes or all byte accesses are due to DMA operations. A typical memory operation consists of three cycles: T_1, T_2 and T_3, as shown in Timing Diagram 5. The user may select to insert 1, 2, or 4 software wait states after state T_2 and before state T_3 by programming the Current Address register Tag field. If the Wait Line Enable bit in the Master Mode register is set, the user may also insert hardware wait states after state T_2 and before state T_3 by driving a LOW on the $\overline{CS}/\overline{WAIT}$ signal.

The operation of Flowthru memory transaction are performed identical to the Flowthru I/O transactions except for $\overline{DS}$ width. (See Timing Diagrams.)

Flyby Transactions

Flyby transfers and transfer-and-search operations are performed in a single step, providing a transfer rate significantly faster than that available from Flowthrus. In Flyby, operations can only be performed between memory and peripherals or between peripherals and peripherals. Memory-to-memory operations can not be performed in Flyby mode; these must be done using Flowthru.

Flyby Memory-peripheral operations can only be used with peripherals having a special Flyby signal input. This peripheral input is connected to the channel's $\overline{DACK}$ output. For memory-peripheral Flyby, the address of the source memory location, must be programmed in the Current ARA register. The Current ARB register must be programmed with the destination memory location for peripheral-memory Flyby. Flyby peripheral-to-peripheral operations can only be performed if one of the peripherals has a Flyby signal input. This peripheral input must be connected to the channel's $\overline{DACK}$ output. If both peripherals have a Flyby input, only one should be connected to $\overline{DACK}$; the other peripheral's Flyby input should be held high during the Flyby operation. The address of the peripheral not connected to the channel's $\overline{DACK}$ output should be programmed in the Current ARB register. Note that Flip bit is set (CM4 = 1) for I/O to memory write transactions and cleared (CM4 = 0) for memory read to I/O transaction.

A Flyby operation is performed using three states: T_1, T_2, and T_3. During T_1 the channel pulses AS and outputs the address information. See Timing Diagram 6. The $R/\overline{W}$ line is HIGH if the Current ARA specifies the source and Low if the Current ARB specifies the destination.

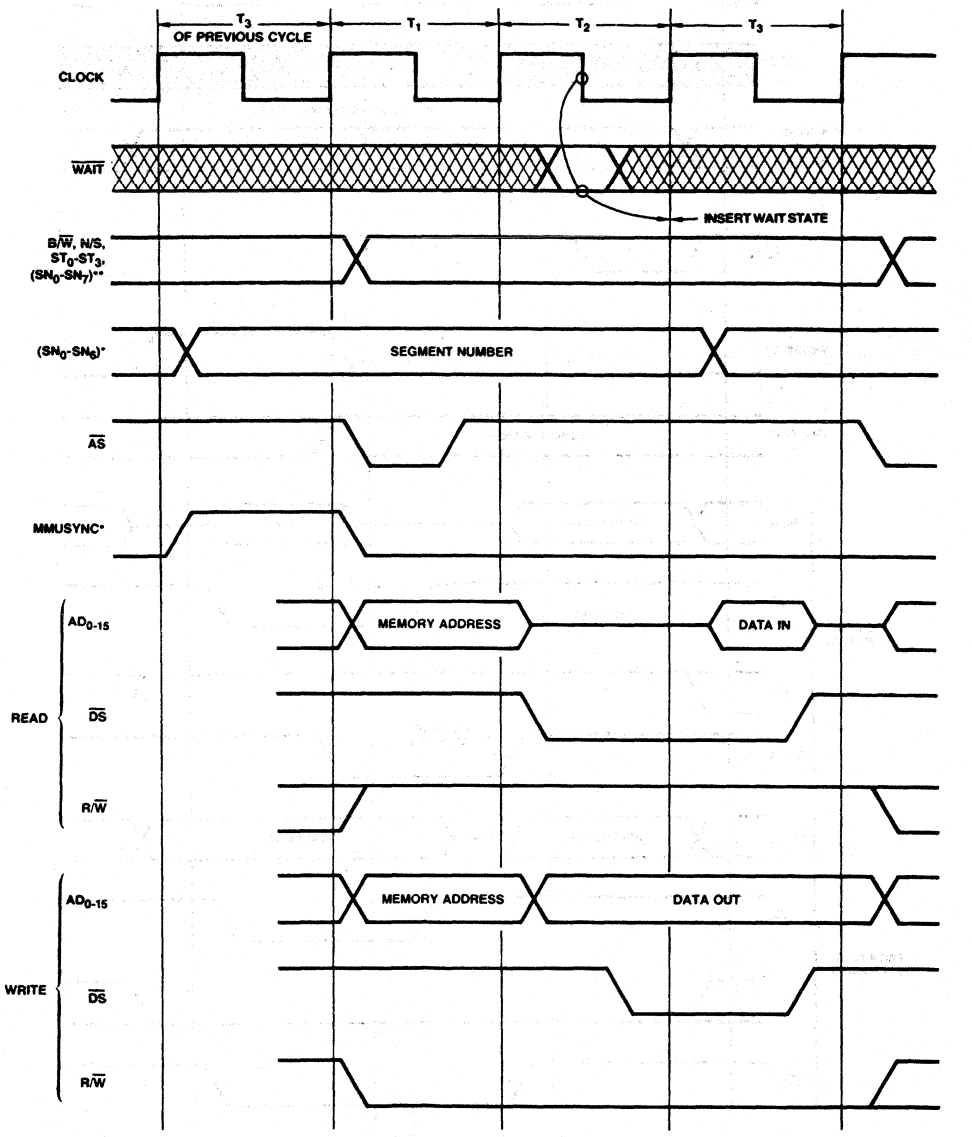
The channel's status lines ($ST_0 - ST_3$) and the N/S line are coded as specified by the Current ARA or ARB Tag field. The $B/\overline{W}$ line indicates the operand size programmed in the Channel Mode registers Operations field. During state T_2 the channel drives both $\overline{DS}$ and $\overline{DACK}$ active. The "Flyby Peripheral" connected to $\overline{DACK}$ inverts $R/\overline{W}$ to determine whether it is being read from or written to.

The $\overline{DACK}$ input serves two purposes: To select the peripheral for the Read/Write, and to provide timing information on when to drive data onto or input data from the $AD_0 - AD_{15}$ bus. Note that because the "Flyby Peripheral" never gets explicitly addressed by $AD_0 - AD_{15}$, it must know which internal register to load from or drive onto the $AD_0 - AD_{15}$ bus. On state T_3, the $\overline{DS}$ and $\overline{DACK}$ lines are driven inactive to conclude the transfer. In Transfer-and-Search mode, data is loaded into the DTC's Temporary register on the LOW-to-HIGH $\overline{DS}$ transition in order to perform the search function.

To provide adequate data setup time the rising edge of $\overline{DS}$ or $\overline{DACK}$ should be the edge used to perform the write to the transfer destination. To extend the active time of $\overline{DS}$ and $\overline{DACK}$, wait states can be inserted between T_2 and T_3. Software wait states can be inserted by programming the appropriate code in the Tag field of the Current ARA or ARB registers. Hardware wait states can be inserted by pulling $\overline{CS}/\overline{WAIT}$ LOW if the Wait Line Enable bit in the Master Mode register is set. The $\overline{CS}/\overline{WAIT}$ line is sampled in the middle of the T_2 and TWA states.

Termination

There are three ways a Transfer-and-Search or Search operation can end and two ways a Transfer operation can end. When a channel's Current Operation Count goes to 0, the DMA operation being performed will end. This is called a TC or Terminal Count termination. A DMA operation can also be stopped by driving the $\overline{EOP}$ pin LOW with external logic. This is called an EOP termination. Search and Transfer-and-Search operations have a third method of terminating called Match Condition or MC termination. An MC termination occurs when the data being Transferred-and-Searched or Searched meets the match condition programmed in Channel Mode register bits $CM_{17} - CM_{16}$. These bits allow the user to stop when a match occurs between the unmasked Pattern register bits and the data read from the source, or when a no-match occurs. Both byte and word matches are supported. MC terminations do not apply to Transfer operations since the pattern matching logic is disabled in Transfer mode.

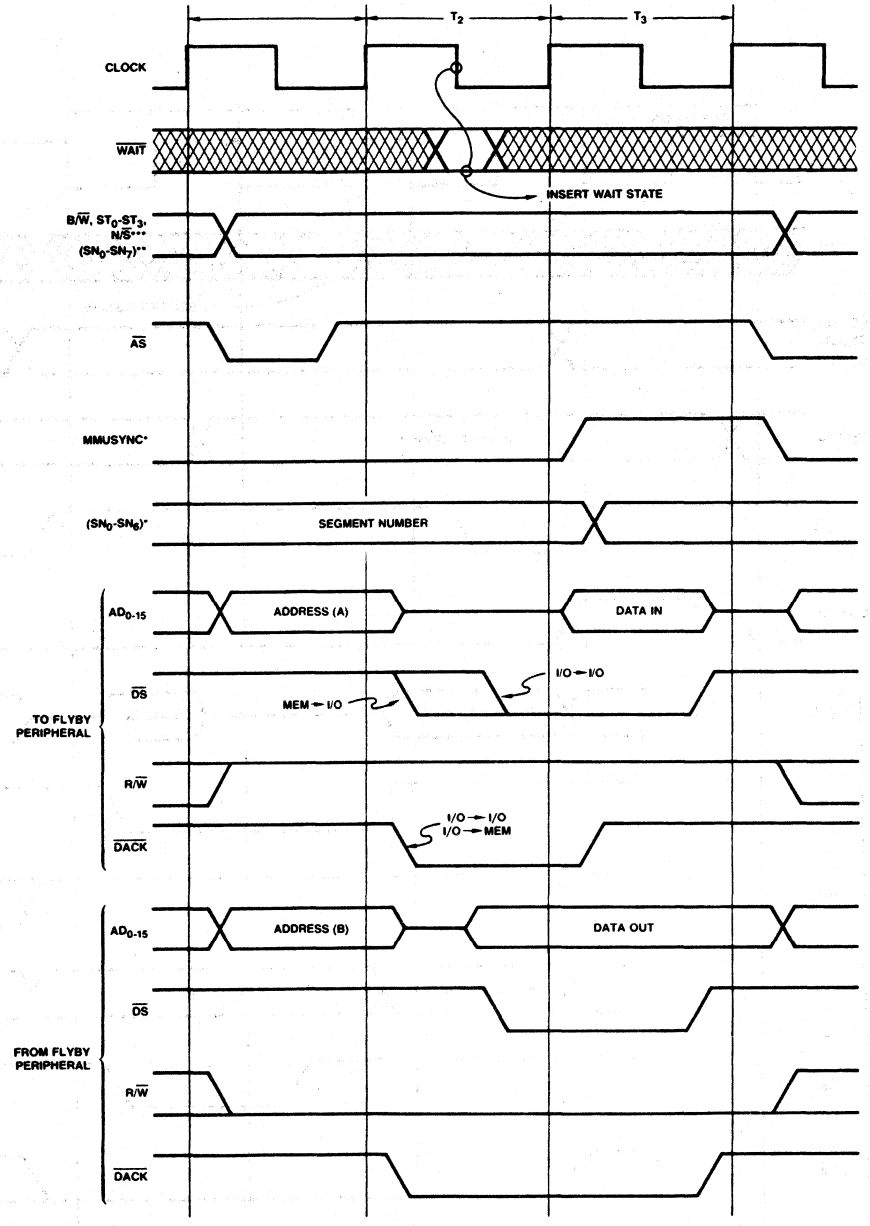


WF005230

*For logical addressing only
**For physical addressing only

Timing Diagram 5. Flowthru Memory Transactions

Refer to page 7-1 for Essential Information on Military Devices



WF005390

*For logical addressing only
**For physical addressing only
***N/S will be LOW for I/O to I/O Transactions
(A) Address is current ARA
(B) Address is current ARB

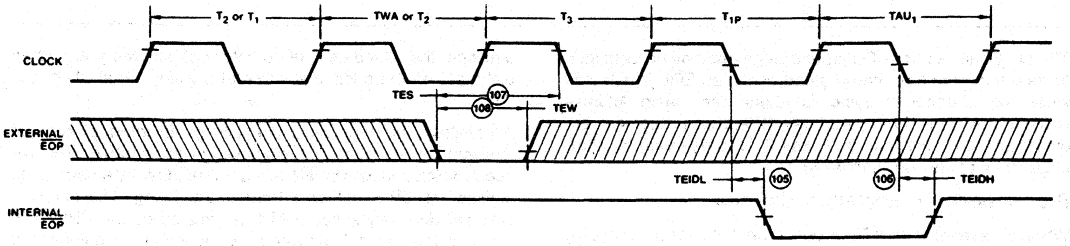
Timing Diagram 6. Flyby Transaction

Refer to page 7-1 for Essential Information on Military Devices

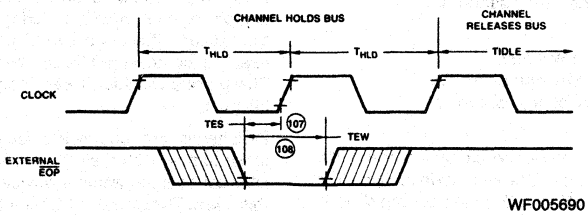
End-of-Process

The End-of-Process ($\overline{EOP}$) interface pin is a bidirectional signal. Whenever a TC, MC or $\overline{EOP}$ termination occurs, the DTC will drive the $\overline{EOP}$ pin LOW. During DMA operations, the $\overline{EOP}$ pin is sampled by the DTC to determine if $\overline{EOP}$ is being driven LOW by external logic. Timing Diagram 7 shows when internal $\overline{EOP}$s are generated marking termination of all Transfers. These figures also show the point during the DMA iteration when the $\overline{EOP}$ pin is sampled. The generation of internal $\overline{EOP}$s and sampling of external $\overline{EOP}$s for Transfer-and-Searches follows the same timing used for Transfers. Since there is a single $\overline{EOP}$ pin for both channels, $\overline{EOP}$ should only be driven LOW by a channel while that channel is being serviced. This can be accomplished by selecting a level $\overline{DACK}$ output (CMR 18 = 0) and gating each channel's $\overline{EOP}$ request with $\overline{DACK}$, as shown in Figure 8.

Some users may connect the $\overline{EOP}$ pin to the MMU's suppress ($\overline{SUP}$) output to detect the illegal memory accesses. To allow this abort feature for all memory accesses, the DTC samples $\overline{EOP}$ during chaining, as shown in Timing Diagram 7 (A). If $\overline{EOP}$ is LOW, the Chain Aborted bit in the active channel's Status register is set, the channel relinquishes bus control and the channel is inhibited from performing a new DMA operation until either a new Chain Address Segment-Tag word or offset word is loaded.



a) $\overline{EOP}$ Sampling and Generation During DMA Operations



b) Sampling of $\overline{EOP}$ During Bus Hold

Notes: 1. The diagram lists state names for both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_{1P} is a pseudo-T_1 state, without active $\overline{AS}$, generated following termination of any DMA operation.
3. State TAU_1 is an auto-initialization state, generated following the TC, MC or EOP termination.

Timing Diagram 7. $\overline{EOP}$ Timing

Refer to page 7-1 for Essential Information on Military Devices

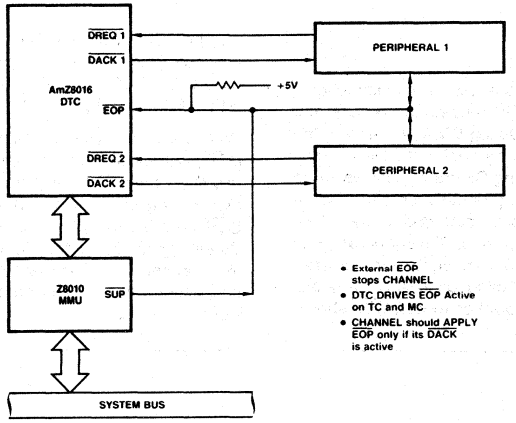


Figure 8. $\overline{EOP}$ Connection

The old Chain Address Offset and Segment may be examined to determine the error-causing address. If an $\overline{EOP}$ is detected while the channel is trying to reload the Chain Address register, the new Chain Address Offset and Segment are discarded and the old address+2 is preserved to allow inspection of the erroneous address.

Programming Completion Options

When a channel ends a DMA operation, the reason for ending is stored in the Completion Status Field of the channel's Status register. See Figure 16. This information is retained until the next DMA operation ends at which time the Status register is updated to reflect the reason(s) for the latest termination. Note that it is conceivable that more than one bit in the Completion Field could be set. As an extreme example, if a channel decremented it's Current Operation Count to zero, causing a TC termination; input data from the source generated a match causing an MC termination; and there was a LOW on $\overline{EOP}$ resulting in an EOP termination, all three of the channel's Status register completion bits would be set.

When a DMA operation ends, the channel can:

(a) Interrupt the host CPU;
(b) Perform Base-to-Current reloading;
(c) Chain reload the next DMA operation;
(d) Perform any combination of the above; or
(e) None of the above.

The user selects the action to be performed by the channel in the Completion option field of the Channel Mode register. For each type of termination (TC, MC or EOP) the user can choose which action or actions are to be taken. If no actions are selected for the type of termination that occured, the NAC bit in the Status register will be set.

More than one action can occur when a DMA operation ends. This may arise because more than one action was programmed for the applicable termination. They occur in the following order: interrupt, base-to-current reloading, and then chaining.

Interrupts

In order to allow the DTC to start executing a new DMA operation after issuing an interrupt, but before an interrupt acknowledge is received, a two-deep interrupt queue is implemented on each channel. The following discussion will describe the standard Z8000 interrupt structure and then elaborate on the additional interrupt queuing capability of the DTC.

A complete interrupt cycle consists of an interrupt request followed by an interrupt acknowledge transaction. The request, which consists of $\overline{INT}$ being pulled LOW by a peripheral, notifies the CPU that an interrupt is pending. The interrupt acknowledge transaction, which is initiated by the CPU as a result of the request, performs two functions: it selects the peripheral whose interrupt is to be acknowledged, and it obtains a vector that identifies the selected device and cause of interrupt.

A peripheral can have one or more sources of interrupt. Each interrupt source has three bits that control how it generates interrupts. These bits are a Channel Interrupt Enable bit (CIE), an Interrupt Pending bit (IP) and an Interrupt Under Service bit (IUS). On the DTC, each channel is an interrupt source. The three interrupt control bits are located in bits $CM_{15} - CM_{13}$ of each channel's Status register.

Each channel has its own vector register for identifying the source of the interrupt during an interrupt acknowledge transaction. There are two bits in the Master Mode register used for controlling interrupt behavior for the whole device. These are a Disable Lower Chain bit (DLC), and a No Vector bit (NV).

Peripherals are connected together via an interrupt daisy chain formed with their IEI and IEO pins (See Figure 9). The interrupt sources within a device are similarly connected into this chain. The overall effect is a daisy chain connecting all the interrupt sources. The daisy chain has two functions: during an interrupt-acknowledged transaction it determines which interrupt source is being acknowledged; at all other times it determines which interrupt sources can initiate an interrupt request.

Figure 10 is a state diagram for interrupt processing for an interrupt source. An interrupt source with an interrupt pending (IP = 1) makes an interrupt request (by pulling $\overline{INT}$ LOW) if, all of the following conditions are met: It is enabled (CIE = 1), is does not have an interrupt under service (IUS = 0), no higher priority interrupt is being serviced (IEI is HIGH), and no interrupt-acknowledge transaction is in progress. IEO is not pulled down by the interrupt source at this time; IEO continues to follow IEI until an interrupt-acknowledge transaction occurs.

Refer to page 7-1 for Essential Information on Military Devices

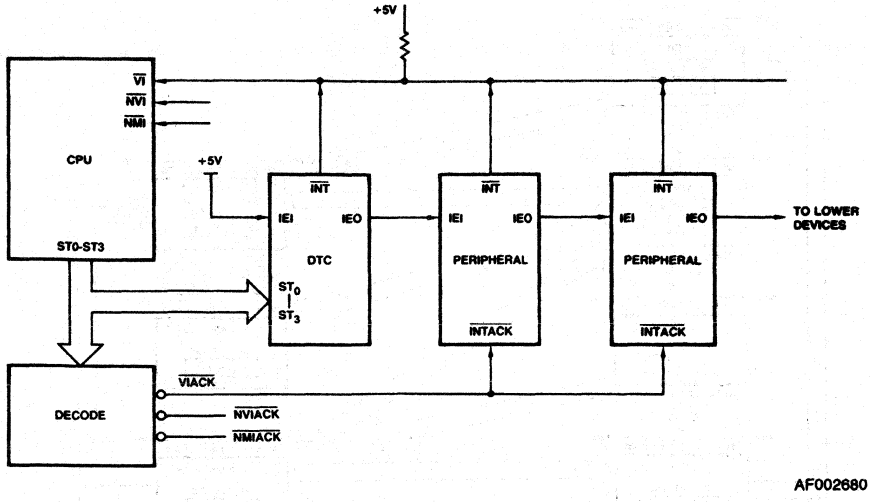


Figure 9. Interrupt Daisy Chain

Most Z8000 peripherals have an $\overline{INTACK}$ pin to signal when an interrupt acknowledge cycle is being performed. The DTC uses a different approach of monitoring status lines $ST_0 - ST_3$ to detect interrupt acknowledge cycles. It is important that the Master Mode register bits MM_6 and MM_7 be programmed to select the IEI, IEO daisy chain that the DTC is located on. Some time after $\overline{INT}$ has been pulled LOW, the CPU initiates an interrupt-acknowledge transaction. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the IEI/IEO daisy chain settles.

Once a channel issues an interrupt, it is desirable to allow the channel to proceed with the next DMA operation before the interrupt is acknowledged. This could lead to problems if the DTC channel attempted to chain reload the Vector register contents. In such a situation, it may not be clear whether the old or new vector would be returned during the acknowledge. This dilemma is resolved in the DTC by providing each channel with an Interrupt Save Register. When the channel sets IP as part of the procedure followed to issue an interrupt, the contents of the Vector register and some of the Status register bits are saved in an Interrupt Save register. See Figure 17. When an Interrupt Acknowledge cycle is performed, the contents of the Interrupt Save register are driven onto the bus. Although the use of an Interrupt Save register allows the channel to proceed with a new task, problems can still potentially arise if a second interrupt is to be issued by the channel before the first interrupt is acknowledged. To avoid conflicts between the first and second interrupt, each channel has a Second Interrupt Pending (SIP) bit in its Status register. When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit is set and the channel relinquishes the bus until an acknowledge occurs. For compatibility with polled interrupt schemes, the Interrupt save register can be read by the host CPU without wait states. As an aid to debugging a system's interrupt logic, whenever IP is set, the Interrupt Save register is loaded from the Vector and Status register.

Note that the SIP bit is transferred to the IP bit when IP is cleared by the host CPU. Whenever IEI is HIGH, CIE is set and IUS is cleared, $\overline{INT}$ will go LOW as soon as IP is set. IP can be cleared as soon as the first interrupt is acknowledged. The acknowledge will, as always, automatically set IUS.

The DTC stops driving $\overline{INT}$ low as soon as IUS is set. IUS must be cleared by the CPU before $\overline{INT}$ can be driven low for the second time.

Base-to-Current Reloading

When a channel finishes a DMA operation, the user may select to perform a Base-to-Current Reload. (Base-to-Current reloading is also referred to as Auto-reloading is this document.) In this type of reload, the Current Address Registers A and B are loaded with the data in the Base Address Registers A and B respectively, and the Current Operation Count register is loaded with the data in the Base Operation Count. The Base-to-Current reload operation facilitates repetitive DMA operations without the multiple memory accesses required by chaining. Although the channel must have bus control to perform Base-to-Current reloading, the complete reloading operation occurs in four clock cycles. Note that if the channel had to relinquish the bus because two unacknowledged interrupts were queued, it will have to regain bus control to perform any Base-to-Current reloading (or chaining, for that matter). In this case it acquires the BUS once an interrupt acknowledge is received, even if it immediately afterward will relinquish the bus because no hardware or software request is present.

Chaining

If the channel is programmed to chain at the end of a DMA operation, it will use the Chain Address register to point to a Chain Control Table in memory. The first word in the table is a Reload word, specifying the register(s) to be loaded. Following the Reload word are the data values to be transferred into register(s). Chaining is described in detail in the "Channel Initialization" section.

Because chaining occurs after Base-to-Current reloading, it is possible to reset the Current Address registers A and B and the Current Operation Count register to the values used for previous DMA operations and then chain reload one or two of these registers to some special value to be used perhaps for this DMA operation only. If the Base values are not reloaded during chaining, the channel can revert back to the Base values at a later cycle.

Refer to page 7-1 for Essential Information on Military Devices

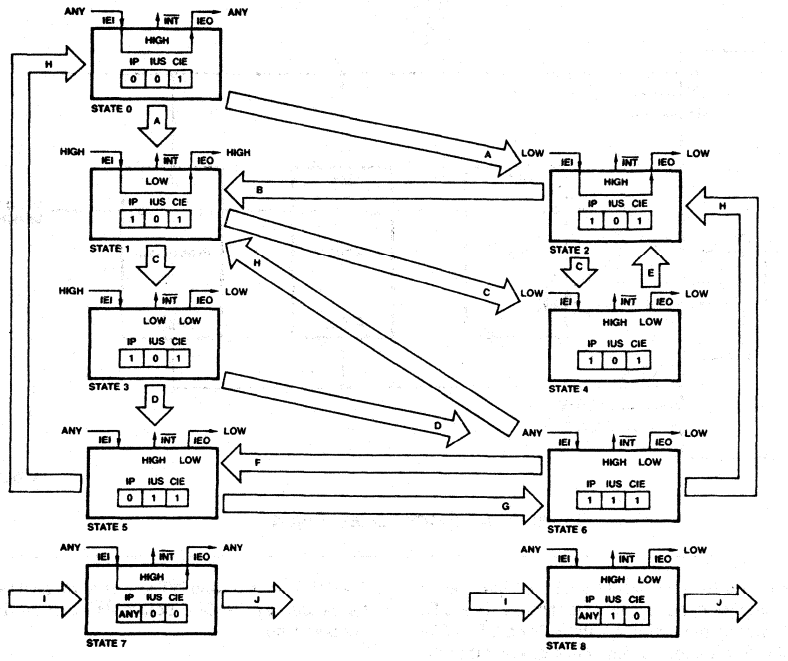


Figure 10. State Diagram for Interrupt Processing

Transition Legend

A The peripheral detects an interrupt condition and sets Interrupt Pending.

B All higher priority peripherals finish interrupt service, thus allowing IEI to go HIGH.

C An interrupt acknowledge transaction starts and the IEI/IEO daisy chain settles.

D The interrupt acknowlege transaction terminates with the peripheral selected. Interrupt Under Service (IUS) is set to 1, and Interrupt Pending (IP) may not be reset.

E The interrupt acknowlege transaction terminates with a higher priority device having been selected.

F The Interrupt Pending bit in the peripheral is reset by an I/O operation.

G A new interrupt condition is detected by the peripheral, causing IP to be set again.

H Interupt service is terminated for the peripheral by resetting IUS.

I CIE is reset to zero, causing interrupts to be disabled (Note 1).

J CIE is set to one, re-enabling interrupts (Note 2).

State Legend

0 No interrupts are pending or under service for this peripheral.

1 An interrupt is pending, and interrupt request has been made by pulling $\overline{INT}$ LOW.

2 An interrupt is pending, but no interrupt request has been made because a higher priority peripheral has an interrupt under service, and this has forced IEI LOW.

3 An interrupt acknowledge sequence is in process, and no higher priority peripheral has a pending interrupt.

4 An interrupt acknowledge sequence is in progress, but a higher priority peripheral has a pending interrupt, forcing IEI LOW.

5 The peripheral has an interrupt under service. Service may be temporarily suspended (indicated by IEI going LOW) if a higher priority device generates an interrupt.

6 This is the same as State 5 except that an interrupt is also pending in the peripheral.

7 Interrupts are disabled from this source because CIE = 0.

8 Interrupts are disabled from this source and lower priority sources because CIE = 0 and IUS = 1.

Notes: 1. Transition I to state 6 or 7 can occur from any state except 3 or 4 (which only occur during interrupt acknowledge).
2. Transition J from state 6 or 7 can be to any state except 3 or 4, depending on the value of IEI, IP, and IUS.

Refer to page 7-1 for Essential Information on Military Devices

If an all zero Reload word is fetched during chaining, the chain operation will not reload any registers but in all other respects it will perform like any other chaining operation. Thus, the Chain Address will be incremented by 2 point to the next word in memory and at the end of the all Zero-Reload word chain operation the channel will be ready to perform a DMA operation. All zero Reload words are useful as "Stubs" to start or terminate linked lists of DMA operations traversed by chaining. On the other hand, care must be taken in their use since the channel may perform an erroneous operation if it is unintentionally started after the chaining operation.

Command Descriptions

Figure 11 shows a list of DTC commands. The commands are executed immediately after being written by the host CPU into the DTC's Command register. A description of each command follows.

Reset

This command causes the DTC to be set to the same state generated by a Hardware Reset. The Master Mode register is set to all zeros and the NAC and CA bits in each channel's Status register are set. The Chain Address should be programmed since its state may be indeterminate after a Reset. The lockout preventing channel activity is cleared by programming the Segment/Tag word or the Offset word and then issuing a "Start Chain" command.

Start Chain Channel 1/Channel 2

This command causes the selected channel to clear the No Auto-Reload or Chain (NAC) bit in the channel's Status register, and to start a chain reload operation of the channel's registers, as described in the "Channel Initialization" section. These effects will take place even if the Reload word fetched is all zeros. This command will only be honored if the Chain Abort (CA) bit and Second Interrupt Pending (SIP) bit in the Channel's Status register are clear. If either the CA or SIP bit is set, this command is disregarded.

Software Request Channel 1/Channel 2

This command sets the software request bit in the selected channel's Mode register. If the Second Interrupt Pending (SIP) bit and No Auto-Reload or Chain (NAC) bit in the channel's Status register are both clear, the channel will start executing the programmed DMA operation. If either the SIP or NAC bit is set, the channel will not start executing a DMA operation until both bits are cleared. The SIP bit will clear when the channel receives an interrupt acknowledge. One way to clear the NAC bit is to issue a Start Chain command to the channel. If the fetched Reload Word is all zeros, the channel's registers will remain unchanged and the software request bit, if set earlier by command, will cause the programmed DMA operation to start immediately. If during chaining new information is loaded into the Channel Mode register this new information will, of course, overwrite the software request bit.

Command	Opcode Bits 7654	Opcode Bits 3210	Example Code (HEX)
Reset	000X	XXXX	00
Start Chain Channel 1	101X	XXX0	A0
Start Chain Channel 2	101X	XXX1	A1
Set Software Request Channel 1	010X	XX10	42
Set Software Request Channel 2	010X	XX11	43
Clear Software Request Channel 1	010X	XX00	40
Clear Software Request Channel 2	010X	XX01	41
Set Hardware Mask Channel 1	100X	XX10	82
Set Hardware Mask Channel 2	100X	XX11	83
Clear Hardware Mask Channel 1	100X	XX00	80
Clear Hardware Mask Channel 2	100X	XX01	81
Set CIE, IUS, IP Channel 1	001E	SP10	32*
Set CIE, IUS, IP Channel 2	001E	SP11	33*
Clear CIE, IUS, IP Channel 1	001E	SP00	30*
Clear CIE, IUS, IP Channel 2	001E	SP01	31*
Set Flip Bit Channel 1	011X	XX10	62
Set Flip Bit Channel 2	011X	XX11	63
Clear Flip Bit Channel 1	011X	XX00	60
Clear Flip Bit Channel 2	011X	XX01	61

*Notes: 1. E = Set to 1 to perform set/clear on CIE. Clear to 0 for no effect on CIE.
2. S = Set to 1 to perform set/clear on IUS. Clear to 0 for no effect on IUS.
3. P = Set to 1 to perform set/clear on IP. Clear to 0 for no effect on IP.
4. X = "don't care" bit. This bit is not decoded and may be 0 or 1.

Figure 11. DTC Command Summary

Set/Clear Hardware Mask 1/Mask 2

This command sets or clears the Hardware Mask bit in the selected channel's Mode register. This command always takes effect. The Hardware Mask bit inhibits recognition of an active signal on the channel's $\overline{DREQ}$ input; this bit does not affect recognition of a software request. If the channel is in single transfer mode, it performs DMA operations upon receipt of a transition on $\overline{DREQ}$ rather than in response to a $\overline{DREQ}$ level; the transition occurring while the Hardware Mask bit is set (CM_{19} = 1) will be stored and serviced when the Hardware Mask is cleared (assuming the channel has not chained in the interim). The DTC will request the system bus 1-1/2 to 2 clocks after the receipt of any $\overline{DREQ}$, after which a minimum of one DMA iteration is unavoidable. $\overline{DREQ}$ transitions are only stored for the current DMA operation. If the channel performs a chain operation of single transfer mode, any $\overline{DREQ}$ transition stored for later servicing is cleared. Timing Diagrams 1 and 2 show the minimum times after each of these events a new $\overline{DREQ}$ edge can be applied if it is to be serviced by the new DMA operation. Note in the diagrams the notation of First iteration and Last iteration. This means, for example, $\overline{DREQ}$ may be asserted during the second T_1 of a Flow-through transaction, but may never be asserted during T_1 of a Flyby transaction because Flyby is done in one iteration.

Refer to page 7-1 for Essential Information on Military Devices

Set/Clear CIE, IUS AND IP Channel 1/Channel 2

This command allows the user to either set or clear any combination of the CIE, IUS and IP bits in the selected channel's Status register. These bits control the operation of the channel's interrupt structure and are described in detail in the "Interrupts" section. Setting the IP bit causes the Interrupt Save register to be loaded with current Vector and Status. The IP and IUS bits can be simultaneously cleared to facilitate an efficient conclusion to the processing of an interrupt.

Set/Clear Flip Bit in Channel 1/Channel 2

The Flip Bit in the selected channel's Mode register can be cleared and set by this command. This allows the user to reverse the source and destination and thereby reverse the data transfer direction without reprogramming the channel. This command will be most useful when repetitive DMA operations are being performed by the channel, using Base-to-Current reloading for channel reinitialization and using this command to control the direction of transfer. Chaining new information into the Channel Mode register will, of course, overwrite the Flip bit.

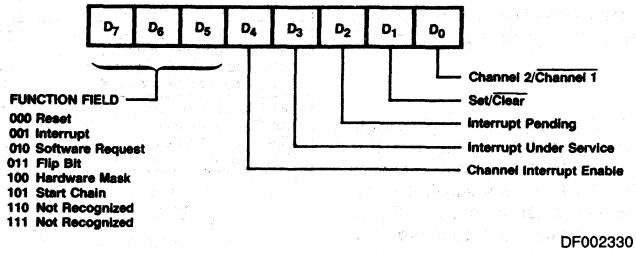


Figure 11A. Command Register

PROGRAMMING INFORMATION

Register Description

The AmZ8016 block diagram illustrates the internal registers. Figure 12 lists each register along with its size and read/write access restrictions. Registers which can be read by the CPU are either fast (F) or slow (S) readable. Fast registers can be read by an I/O operation without additional wait states. Reading slow registers requires multiple wait states (see Timing Parameters-88 for requirement). It is the responsibility of the user to supply the necessary external logic if slow readable registers are to be read. Registers can be written to by the host CPU (W) and/or can be loaded by the DMA channel itself during chaining (C). All reads or writes must be word accesses since the DTC ignores the B/$\overline{W}$ line in slave mode.

The DTC registers can be categorized into chip-level registers, which control the overall operation and configuration of the DTC, and channel-level registers which are duplicated for each channel. The four chip-level registers are the Master Mode register, the Command register, the Chain Control register and the Temporary register. The Master Mode register selects the way the DTC chip interfaces to the system. The Command register is written to by the host CPU to initiate certain operations within the DTC chip, such as resetting the unit. The Chain Control register is used by a channel while it is reloading its channel-level registers from memory. The Temporary register is used to hold data for Flowthru Transfer-and-Searches.

Master Mode Register

The 8-bit Master Mode register, shown in Figure 13, controls the chip-level interfaces. It can be read from and written to by the host CPU without wait states through pins $AD_0 - AD_7$ but it is not loadable by chaining. On a reset, the Master Mode register is cleared to all zeroes. The function of each of the Master Mode bits is described in the following paragraphs.

The Chip Enable bit CE = 1 enables the DTC to request the bus. When enabled, the DTC can perform DMA Operations and reload registers. It can always issue interrupts and respond to interrupt acknowledges. When the Chip Enable bit is cleared, the DTC is inhibited from requesting control of the system bus and, therefore, inhibited from performing chaining or DMA operations. The Chip Enable bit (MMO) should not be used as the gating item in starting or stopping the DTC. Channels should be enabled or disabled by using the Set/Clear software request or Hardware Mask Commands. The $\overline{BAO}$ signal follows $\overline{BAI}$ while the Chip Enable bit is cleared.

The Logical/$\overline{Physical}$ Address space bit selects the address space in which the DTC resides. Figure 14 shows the different configuration options. If the DTC outputs addresses which are translated by an MMU, Logical space must be selected. If the addresses output by the DTC pass directly onto the system backplane and if the host CPU is an Z8001 using an MMU, the Logical/$\overline{Physical}$ bit must be set to Physical ($MM_1 = 0$). If Z8001 addresses are not translated by an MMU, the DTC must be set to Logical ($MM_1 = 1$). In an Z8002 based system, the user may use either the Physical or Logical address space setting.

Refer to page 7-1 for Essential Information on Military Devices

Name	Size	Number	Access Type	Port Address CH-1/CH-2
Master Mode Register	8 bits	1	FW	38
Chain-Control Register	10 bits	1	C	
Temporary Register	16 bits	1	D	
Command Register	8 bits	1	W	2E/2C*
Current Address Register — A:				
Segment/Tag field	15 bits	2	CFW	1A/18
Offset field	16 bits	2	CFW	0A/08
Current Address Register — B:				
Segment/Tag field	15 bits	2	CFW	12/10
Offset field	16 bits	2	CFW	02/00
Base Address Register — A:				
Segment/Tag field	15 bits	2	CFW	1E/1C
Offset field	16 bits	2	CFW	0E/0C
Base Address Register — B:				
Segment/Tag field	15 bits	2	CFW	16/14
Offset field	16 bits	2	CFW	06/04
Current Operation Count	16 bits	2	CFW	32/30
Base Operation Count	16 bits	2	CFW	36/34
Pattern Register	16 bits	2	CSW	4A/48
Mask Register	16 bits	2	CSW	4E/4C
Status Register	16 bits	2	F	2E/2C
Interrupt Save Register	16 bits	2	F	2A/28
Interrupt Vector Register	8 bits	2	CSW	5A/58
Channel Mode Register — High	5 bits	2	CS	56/54
Channel Mode Register — Low	16 bits	2	CSW	52/50
Chain Address Register:				
Segment/Tag field	10 bits	2	CFW	26/24
Offset field	16 bits	2	CFW	22/20

Access Codes: C = Chain Loadable
D = Accessible by DTC channel
F = Fast Readable
S = Slow Readable
W = Writable

Note: Upper Register Address is determined by user's Chip Select Decode Logic. Only Lower Register Address is shown here.
*The port addresses of Command register can be used alternately for both channels except when issuing a "set/clear IP" command.

Figure 12. DTC Internal Register

When set to Logical address space, the segment and offset portions of the Current ARA and ARB registers are viewed as separate portions of the address. Incrementing and decrementing the register affects only the offset portion of the address; no carry or borrow signal is generated into the segment. Only the lower 7 bits of the segment field are used; the setting of the most significant segment bit is disregarded. The 16-bit offset portion of the address appears on pins $AD_0 - AD_{15}$ when $\overline{AS}$ is LOW and the 7-bit segment number appears on pins $SN_0 - SN_6$ for the duration of the transaction. The SN_7/MMUSync signal outputs a HIGH pulse prior to each memory transaction, and is never three-stated.

When the Logical/$\overline{\text{Physical}}$ Space bit is set to Physical, the segment and offset portions of the Current ARA and ARB registers are treated as a single linear address. All eight segment bits in the register are used. When an address is incremented or decremented, the carry/borrow signal propagates across the full 24-bit address updating both the segment and offset portions of the address. Both I/O and memory addresses in Physical space are generated by driving the offset portion of the Current Address register onto the $AD_0 - AD_{15}$ bus and driving the segment portion of the Current Address register onto the $SN_0 - SN_7$ bus. Timing Diagrams 4, 5 and 6 show how the Logical/$\overline{\text{Physical}}$ Space bit affects the DTC timing. The $AD_0 - AD_{15}$ timing is not affected. The $SN_0 - SN_7$ lines are shifted from T_3 to T_1 so that they are valid during the entire transaction. In the Logical Address mode, the MMU insures the addresses are valid during the entire transaction.

The CPU Interleave bit enables interleaving between the CPU and the DTC.

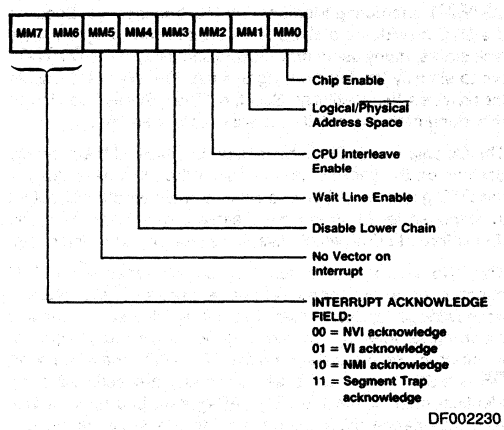
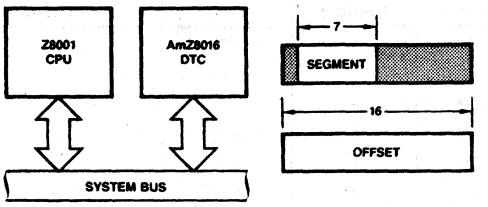
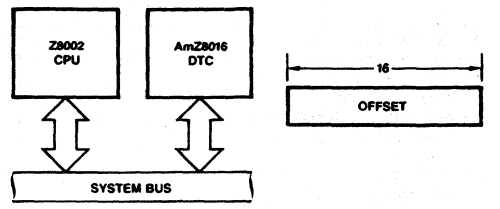


Figure 13. Master Mode Register



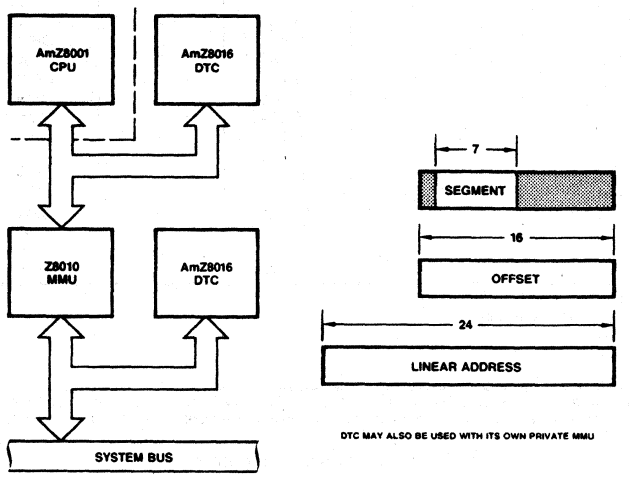
AF002711

a) DTC with Z8001 (Segmented) CPU



AF002731

b) DTC with Z8002 (Nonsegmented) CPU



AF002720

c) DTC with Z8001 (Segmented) CPU and AmZ8010 MMU

Figure 14. DTC Configuration Options

The Wait Line Enable bit is used to enable sampling of the $\overline{CS}/\overline{WAIT}$ line during Memory and I/O transactions. Because the DTC provides the ability to insert software programmable wait states, many users may disable sampling of the $\overline{CS}/\overline{WAIT}$ pin to simplify the logic driving this pin. The Wait Line Enable bit provides this flexibility. See the "Wait States" section of this document for details on wait state insertion.

The Disable Lower Chain bit is used to inhibit all lower priority devices on the interrupt daisy chain. When this bit is cleared, the DTC generates LOW and HIGH signals on the IEO output in response to IEI. When the Disable Lower Chain bit is set, IEO is forced LOW, which disables all lower priority interrupts.

The "No Vector on Interrupt" bit selects whether the DTC channel or a peripheral returns a vector during interrupt acknowledge cycles. When this bit is cleared, a channel receiving an interrupt acknowledge will drive the contents of its Interrupt Save register onto the $AD_0 - AD_{15}$ data bus while $\overline{DS}$ is LOW. If this bit is set, interrupts are serviced in an identical manner but the $AD_0 - AD_{15}$ data bus remains in a high impedance state throughout the acknowledge cycle.

The Vectored/Non-Vectored/Non-Maskable/Segment Trap Field of two bits selects which type of interrupt acknowledge cycle the DTC is to respond to. The DTC decodes from $ST_0 - ST_3$ that an interrupt acknowledge cycle is underway. The setting of this 2-bit field must correspond to the IEI/IEO daisy chain on which the DTC is located to prevent unpredictable results. For example a DTC programmed for vectored interrupts should not be placed on the non-vectored priority chain.

Chain Control Register

When a channel starts a chaining operation, it fetches a Reload word from the memory location pointed to by the Chain Address register. This word is then stored in the Chain Control register. The Chain Control register cannot be written to or read from by the CPU. Once a channel starts a chain operation, the channel will not relinquish bus control until all registers specified in the Reload word are reloaded unless an $\overline{EOP}$ signal is issued to the chip. Issuing an $\overline{EOP}$ to a channel during chaining will prevent the chain operation from resuming and the contents of the Reload Word register can be discarded.

Temporary Register

The Temporary register is used to stage data during Flowthru transfers and to hold data being compared during a Search or a Transfer-and-Search. The Temporary register cannot be written to, or read from by the CPU. In byte-word funneling, data may be loaded into or from the Temporary register on a byte-by-byte basis, with bytes sometimes moving between the low byte of the data bus and the high byte of the Temporary register or vice-versa. See the "Transfer" section for details.

Refer to page 7-1 for Essential Information on Military Devices

Command Register

The DTC Command register is an 8-bit write-only register written to by the host CPU to execute commands. The Command register is loaded from the data on $AD_7 - AD_0$; the data on $AD_{15} - AD_8$ is disregarded. A complete discussion of the commands is given in the "Command Descriptions" section.

Current and Base Address Registers A and B

The Current Address registers A and B (Current ARA and ARB) are used to point to the source and destination addresses for DMA operations. The Base ARA and ARB register contents are transferred into the Current ARA and ARB registers at the end of a DMA operation if the user enables Base-to-Current reloading in the Completion Field of the Channel Mode register. This facilitates DMA operations without reloading of the Current registers. The ARA and ARB registers can be loaded during chaining, can be written to by the host CPU without wait states and can be read by the CPU.

Each of the Base and Current ARA and ARB registers consists of two words organized as a 7-bit Tag Field and an 8-bit segment in one word and a 16-bit offset in the other. See Figure 6. The Segment and Offset contain the actual address driven onto the bus. The Tag Field selects whether the address is to be incremented, decremented or left unchanged, and the status codes associated with the address. The Tag field also allows the user to insert 0, 1, 2 or 4 wait states into memory or I/O accesses addressed by the offset and segment fields.

The Address Reference Select Field in the Tag field selects whether the address pertains to memory space or I/O space. Note that the N/$\overline{S}$ output pin is always LOW (indicating System) for I/O space but may be either HIGH (indicating Normal) or LOW (indicating System) for memory space, as selected in the Address Space field. At the end of each iteration of a DMA Operation, the user may select to leave the address unchanged or to increment it or to decrement it. I/O addresses, if changed, are always incremented/decremented by 2. Memory addresses are changed by 1 if the address points to a byte operand (as programmed in the Channel Mode registers Operation field) and by 2 if the address points to a word operand. Note that if an I/O or memory address is used to point to a word operand, the address must be even to avoid unpredictable results. An address used to point to a byte operand may be even or odd. Since memory byte operand addresses will increment/decrement by 1, they will toggle between even and odd values. Since I/O byte operand addresses will increment/decrement by 2, once programmed to an even or odd value, they will remain even or odd, allowing consecutive I/O operations to access the same half of the data bus. High bus is for even address and low bus for odd.

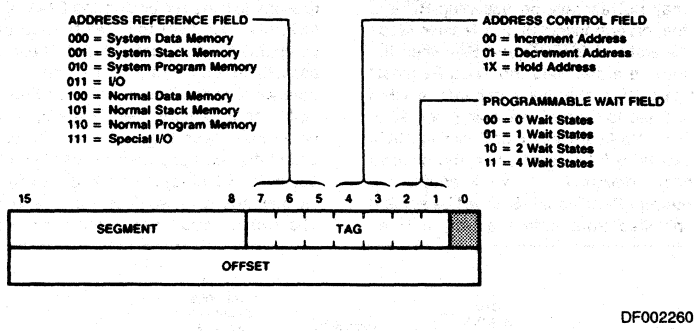


Figure 15. Address Register A and B

Current and Base Operation Count Registers

Both the Current and Base Operation Count registers may be loaded during chaining, and may be written to, and read from by the host CPU.

The 16-bit Current Operation Count register is used to specify the number of words or bytes to be transferred, searched or transferred-and-searched. For word-to-word operations and byte-word funneling, the Current Operation Count register must be programmed with the number of words to be transferred or searched.

Each time a datum is transferred or searched, the Operation Count register is decremented by 1. Once all of the data is transferred or searched, the transfer or search operation will stop, the Current Operation Count register will contain all zeros, and the TC bit in Status Register will be '1'. If the transfer or search stops before the Current Operation Count register reaches 0, the contents of the register will indicate the number of bytes or words remaining to be transferred or searched. This allows a channel which had been stopped prematurely, to be restarted where it left off without requiring reloading of the Current Operation Count register.

If the DTC is configured for Physical Address Space operation (Master Mode register bit $MM_1 = 0$), the maximum number of words that can be transferred or searched is 64K words. This is specified by setting a word count of 0000. If the DTC is configured for Logical Address Space operation (Master Mode register bit $MM_1 = 1$), the maximum number of words specified to be transferred or searched with either an incrementing or decrementing source or destination address is 32K (8000 hex). This is because in Logical Address Space, offset addresses incremented past FFFF (hex) or decremented below 0000 do not increment/decrement the segment number. Thus, after transferring or searching more than 32K words, the address wraps around within the segment over the same data previously transferred or searched.

For the byte-to-byte operations, the Current Operation Count register should specify the number of bytes to be transferred or searched. The maximum number of bytes which can be specified is 64K bytes; by setting the Current Operation Count register to 0000.

Pattern and Mask Registers

The 16-bit Pattern and Mask registers are used in Search and Transfer-and-Search operations. Both the Pattern and Mask registers may be loaded by chaining, may be written to by the host CPU, and may be read from by the host CPU, provided wait states are inserted, since these registers are slow readable. The Pattern register contains the pattern that the read data is compared to. Setting a Mask register bit to '1' specifies that the bit always matches. See the "Search" and "Transfer-and-Search" sections for further details.

Status Register

The two 16-bit Status registers, depicted in Figure 16, are read-only registers which can be read by the CPU without wait states. Each of these registers reports on the status of its associated channel.

The Interrupt Status Field in the Status register contains the Channel Interrupt Enable (CIE), Interrupt Pending (IP) and Interrupt Under Service (IUS) bits. These bits are described in detail in the "Interrupt" section of this document.

The DTC status field reports on the current channel state to the CPU. The "channel initialized and waiting for request" status is not explicitly stated – it is reflected by Status register bits ST_{12} through ST_9 being all zero. The "Waiting for Bus" (WFB) status will cause bit ST_{10} to be set and indicates that the channel wants bus control to perform a DMA operation. The channel may or may not actually be asserting $\overline{BUSRQ}$ LOW, depending on the programming of the Master Mode Chip Enable bit and the state of $\overline{BUSRQ}$ and $\overline{BAI}$ when the channel decided it wanted the bus. See the "Bus Request/ Grant" section for details. If a channel completes a DMA operation and neither Base-to-Current reloading nor auto-chaining were enabled, the No Auto-Reload or Chaining (NAC) bit will be set. The NAC bit will be reset when the channel receives a "Start Chain Command." If two interrupts are queued, the Second Interrupt Pending bit (SIP) will be set and the channel will be inhibited from further activity until an interrupt acknowledge occurs. See the "Interrupt" section for details. Finally, if the channel is issued an $\overline{EOP}$ during chaining, the Chain Abort (CA) and the NAC will be set. These bits are also set when a "reset" is issued to the DTC. The CA bit holds the NAC bit in the set state. The CA is cleared when a new Chain Address Segment and Tag word or offset word is loaded into the channel.

The Hardware Interface Field provides a Hardware Request (HRQ) bit which provides a means of monitoring the channels $\overline{DREQ}$ input pin. When the $\overline{DREQ}$ pin is LOW, the HRQ bit will be '1' and vice-versa. The Hardware Mask (HM) bit, when set, prevents the DTC from responding to a LOW on $\overline{DREQ}$. Note, however, that the Hardware Request bit always reports the true (unmasked) status of $\overline{DREQ}$ regardless of the setting of the HM bit. The HM bit can be cleared by software command.

The Completion Field stores data at the end of each DMA operation. This data indicates why the DMA operation ended. When the next DMA operation ends, new data is loaded into these bits overwriting, and thereby erasing the old setting. Three bits indicate whether the DMA operation ended as a result of a TC, MC or EOP termination. The TC bit will be '1' if the Operation Count reaching zero ended the DMA operation. The MC bit will be '1' if an MC termination occurred regardless of whether Stop-on-Match or Stop-on-no-Match was selected. The EOP bit is set only when an $\overline{EOP}$ ends a DMA transfer; it is not set for EOPs issued during chaining. Note that two or even all three of MC, TC and $\overline{EOP}$ may be set if multiple reasons existed for ending the DMA operation. The MCH and MCL bits report on the match state of the upper and lower comparator bytes respectively. These bits are set when the associated comparator byte has a match and are reset otherwise, regardless of whether Stop-on-Match or Stop-on-no-Match is programmed. Regardless of the DMA operation performed, these bits will reflect the comparator status at the end of the DMA operation. These two bits are provided to help determine which byte matched or didn't match when using 8-bit matches with word searches and transfer-and-searches. The two reserved bits return zeroes during reads.

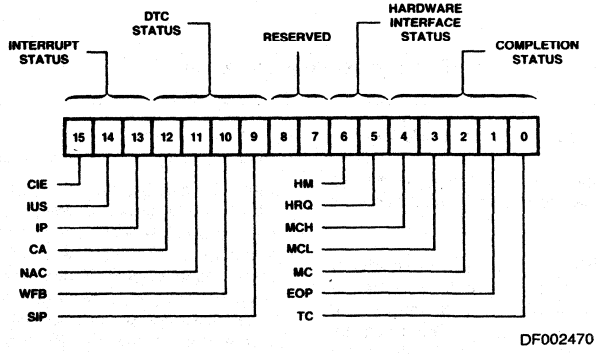


Figure 16. Status Register

Refer to page 7-1 for Essential Information on Military Devices

Interrupt Vector and Interrupt Save Registers

Each channel has an Interrupt Vector register and an Interrupt Save register. The Interrupt Vector is 8-bits wide and is written to and read from on $AD_0 - AD_7$. The Interrupt Save register may be read from by the CPU without wait states. The Interrupt Vector register contains the vector or identifier to be output during an Interrupt Acknowledge cycle. When an interrupt occurs (IP = 1), either because a DMA operation terminated or because the $\overline{EOP}$ pin was driven LOW during chaining, the contents of the Interrupt Vector register and part of the Channel Status register are stored in the 16-bit Interrupt Save register (See Figure 17). Because the vector and status are stored, a new vector can be loaded during chaining, and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. If another interrupt occurs on the channel before the first is acknowledged, further channel activity is suspended.

As soon as the first clear IP command is issued, the status and vector for the second interrupt is loaded into the Interrupt Save register and channel operation resumes. The DTC can retain only two interrupts for each channel; a third operation cannot be initiated until the first interrupt has been cleared. See the "Interrupt" section for further details.

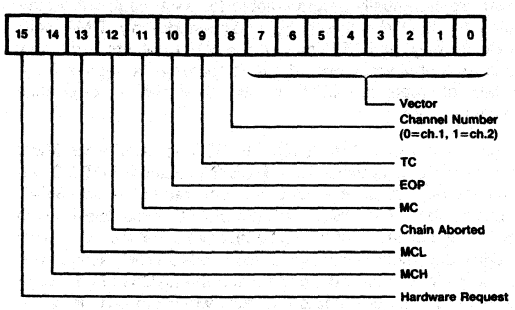


Figure 17. Interrupt Save Register

Channel Mode Register

Associated with each channel is a Channel Mode register. There are 21 bits defined in each Channel Mode register; the other 11 bits are unused. See Figure 18. The Channel Mode registers may be loaded during chaining and may be read from and written to by the host CPU. CPU reads from the Channel Mode register are slow reads and require insertion of multiple wait states. The Channel Mode register selects what type of DMA operation the channel is to perform, how the operation is to be executed, and what action, if any, is to be taken when the channel finishes.

The Data Operation Field and the Transfer Type field select the type of operation the channel is to perform. They also select the operand size of bytes or words see Figure 19 for code-definition. The different types of operations are described in detail in the "DMA Operations" section. The Flip bit is used to select whether the Current ARA register points to the source and the Current ARB register points to the destination or vice-versa.

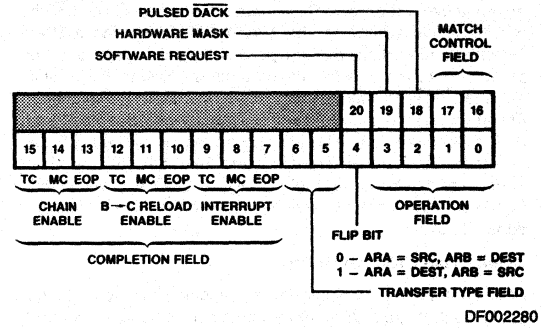


Figure 18. Channel Mode Register

DATA OPERATION FIELD			
	Operand Size		
Code/Operation	ARA	ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

TRANSFER FIELD AND MATCH CONTROL FIELD		
Code	Transfer Type	Match Control
00	Single Transfer	Stop on No Match
01	Demand (Bus Hold)	Stop on No Match
10	Demand (Bus Release)	Stop on Word Match
11	Demand Interleave	Stop on Byte Match

Figure 19. Channel Mode Coding

The Completion Field is used to program the action taken by the channel at the end of a DMA operation. This field is discussed in the "Completion Options" section. The 2-bit Match Control field selects whether matches use an 8-bit or 16-bit pattern and whether the channel is to stop-on-match or stop-on-no-match. See Figure 19 and the "Search" section for details. The Software Request bit and Hardware Mask bit can be set and cleared by software command. Only the lower 16 bits can be loaded in parallel with a CPU instruction. These bits are described in detail in the "Initiating DMA Operations" section.

The $\overline{DACK}$ Control bit is used to specify when the $\overline{DACK}$ pin is driven active. When this bit is cleared, the channel's $\overline{DACK}$ pin will be active whenever the channel is performing a DMA Operation, regardless of the type of transaction. Note that the pin will not be active while the channel is chaining. If this bit is set, the $\overline{DACK}$ pin will be inactive during chaining, during both Flowthru Transfers and Flowthru Transfer-and-Searches and during Searches, but $\overline{DACK}$ will be pulsed active during Flyby Transfers and Flyby Transfers-and-Searches at the time necessary to strobe data into or out of the Flyby peripheral. Flyby operations are discussed in detail in the "Flyby Transactions" section.

Chain Address Register

Each channel has a Chain Address register which points to the chain control table in memory containing data to be loaded into the channel's registers. The Chain Address register, as shown in Figure 20 is two words long. The first word consists of the Segment and Tag fields. The second word contains the 16-bit offset portion of the memory address. The highest bit in the segment field is not used when the DTC is configured for Logical Address space (MM1 = 1). The Tag field contains 2 bits used to designate the number of wait states to be inserted during accesses to the Chain Control Table.

The Chain Address register may be loaded during chaining and may be read from and written to by the host CPU without wait states. If an $\overline{EOP}$ is issued to the DTC during chaining, the Chain Address register holds the old address. This is true even if the access failure occurred while new Chain Address data was being loaded, since the old data is restored unless both words of the new data are successfully read. Note, however, that $\overline{EOP}$s that occur when chaining and while loading a new Chain Address cause the new data to be lost.

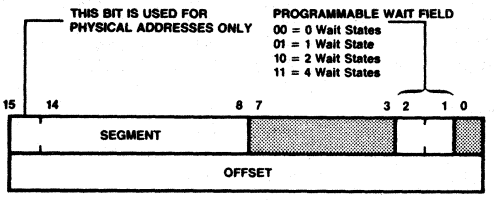


Figure 20. Chain Address Register

APPLICATIONS INFORMATION

Figure 21 shows the configuration of the AmZ8016 DMA Transfer Controller (DTC) and a microprocessor system. The DTC issues a $\overline{BUSRQ}$ active low signal to the CPU to request bus control. When the CPU replies with a Bus Acknowledge $\overline{BUSAK}$ signal through the $\overline{BAI}$ $\overline{BAO}$ daisy chain to the DTC which issued the $\overline{BUSRQ}$, the DTC takes control of the Address-Data bus and the Control bus. In addition to hardware reset the 8016 can be given a software 'reset' command (i.e., loading all zero to Command Register). Two DMA channels are provided per each DTC device. The logic blocks A and B are shown in Figure 22 and 23.

Figure 22 shows the bus request logic used for bidirectionally buffering $\overline{BUSRQ}$. See 'Bus Request/Grant' section for detail. Figure 23 shows the $\overline{CS}/\overline{WAIT}$ logic for the multiplexed $\overline{CS}/\overline{WAIT}$ pin of DTC. See 'Wait States' for detailed description.

AmZ8016 DTC to AmZ8010 MMU Interface

The AmZ8010 Memory Management Unit (MMU) is a high performance LSI product that adds sophisticated address translation and memory protection capabilities to AmZ8001 CPU systems. The MMU contains table of access attributes that are individually programmable for each segment. Attributes provided are read-only, system-mode-only, DMA only, execute only, and CPU only. If the MMU detects a memory access that violates one of the attributes of a segment, the MMU interrupts the CPU or DMA to inhibit an erroneous memory write.

Figure 24 shows the AmZ8016 DTC to AmZ8010 MMU Interface Configuration. The MMUSYNC output of DTC ORed with the $\overline{BUSAK}$ signal of CPU is connected to the DMASYNC input of MMU. The MMUSYNC pin of DTC is multiplexed with SN7. If the Master Mode register bit MM1 is set (i.e., in Logical Addressing mode), this pin outputs MMUSYNC active high pulse prior to each DMA cycle when the DTC is in control of the system bus or outputs low level when the DTC is not in control of the system bus. If the MM1 is clear (i.e., in Physical Addressing mode), this pin outputs the SN7 signal when DTC is a bus master or is driven high impedance off when DTC is not on control of the system bus.

The $\overline{SUP}$ output of MMU is connected to the $\overline{EOP}$ pin of the DTC for terminating the DMA operation whenever a violation has been detected.

Refer to page 7-1 for Essential Information on Military Devices

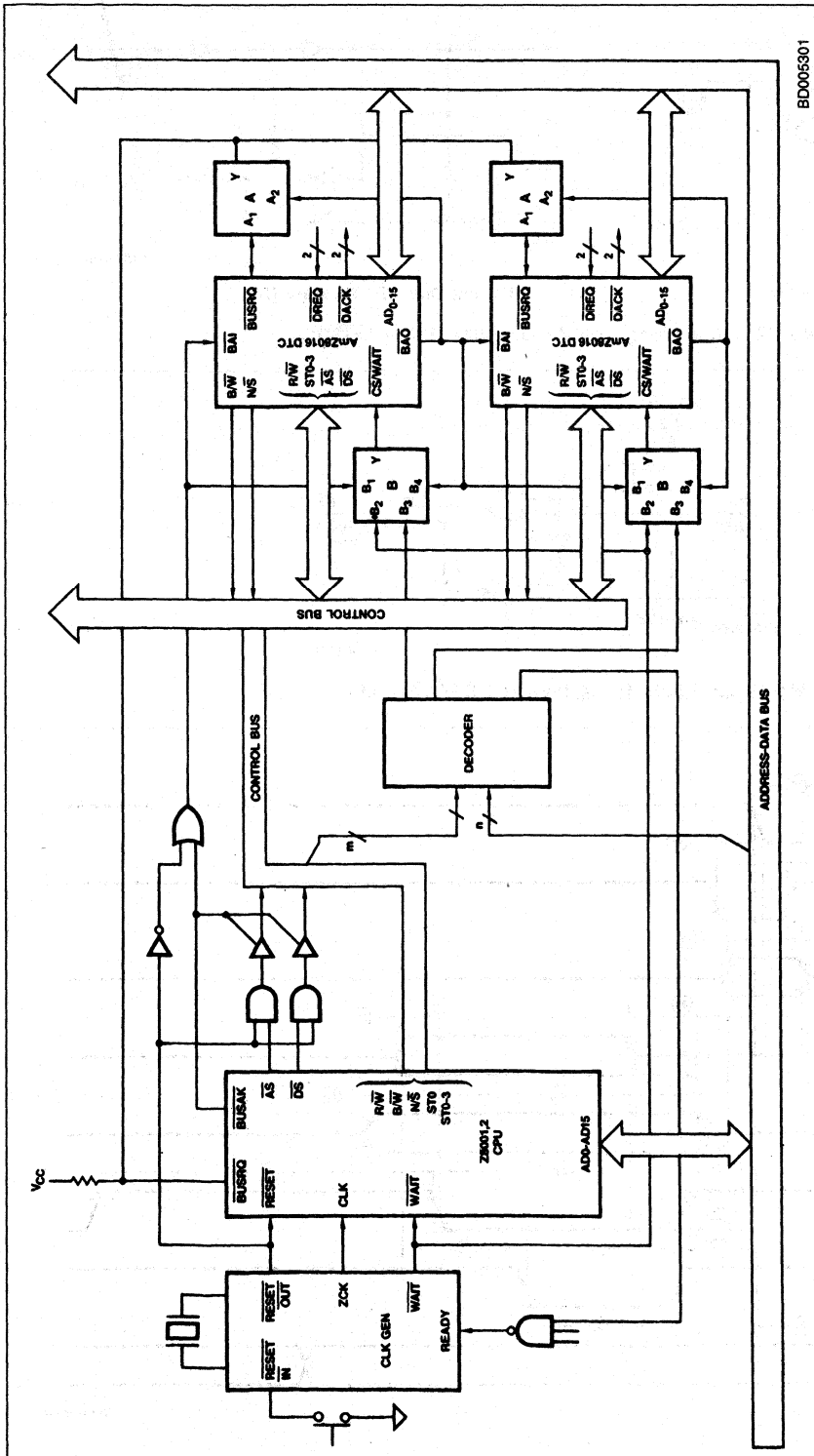
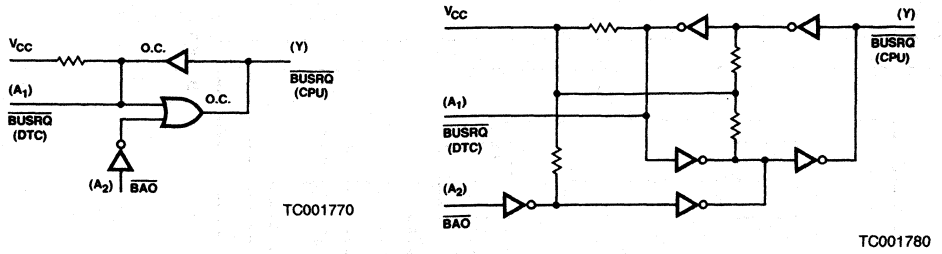


Figure 21. Basic DTC Configuration



a) Using Three Different Gates

b) Using One TTL Package (7405)

Figure 22. Logic Used to Bidirectionally Buffer $\overline{BUSRQ}$

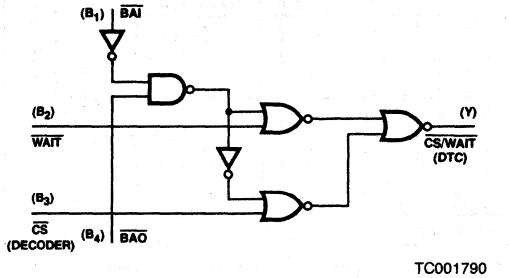
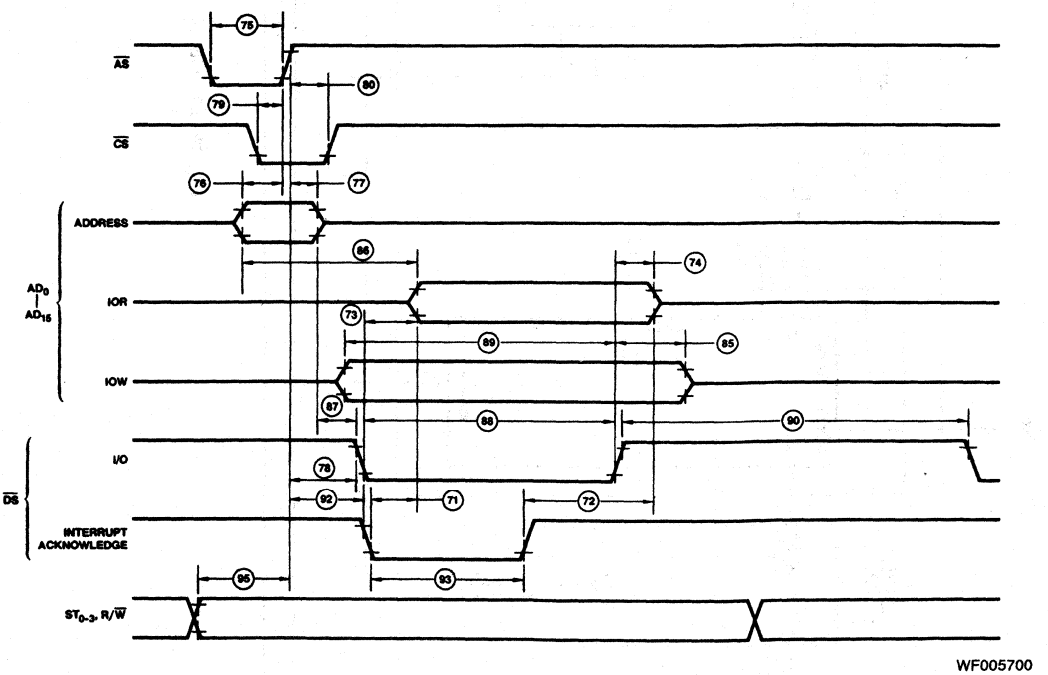


Figure 23. Logic for Multiplexed $\overline{CS}/\overline{WAIT}$ pin



Timing Diagram 8. AC Timing when DTC is Bus Slave

Refer to page 7-1 for Essential Information on Military Devices

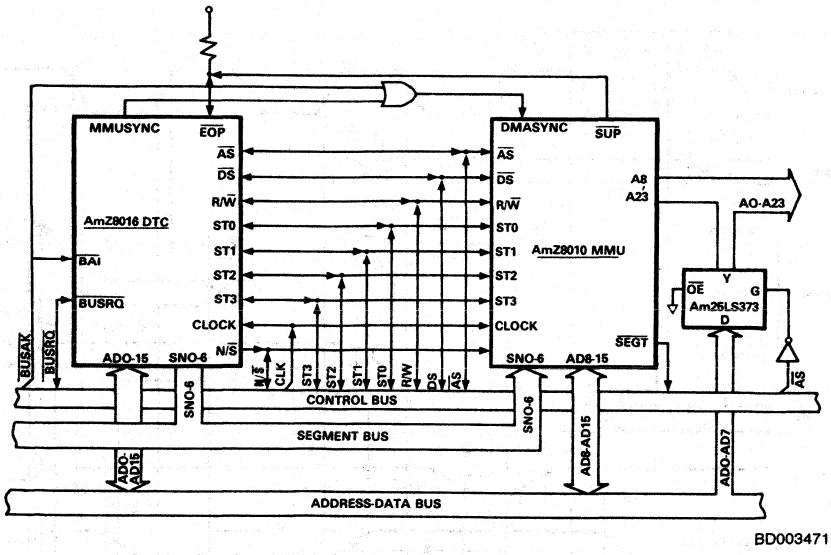
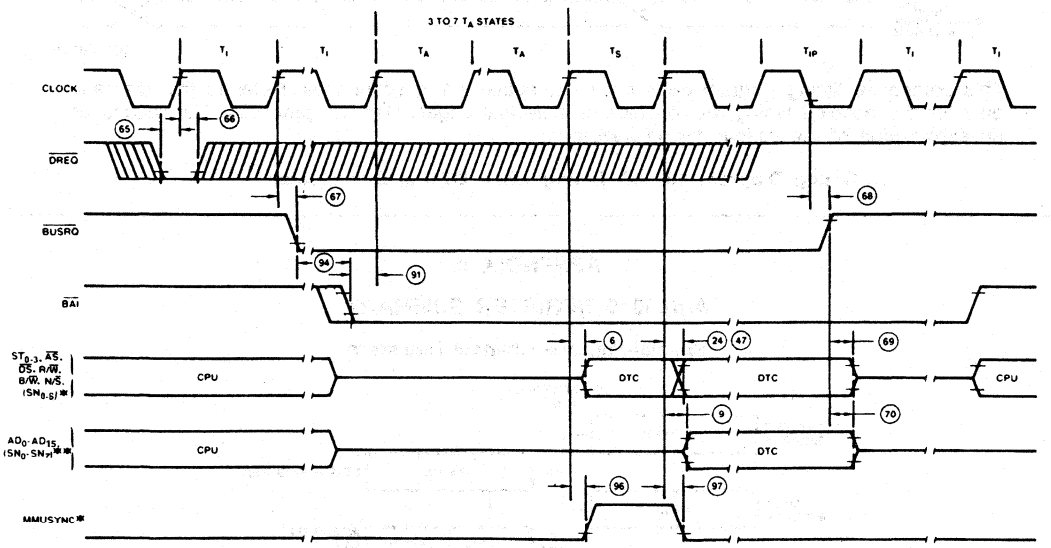


Figure 24. AmZ8016 DTC to AmZ8010 MMU Interface Configuration

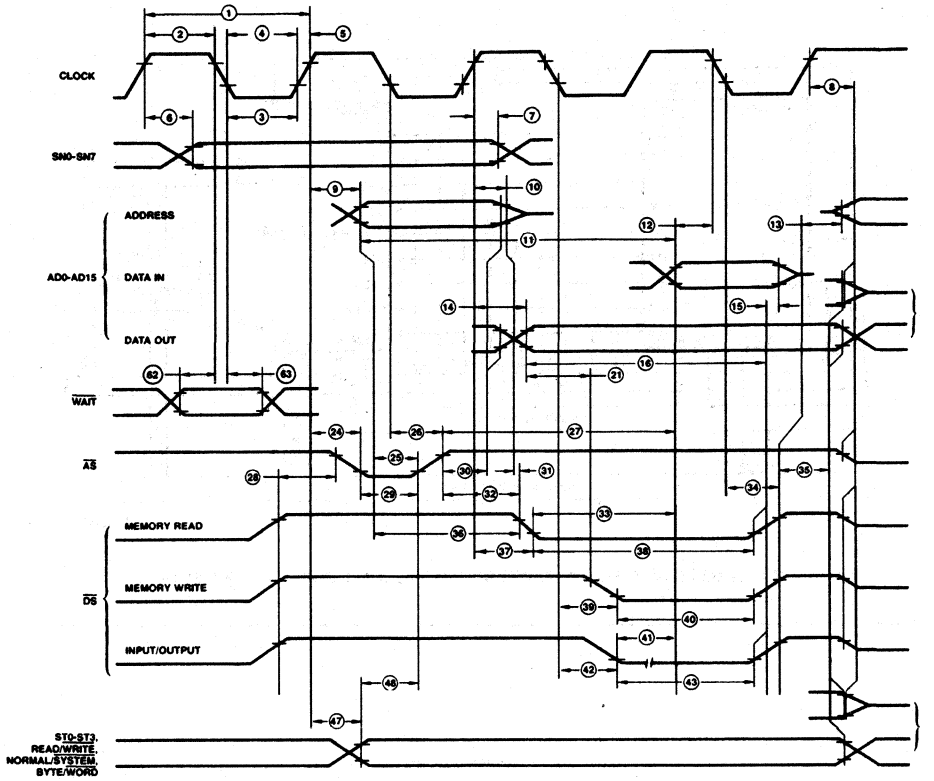


*For logical addressing only

**For physical addressing only

Timing Diagram 9. Bus Exchange Timing

Refer to page 7-1 for Essential Information on Military Devices



WF005720

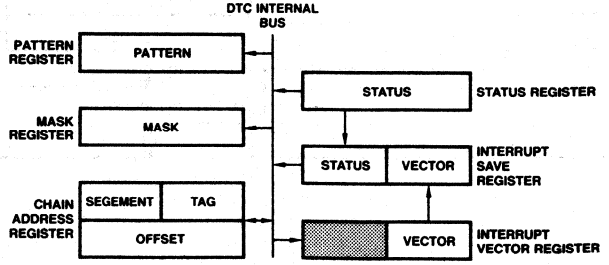
This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Timing Diagram 10. AC Timing when DTC is Bus Master

APPENDIX A

AmZ8016 REGISTER SUMMARY

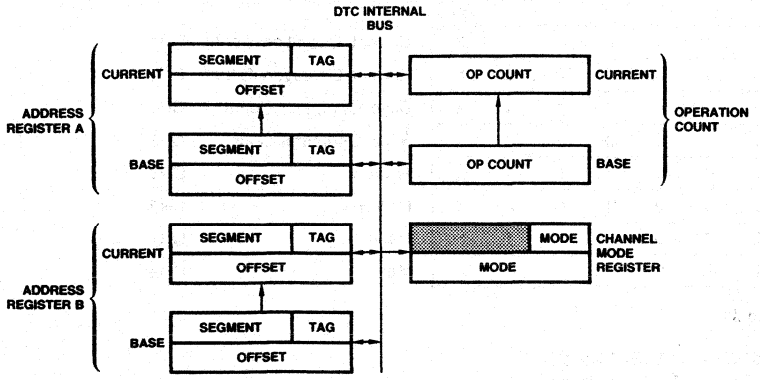
Special-Purpose Channel Registers



DF002340

Refer to page 7-1 for Essential Information on Military Devices

General-Purpose Channel Registers



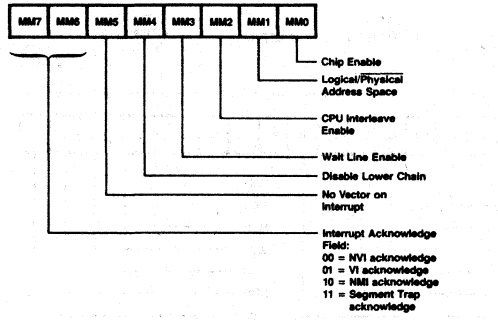
DF002350

Master Mode Register

Address

A7							A0
X	0	1	1	1	0	0	X

Fast Readable
Writable



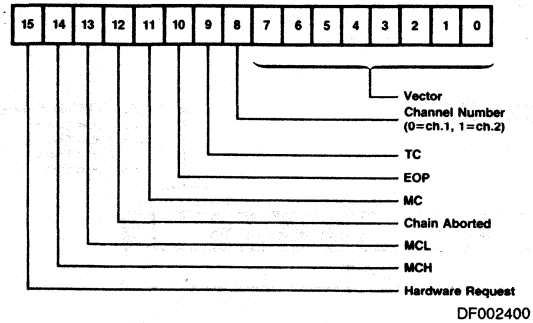
DF002390

Interrupt Save Register

Address

A7							A0	
X	0	1	0	1	0	1	X	CH1
X	0	1	0	1	0	0	X	CH2

Fast Readable



DF002400

Miscellaneous Registers

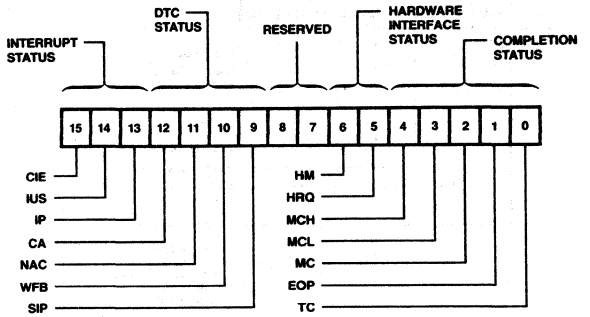
	A7							A0		
Address	X	0	1	1	0	0	1	X	Current Operation Count	CH1
	X	0	1	1	0	0	0	X	Current Operation Count	CH2
	X	0	1	1	0	1	1	X	Base Operation Count	CH1
	X	0	1	1	0	1	0	X	Base Operation Count	CH2
	X	1	0	0	1	0	1	X	Pattern	CH1
	X	1	0	0	1	0	0	X	Pattern	CH2
	X	1	0	0	1	1	1	X	Mask	CH1
	X	1	0	0	1	1	0	X	Mask	CH2

Chain Loadable
Writable
Pattern and Mask — Slow Readable
Operation Count — Fast Readable

Status Register

	A7							A0	
Address	X	0	1	0	1	1	1	X	CH1
	X	0	1	0	1	1	0	X	CH2

Fast Readable

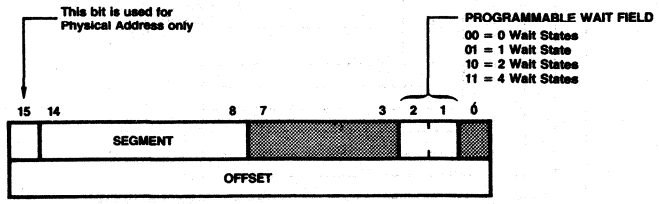


DF002590

Chain Address Register

	A7							A0		
Address	X	0	1	0	0	1	1	X	Segment/Tag	CH1
	X	0	1	0	0	1	0	X	Segment/Tag	CH2
	X	0	1	0	0	0	1	X	Offset	CH1
	X	0	1	0	0	0	0	X	Offset	CH2

Fast Readable/Writable
Chain Loadable



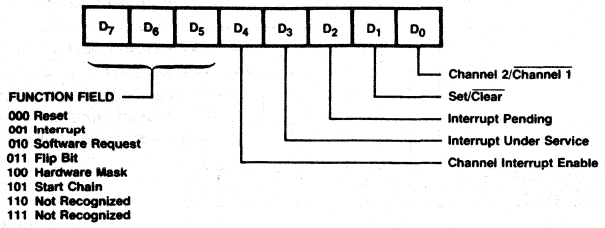
DF002600

Refer to page 7-1 for Essential Information on Military Devices

Command Register

	A7						A0
Address	X	0	1	0	1	1	X	X

Writable Only

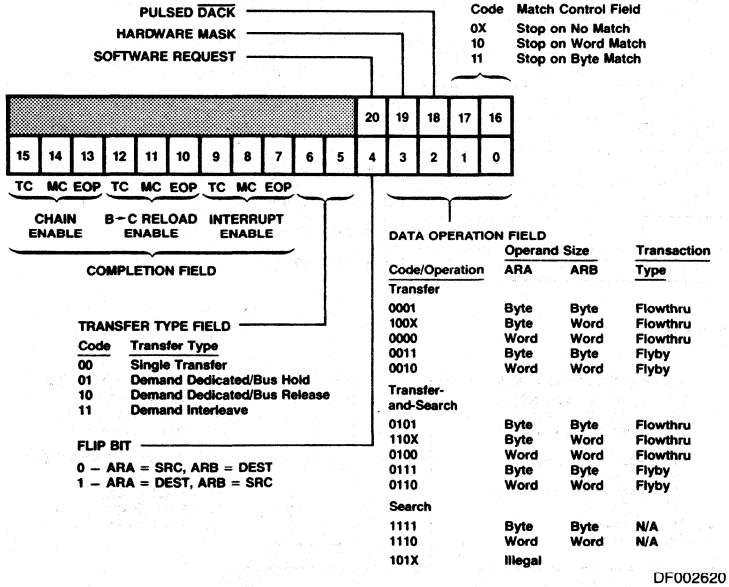


DF002610

Channel Mode Register

A7							A0	
X	1	0	1	0	1	1	X	High CH1
X	1	0	1	0	1	0	X	High CH2
X	1	0	1	0	0	1	X	Low CH1
X	1	0	1	0	0	0	X	Low CH2

Chain Loadable
Writable (Lower 16 bits)
Slow Readable

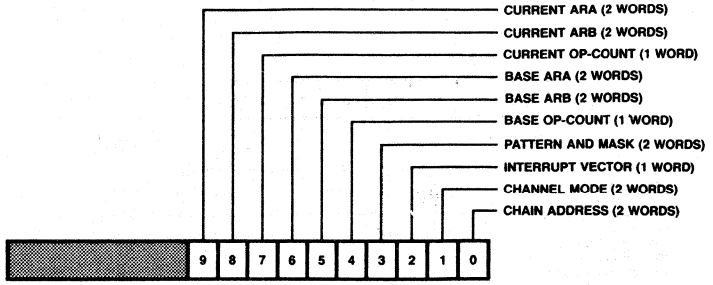


DF002620

Refer to page 7-1 for Essential Information on Military Devices

Chain Control Register

Chain Loadable Only

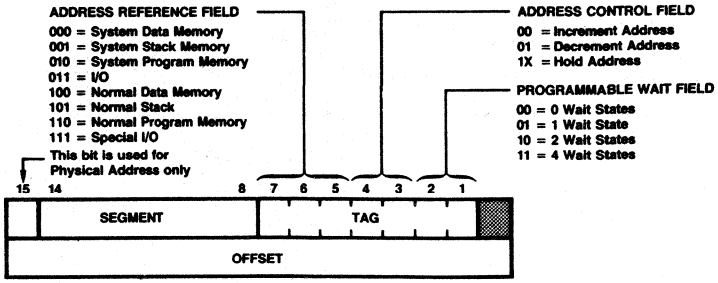


DF002630

Address Register

	A7							A0		
Address	X	0	0	1	1	0	1	X	Current ARA Segment/Tag	CH1
	X	0	0	1	1	0	0	X	Current ARA Segment/Tag	CH2
	X	0	0	0	1	0	1	X	Current ARA Offset	CH1
	X	0	0	0	1	0	0	X	Current ARA Offset	CH2
	X	0	0	1	0	0	1	X	Current ARB Segment/Tag	CH1
	X	0	0	1	0	0	0	X	Current ARB Segment/Tag	CH2
	X	0	0	0	0	0	1	X	Current ARB Offset	CH1
	X	0	0	0	0	0	0	X	Current ARB Offset	CH2
	X	0	0	1	1	1	1	X	Base ARA Segment/Tag	CH1
	X	0	0	1	1	1	0	X	Base ARA Segment/Tag	CH2
	X	0	0	0	1	1	1	X	Base ARA Offset	CH1
	X	0	0	0	1	1	0	X	Base ARA Offset	CH2
	X	0	0	1	0	1	1	X	Base ARB Segment/Tag	CH1
	X	0	0	1	0	1	0	X	Base ARB Segment/Tag	CH2
	X	0	0	0	0	1	1	X	Base ARB Offset	CH1
	X	0	0	0	0	1	0	X	Base ARB Offset	CH2

Chain Loadable
Fast Readable and Writable



DF002640

Refer to page 7-1 for Essential Information on Military Devices

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages with
Respect to V_{SS} ..... −0.5 to +7.0V
Power Dissipation (Package Limitation) 2W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	T_A	V_{CC}
Commercial	0°C to 70°C	5V ±5%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

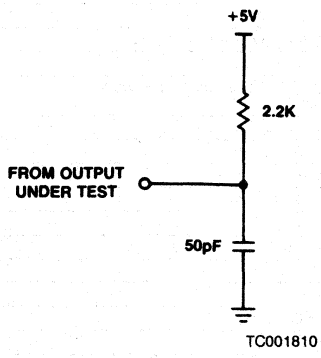
Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	V_{CC}−0.4	V_{CC}+0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.4	V_{CC}+0.3	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		Volts
V_{OL}	Output Low Voltage	I_{OL} = +2.0mA		0.4	Volts
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$		±10	μA
I_{OL}	Output Leakage	$V_{SS} \leq V_{OUT} \leq V_{CC}$		±10	μA
I_{CC}	V_{CC} Supply Current	T_A = 0°C		350	mA
		T_A = 75°C		200	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.		10	pF
C_{OUT}	Output Capacitance			15	pF
$C_{I/O}$	Bidirectional Capacitance			20	pF

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

+4.75V ≤ V_{CC} ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ +70°C

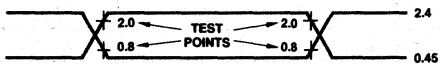
Open-Drain Test Load



Refer to page 7-1 for Essential Information on Military Devices

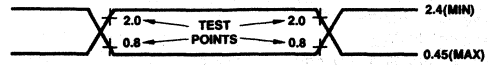
TIMING REFERENCES FOR AC TESTS

Input Waveform



WF005730

Output Waveform



WF005740

All AC parameters assume a load capacitance of 100pF max, except for parameter 6 TdC(SNv)(50pF max).

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

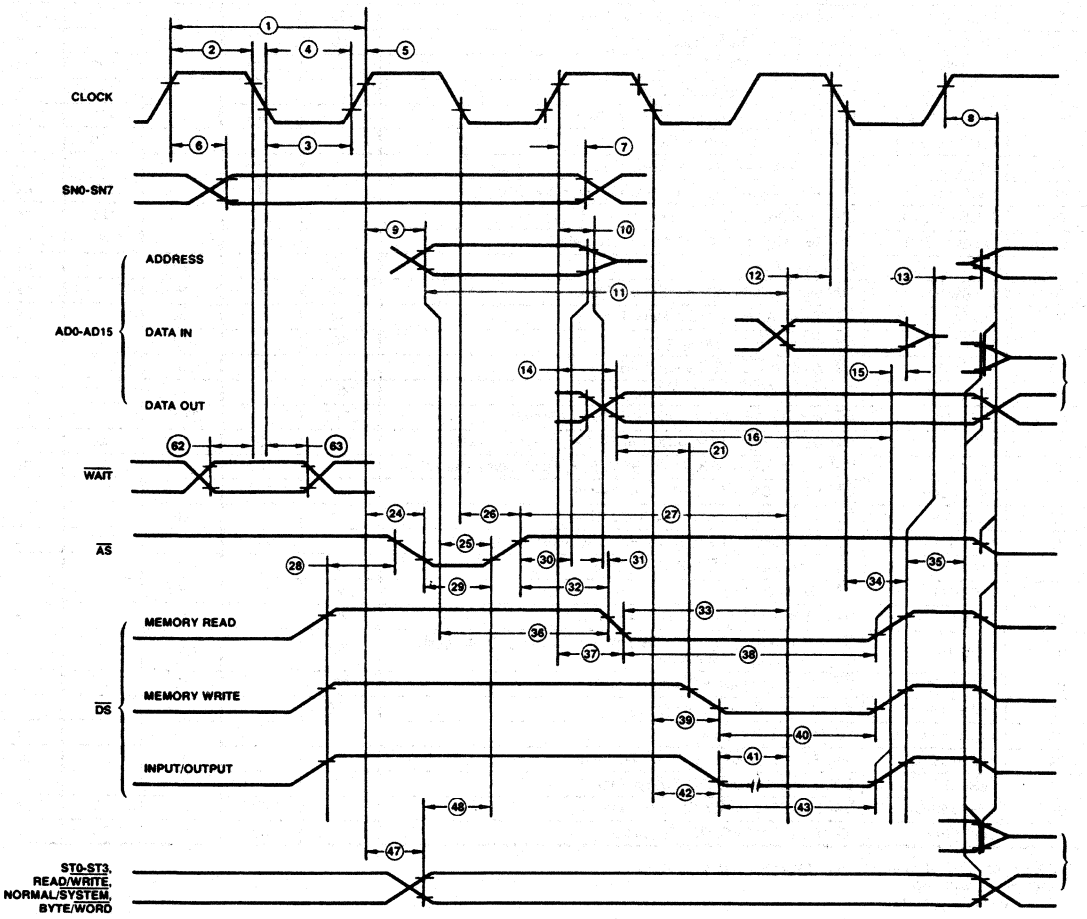
TIMING FOR DTC AS BUS MASTER

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TcC	Clock Cycle Time	250	2000	165		ns
2	TwCh	Clock Width (High)	105		70		ns
3	TwCl	Clock Width (LOW)	105		70		ns
4	TfC	Clock Fall Time		20		10	ns
5	TrC	Clock Rise Time		20		15	ns
6	TdC(SNv)	Clock RE to Segment Number Valid (50pF Load) Delay***		110		90	ns
7	TdC(SNn)	Clock RE to Segment Number Valid Delay		90		60	ns
8	TdC(Bz)	Clock RE to Bus Float Delay		65		50	ns
9	TdC(A)	Clock RE to Address Valid Delay		100		90	ns
10	TdC(Az)	Clock RE to Address Float Delay		65		60	ns
11	TdA(Dl)	Address Valid to Data In Required Valid Delay	400		305		ns
12	TsDl(C)	Data In to Clock FE Set-up Time	20		15		ns
13	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		ns
14	TdC(DO)	Clock RE to Data Out Valid Delay		100		90	ns
15	ThDl(DS)	$\overline{DS}$ RE to Data In Hold Time	0		0		ns
16	TdDO(DS)	Data Out Valid to $\overline{DS}$ RE Delay	230		200		ns
21	TdDO(SW)	Data Out Valid to $\overline{DS}$ FE (Write) Delay	55		35		ns
24	TdC(ASf)	Clock RE to $\overline{AS}$ FE Delay		70		60	ns
25	TdA(AS)	Address Valid to $\overline{AS}$ RE Delay	50		35		ns
26	TdC(ASr)	Clock FE to $\overline{AS}$ RE Delay		80		60	ns
27	TdAS(Dl)	$\overline{AS}$ RE to Data In Required Valid Delay	300			220	ns
28	TdDS(AS)	$\overline{DS}$ RE to $\overline{AS}$ FE Delay	75		35		ns
29	TwAS	$\overline{AS}$ Width (LOW)	80		60		ns
30	TdAS(A)	$\overline{AS}$ RE to Address Valid Delay	60		45		ns
31	TdAz(DSR)	Address Float $\overline{DS}$ (Read) FE Delay	0		0		ns
32	TdAS(DSR)	$\overline{AS}$ RE to $\overline{DS}$ FE (Read) Delay	75		40		ns
33	TdDSR(Dl)	$\overline{DS}$ (Read) FE to Data In Required Valid Delay	165		155		ns
34	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		70		65	ns
35	TdDS(DO)	$\overline{DS}$ RE to Data Out (Write Only) and Status Valid (Read and Write) Delay	85		45		ns
36	TdA(DSR)	Address Valid to $\overline{DS}$ (Read) FE Delay	120		110		ns
37	TdC(DSR)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60	ns
38	TwDSR	$\overline{DS}$ (Read) Width (LOW)	275		185		ns
39	TdC(DSW)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60	ns
40	TwDSW	$\overline{DS}$ (Write) Width (LOW)	160		150		ns
41	TdDSl(Dl)	$\overline{DS}$ (Input) FE to Data In Required Valid Delay	325			210	ns
42	TdC(DSf)	Clock FE to $\overline{DS}$ (I/O) FE Delay		60		60	ns
43	TwDS	$\overline{DS}$ (I/O) Width (LOW)	150*		150		ns
47	TdC(S)	Clock RE to Status Valid Delay		110		80	ns
48	TdS(AS)	Status Valid to $\overline{AS}$ RE Delay	60		35		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock FE Set-up Time	20		20		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	30		30		ns
96	TdC(SNr)	Clock RE to SN7/MMUSync RE Delay**		110		110	ns
97	TdC(SNf)	Clock RE to SN7/MMUSync FE Delay**	20	110		110	ns

*Wait states should be inserted by programming a hardware when accessing slow peripherals.
**Logical Addressing only.
***130ns max with Logical Addressing.
Note: RE = rising edge FE = falling edge

Refer to page 7-1 for Essential Information on Military Devices

Timing Diagram 11. AC Timing when DTC is Bus Master



WF005780

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Refer to page 7-1 for Essential Information on Military Devices

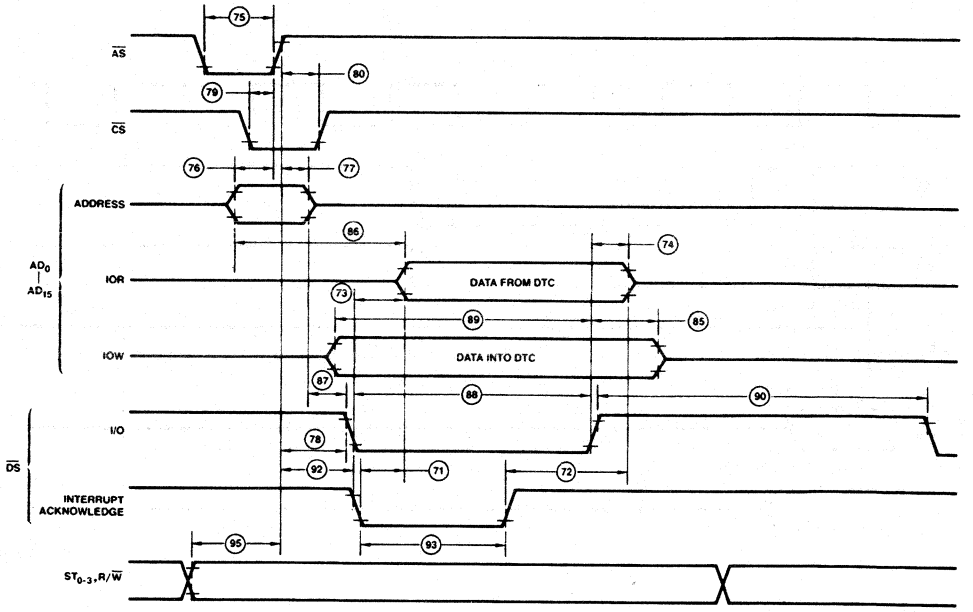
SWITCHING CHARACTERISTICS over operating range unless otherwise specified
TIMING FOR DTC AS BUS SLAVE AND CPU-DTC BUS EXCHANGE

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
64	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		ns
65	TsDRQ(c)	$\overline{DREQ}$ Valid to Clock RE Set-up Time	60		50		ns
66	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		ns
67	TdC(BRQf)	Clock RE to $\overline{BUSRQ}$ FE Delay		150		120	ns
68	TdC(BRQr)	Clock FE to $\overline{BUSRQ}$ RE Delay		165		150	ns
69	TdBRQ(BUSc)	$\overline{BUSRQ}$ RE to Control Bus Float Delay		140		110	ns
70	TdBRQ(BUSd)	$\overline{BUSRQ}$ RE to AD Bus Float Delay		140		110	ns
71	TdDSA(RDV)	$\overline{DS}$ FE (Acknowledge) to Data Output Valid Delay		135		120	ns
72	TdDSA(RDZ)	$\overline{DS}$ RE (Acknowledge) to Data Output Float Delay		80		75	ns
73	TdDSR(DOD)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135		120	ns
74	TdDSR(RDZ)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		75	ns
75	TwAS	$\overline{AS}$ Low Width	70		50		ns
76	TsA(AS)	Address Valid to $\overline{AS}$ RE Set-up Time	30		10		ns
77	ThAS(Av)	$\overline{AS}$ RE to Address Valid Hold Time	50		40		ns
78	TdAS(DS)	$\overline{AS}$ RE to $\overline{DS}$ FE Delay (I/O)	50		40		ns
79	TsCS(AS)	$\overline{CS}$ Valid to $\overline{AS}$ RE Set-up Time	0		0		ns
80	ThCS(AS)	$\overline{AS}$ RE to $\overline{CS}$ Valid Hold Time	40		30		ns
81	TwAS(DS)	$\overline{AS}$ and $\overline{DS}$ Simultaneously LOW Time (Reset)	3TcC		3TcC		ns
82	TdBAI(Az)	$\overline{BAI}$ RE to SN0 – SN7, AD0 – AD15 Float Delay (Reset)		135		120	ns
83	TdBAI(ST)	$\overline{BAI}$ RE to ST0 – ST3, R/$\overline{W}$, B/$\overline{W}$, N/$\overline{S}$ Float Delay (Reset)		100		80	ns
84	TdBAI(DS)	$\overline{BAI}$ RE to $\overline{DS}$, $\overline{AS}$ Float Delay (Reset)		100		85	ns
85	TdDS(Dn)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		40		ns
86	TdAC(DRV)	Address Valid to Data (IOR) Required Valid Delay		540		345	ns
87	TdAZ(DS)	Address Float to $\overline{DS}$ FE (IOR) Delay	0		0		ns
88	TwDS(IO)	$\overline{DS}$ (IO) LOW Width	150*		150		ns
89	TsD(DS)	Data (IOW) Valid to $\overline{DS}$ RE Set-up Time	40		40		ns
90	TrDS(W)	$\overline{DS}$ RE (IOW) to $\overline{DS}$ FE (IOW) (Write Recovery Time applies only for issuing Command)	4TcC		4TcC		ns
91	TsBAK(C)	$\overline{BAI}$ Valid to Clock RE Set-up Time	60		50		ns
92	TdAS(DS)	$\overline{AS}$ RE to $\overline{DS}$ FE (ACK) Delay	100		100		ns
93	TwDS(AK)	$\overline{DS}$ (ACK) LOW Width	150		150		ns
94	TdBRQ(BAI)	$\overline{BUSRQ}$ FE to $\overline{BAI}$ FE Required Delay	0		0		ns
95	TsS(AS)	Status Valid to $\overline{AS}$ RE Set-up Time	40		0		ns
98	TdBAI(BAO)	$\overline{BAI}$ RE, FE to $\overline{BAO}$ RE, FE Delay		80		70	ns
99	TdIEI(IEO)	IEI RE, FE to IEO RE, FE Delay		80		60	ns

*2000ns for reading slow-readable registers (worst case).
Note: RE = rising edge
FE = falling edge

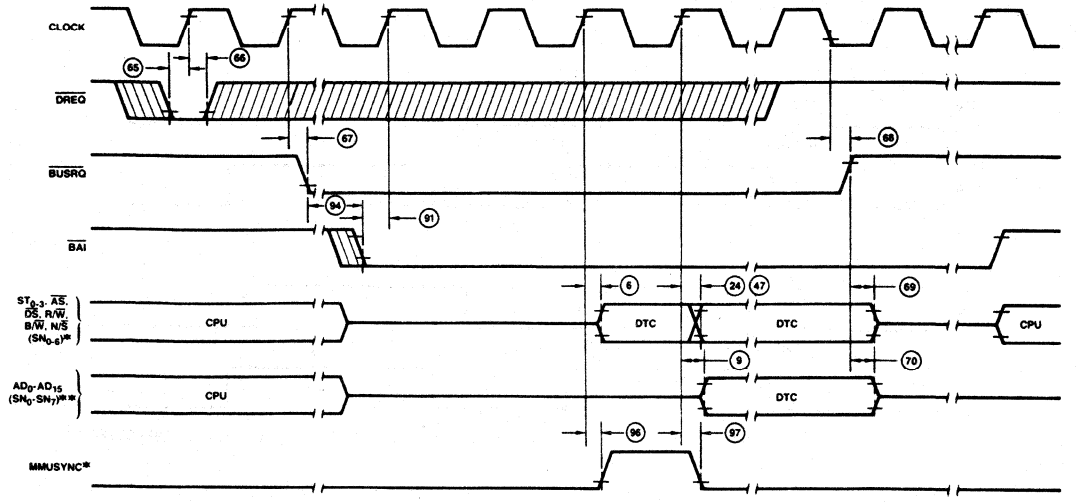
Refer to page 7-1 for Essential Information on Military Devices

Timing Diagram 12. AC Timing when DTC is Bus Slave



WF005790

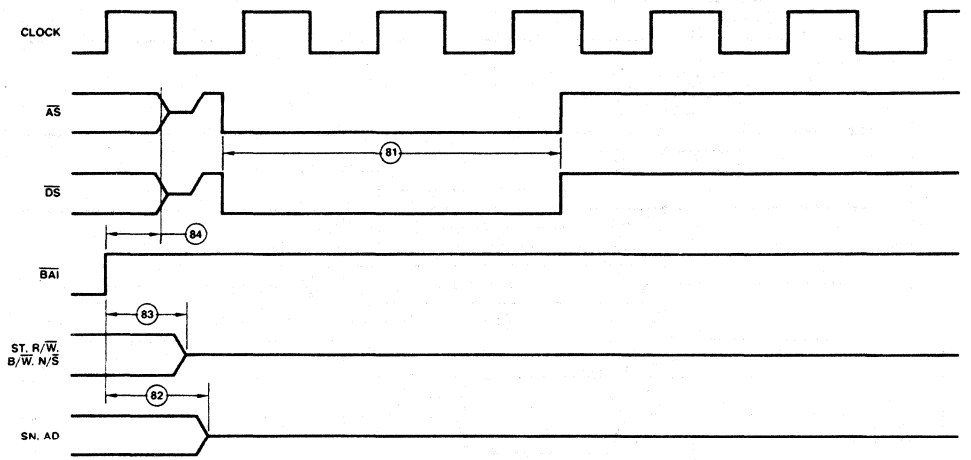
Timing Diagram 13. Bus Exchange Timing



WF005800

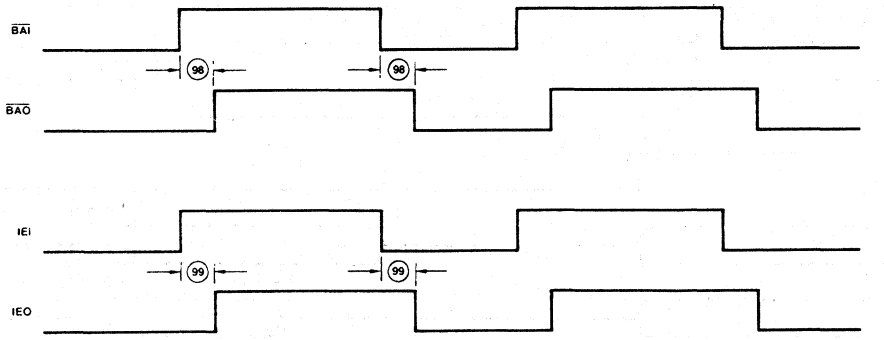
*For logical addressing only
**For physical addressing only

Timing Diagram 14. Reset Timing



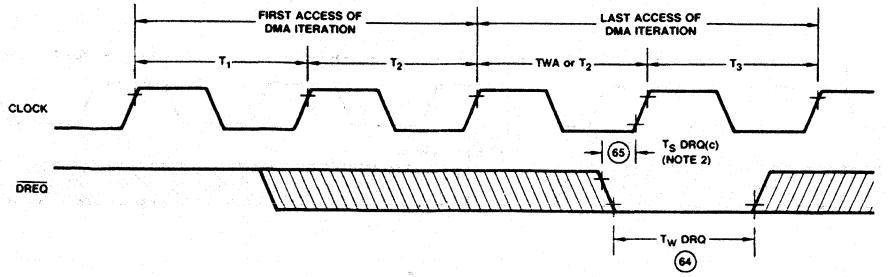
WF005810

Timing Diagram 15. Delay Timings



WF005820

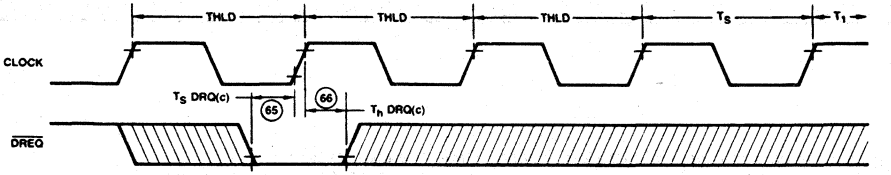
Timing Diagram 16. Sampling $\overline{DREQ}$ during Single Transfer DMA Operations



WF005830

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of the first access of the DMA iteration.
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur T_SDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before T_SDRQ(c) if it has met the T_WDRQ parameter.
3. Flyby and Search transactions have only a single access; parameter T_SDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

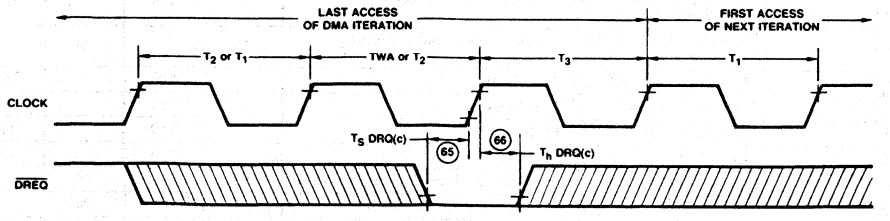
Timing Diagram 17a). Sampling $\overline{DREQ}$ at the End of Chaining



WF005840

Notes: 1. $\overline{DREQ}$ must be LOW from the start of T_SDRQ (c) to the end of T_hDRQ (c) to ensure that the request is recognized.
2. Failure to meet this setup time will result in the channel releasing the bus.

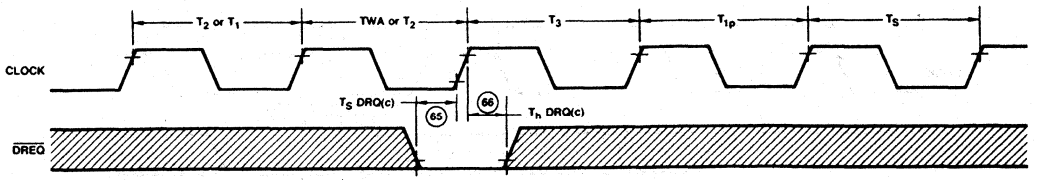
Timing Diagram 17b). Sampling of $\overline{DREQ}$ While in Bus Hold Mode



WF005850

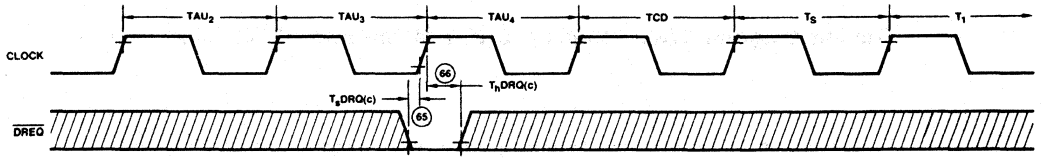
Refer to page 7-1 for Essential Information on Military Devices

Timing Diagram 17c). $\overline{DREQ}$ Sampling in Demand Mode during DMA Operations



WF005860

Timing Diagram 17d). Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading



WF005870

Notes: 1. T_S is a set-up state, generated before entering DMA operation cycle.
2. TAU_2 through TAU_4 are auto-reloading states, followed by TCD (chain decision) state.

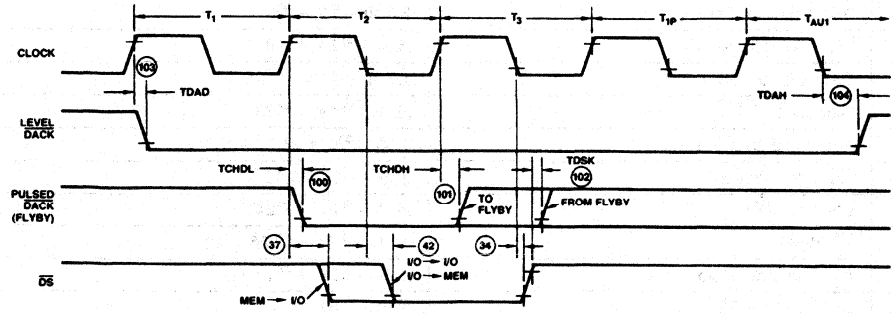
SWITCHING CHARACTERISTICS over operating range unless otherwise specified
TIMING FOR DTC-PERIPHERAL INTERFACE

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
100	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		100	ns
101	TCHDH	Clock FE to Pulsed $\overline{DACK}$ RE Delay (Transactions TO Flyby Peripheral Only)		100		100	ns
102	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (Transactions FROM Flyby Peripheral Only)	20		20		ns
103	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		100	ns
104	TDAH	Clock FE to Level $\overline{DACK}$ Valid Delay		110		110	ns
105	TEIDL	Clock FE to Internal $\overline{EOP}$ FE Delay		110		100	ns
106	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		100	ns
107	TES	External $\overline{EOP}$ Valid to Clock FE Set-up Time During Operation	10		10		ns
108	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		ns
109	TES(BH)	External $\overline{EOP}$ Valid to Clock RE Set-up Time During Bus Hold	10		10		ns
110	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		ns

Note: RE = rising edge FE = falling edge

Refer to page 7-1 for Essential Information on Military Devices

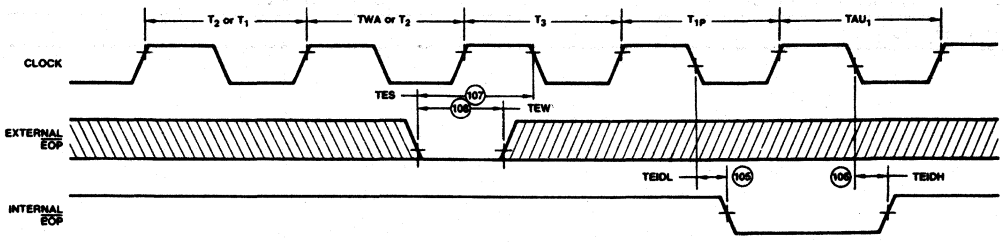
Timing Diagram 18. $\overline{DACK}$ Timing



WF005750

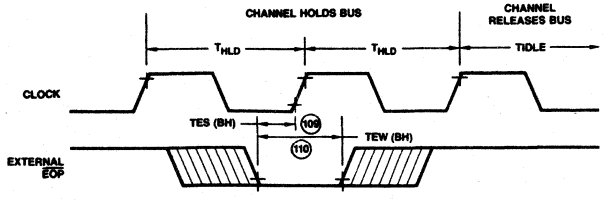
*LEVEL $\overline{DACK}$ RE occurs as shown if auto-reloading is not programmed. LEVEL $\overline{DACK}$ stays LOW for three additional clocks for reloading.

Timing Diagram 19a). $\overline{EOP}$ Sampling and Generation during DMA Operations



WF005760

Timing Diagram 19b). Sampling of $\overline{EOP}$ during Bus Hold



WF005770

Notes: 1. The diagram lists state names of both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_1P is a pseudo-T_1 state, without an active $\overline{AS}$ generated, following termination of any DMA operation.
3. State TAU_1 is an auto-initialization state, generated following the TC, MC or EOP termination.

Refer to page 7-1 for Essential Information on Military Devices

AmZ8016 DMA Derived Timings

Parameters	Description	Derivation
11	Adr. Valid to Data In	2.5 Φ – #9 – #12
13	$\overline{DS}$ ↑ to Adr. Active	.5 Φ – #34 + #9
16	Data Out to $\overline{DS}$ ↑	1.5 Φ – #14 + #34
21	Data Out to $\overline{DS}$ ↓	.5 Φ – #14 + #39
25	Adr. Valid to $\overline{AS}$ ↑	.5 Φ – #9 + (#26 – tr)
27	$\overline{AS}$ ↑ to Data In	2 Φ – #26 – #12
28	$\overline{DS}$ ↑ to $\overline{AS}$↓	.5 Φ – #34 + #24
29	$\overline{AS}$ Width	.5 Φ – #24 + (#26 – tr)
30	$\overline{AS}$ ↑ to Adr. Hold	.5 Φ – #26 + #10
31	Adr. float to $\overline{DS}$ ↓	#10 – #37
32	$\overline{AS}$↑ to $\overline{DS}$ ↓	.5 Φ – #26 + #37
33	$\overline{DS}$ ↓ to Data In (Read)	1.5 Φ – #37 – #12
35	$\overline{DS}$ ↑ to Data Out Hold	.5 Φ – #34 + #8
36	Adr. Valid to $\overline{DS}$ ↓	1 Φ – #9 + #37
38	$\overline{DS}$ Width (Read)	1.5 Φ – #37 + #34
40	$\overline{DS}$ Width (Write)	1 Φ – #39 + #34
41	$\overline{DS}$ ↓ to Data In (Input)	(1 + Wait) Φ – #42 – #12
43	$\overline{DS}$ Width (I/O)	1 Φ – #42 + #34
48	Status Valid to $\overline{AS}$ ↑	.5 Φ – #47 + (#26 – tr)

tr (nominal) = 10μS

Z8030/Z8530 (SCC)

Serial Communications Controller

DISTINCTIVE CHARACTERISTICS

- **Two 1M.bps full duplex serial channels**
  Each channel has independent oscillator, baud-rate generator, and PLL for clock recovery, dramatically reducing external components.
- **Programmable protocols**
  NRZ, NRZI, and FM data encoding supported under program control.
- **Programmable Asynchronous Modes**
  5- to 8-bit characters with programmable stop bits, clock, break detect, and error conditions.
- **Programmable Synchronous Modes**
  SDLC and HDLC and SDLC loop supported with frame control, zero insertion and deletion, abort, and residue handling. CRC-16 and CCITT generators and checkers.
- **Z8000* compatible**
  The Z8030 interfaces directly with the Z8000 CPU bus and to the Z8000 interrupt structure.
- **Compatible with non-multiplexed bus**
  The Z8530 interfaces easily to most other CPUs.

GENERAL DESCRIPTION

The SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with 8- and 16-bit microprocessors. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops, and crystal oscillators, which dramatically reduce the need for external logic.

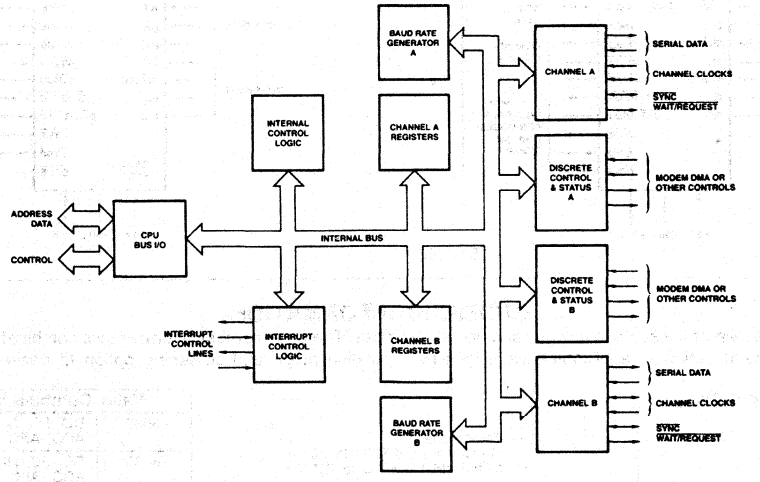
The SCC handles asynchronous formats, synchronous byte-oriented protocols, such as IBM Bisync, and synchronous bit-oriented protocols, such as HDLC and IBM SDLC.

This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drivers, etc.).

The device can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The SCC is offered in two versions. The Z8030 is directly compatible with the Z8000 and 8086 CPUs. The Z8530 is designed for non-multiplexed buses and is easily interfaced with most other CPUs, such as 8080, Z80, 6800, 68000, and †Multibus.

BLOCK DIAGRAM



BD003520

Figure 1.

RELATED PRODUCTS

See the Z8030/Z8530 Serial Communications Controller Technical Manual – 1983 edition (A1Z-2135) for detailed technical information.

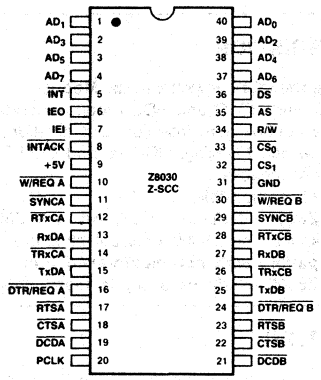
*Z8000, Z8030 and Z8530 are trademarks of Zilog, Inc. †MULTIBUS is a trademark of Intel.

00970D

Refer to page 7-1 for Essential Information on Military Devices

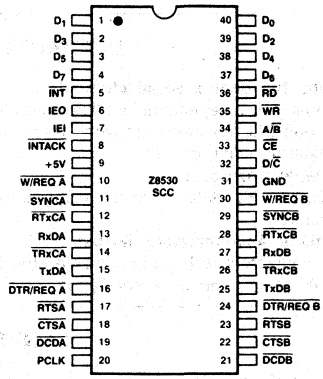
CONNECTION DIAGRAM
Top View

D-40
P-40



CD005350

D-40
P-40

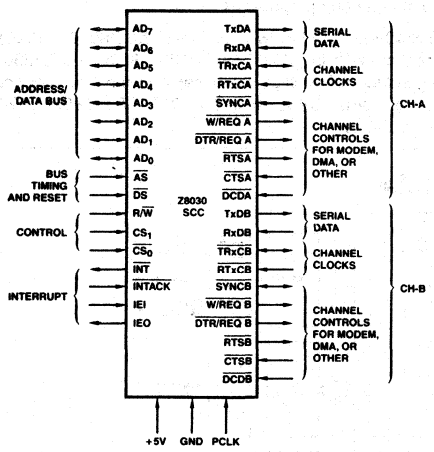


CD005360

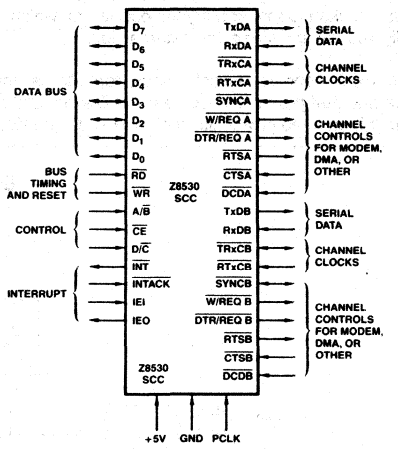
Also available in PLCC. See Section 7 for pinout details.

Note: Pin 1 is marked for orientation

LOGIC SYMBOL



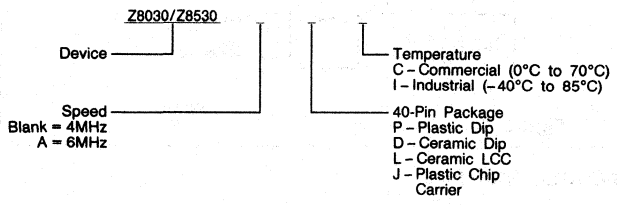
LS001300



LS001311

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Z8030	DC, PC, DI, ADC, APC
Z8530	DC, PC, DI, ADC, APC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

Z8030 PIN DESCRIPTION

Pin No.	Name	I/O	Description
9	V_{CC}		+5V Power Supply.
31	GND		Ground.
40, 1, 39, 2, 38, 3, 37, 4	AD_0-AD_7	I/O	Address/Data Bus (bidirectional, active High, 3-state). These multiplexed lines carry register addresses to the SCC as well as data or control information to and from the SCC.
35	$\overline{AS}$	I	Address Strobe (active Low). Addresses on AD_0-AD_7 are latched by the rising edge of this signal.
33	$\overline{CS_0}$	I	Chip Select 0 (active Low). This signal is latched concurrently with the addresses on AD_0-AD_7 and must be active for the intended bus transaction to occur.
32	CS_1	I	Chip Select 1 (active High). This second select signal must also be active before the intended bus transaction can occur. CS_1 must remain active throughout the transaction.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear to Send (active Low). If these pins are programmed as Auto Enables, a LOW on these inputs enables their respective transmitter. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
36	$\overline{DS}$	I	Data Strobe (active Low). This signal provides timing for the transfer of data into and out of the SCC. If $\overline{AS}$ and $\overline{DS}$ coincide, this is interpreted as a reset.
16, 24	$\overline{DTR/REQA}$, $\overline{DTR/REQB}$	O	Data Terminal Ready/Request (active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is HIGH only if IEI is HIGH and the CPU is not servicing a SCC interrupt or the SCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the SCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{DS}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of $\overline{AS}$.
20	PCLK	I	Clock. This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock, although the frequency of this clock must be at least 90% of the CPU clock frequency for a Z8000. PCLK is a TTL level signal. Maximum transmit rate is 1/4 PCLK.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud-rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request to Send (active Low). When the Request to Send RTS bit in Write Register 5 is set, the $\overline{RTS}$ signal goes LOW. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes HIGH after the transmitter is empty. In synchronous mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pins strictly follow the state of the RTS bit. Both pins can be used as general-purpose outputs.
34	R/$\overline{W}$	I	Read/Write. This signal specifies whether the operation to be performed is read or a write.
11, 29	$\overline{SYNCA}$, $\overline{SYNCB}$	I or O	Synchronization (active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function. In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven LOW two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$. In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronous pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.
15, 25	TxDA, TxDB	O	Transmit Data (active High). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I or O	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
10, 30	$\overline{W/REQA}$, $\overline{W/REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

00970D

Refer to page 7-1 for Essential Information on Military Devices

Z8530 PIN DESCRIPTION

Pin No.	Name	I/O	Description
9	V_{CC}		+5V Power Supply.
31	GND		Ground.
34	A/$\overline{B}$	I	Channel A/Channel B Select. This signal selects the channel in which the read or write operation occurs.
33	$\overline{CE}$	I	Chip Enable (active Low). This signal selects the SCC for a read or write operation.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear To Send (active Low). If these pins are programmed as Auto Enables, a LOW on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
32	D/$\overline{C}$	I	Data/Control Select. This signal defines the type of information transferred to or from the SCC. A HIGH means data is transferred; a LOW indicates a command.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
40, 1, 39, 2, 38, 3, 37, 4	D_0-D_7	I/O	Data Bus (3-state). These lines carry data and commands to and from the SCC.
16, 24	$\overline{DTR}$/$\overline{REQA}$, $\overline{DTR}$/$\overline{REQB}$	O	Data Terminal Ready/Request (active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is HIGH only if IEI is HIGH and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the SCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is HIGH). $\overline{INTACK}$ is latched by the rising edge of PCLK.
20	PCLK	I	Clock. This is the master SCC clock used to synchronize internal signals; PCLK is a TTL level signal.
36	$\overline{RD}$	I	Read (active Low). This signal indicates a read operation and, when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel, $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective $\overline{SYNC}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request To Send (active Low). When the Request to Send (RTS) bit in Write Register 5 is set, the $\overline{RTS}$ signal goes LOW. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes HIGH after the transmitter is empty. In synchronous mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
11, 29	$\overline{SYNCA}$, $\overline{SYNCB}$	I or O	Synchronization (active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function. In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven LOW two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$. In the Internal Synchronizaton mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.
15, 25	TxDA, TxDB	O	Transmit Data (active High). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I or O	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
35	$\overline{WR}$	I	Write (active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.
10, 30	$\overline{W}$/$\overline{REQA}$, $\overline{W}$/$\overline{REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

ARCHITECTURE

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to the Z8000 CPU (Z8030) or to a non-multiplexed CPU bus (Z8530). Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface with modems or other external devices (Figure 1).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two synchronous character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write-only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the Interrupt Pending bits (A only).

The registers for each channel are designated as follows:

WR0 – WR15 – Write Registers 0 through 15.
RR0 – RR3, RR10, RR12, RR13, RR15 – Read Registers 0 through 3, 10, 12, 13, 15.

The following table lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path

The transmit and receive data path illustrated in Figure 2 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and a 20-bit transmit shift register that can be loaded either from the sync-character registers or from the transmit data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

TABLE 1. READ AND WRITE REGISTER FUNCTIONS

READ REGISTER FUNCTIONS

RR0 Transmit/Receive buffer status and External status

RR1 Special Receive Condition status

RR2 Modified interrupt vector
(Channel B only)
Unmodified interrupt vector
(Channel A only)

RR3 Interrupt Pending bits
(Channel A only)

RR8 Receive buffer

RR10 Miscellaneous status

RR12 Lower byte of baud rate generator time constant

RR13 Upper byte of baud rate generator time constant

RR15 External/Status interrupt information

WRITE REGISTER FUNCTIONS

WR0 CRC initialize, initialization commands for the various modes, shift right/shift left command

WR1 Transmit/Receive interrupt and data transfer mode definition

WR2 Interrupt vector (accessed through either channel)

WR3 Receive parameters and control

WR4 Transmit/Receive miscellaneous parameters and modes

WR5 Transmit parameters and controls

WR6 Sync characters or SDLC address field

WR7 Sync character or SDLC flag

WR8 Transmit buffer

WR9 Master interrupt control and reset (accessed through either channel)

WR10 Miscellaneous transmitter/receiver control bits

WR11 Clock mode control

WR12 Lower byte of baud rate generator time constant

WR13 Upper byte of baud rate generator time constant

WR14 Miscellaneous control bits

WR15 External/Status interrupt control

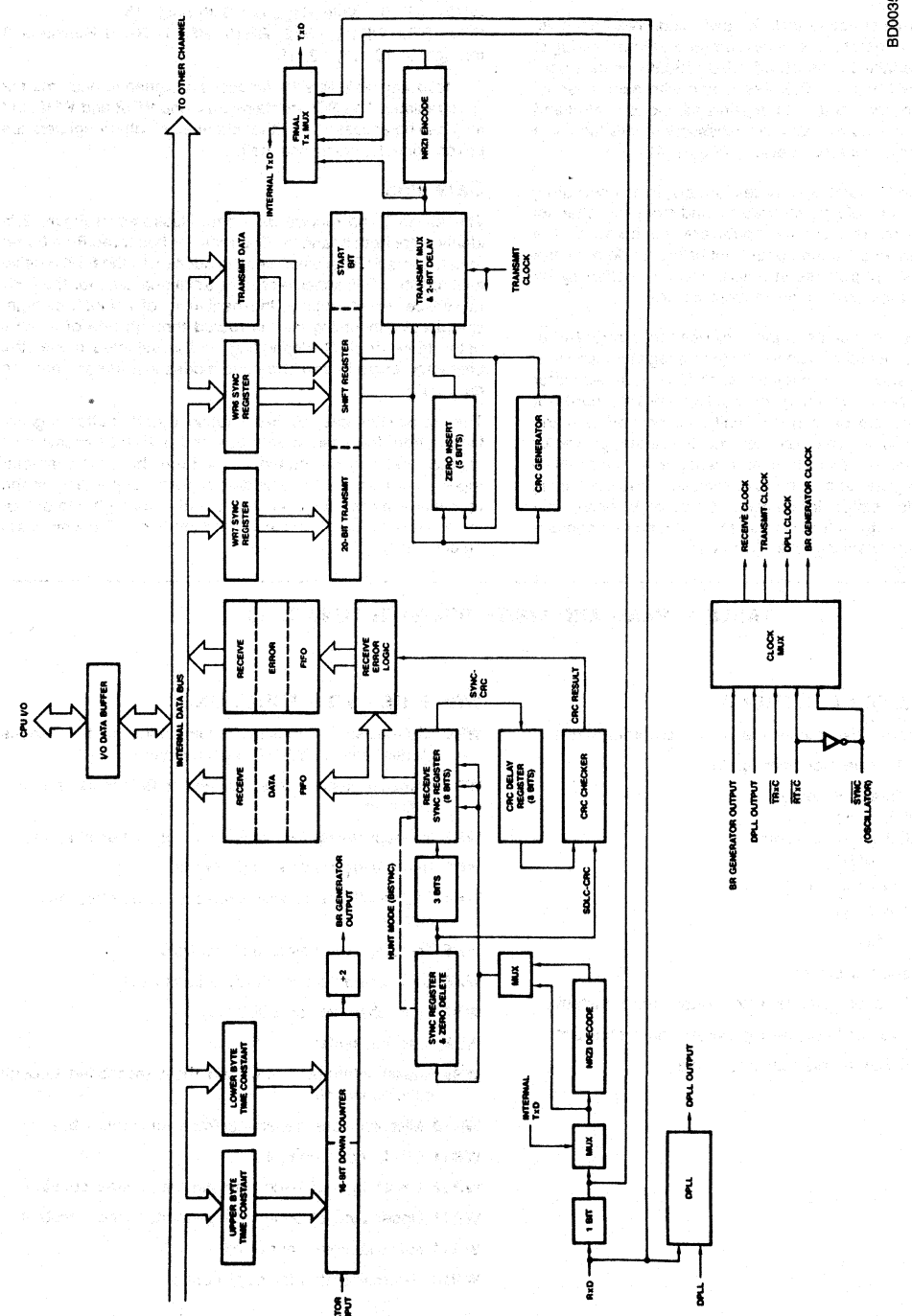


Figure 2. Data Path

00970D

Refer to page 7-1 for Essential Information on Military Devices

DETAILED DESCRIPTION

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

Data Communications Capabilities

The SCC provides two independent full-duplex channels programmable for use in any common asynchronous or synchronous data-communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in the Z8530 Logic Symbol). If the LOW does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals – a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In asynchronous modes, the SYNC pin may be programmed as an input used for functions, such as monitoring a ring indicator.

Synchronous Modes

The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronous pattern (Bisync), or with an external synchronous signal. Leading synchronous characters can be removed without interrupting the CPU.

5- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

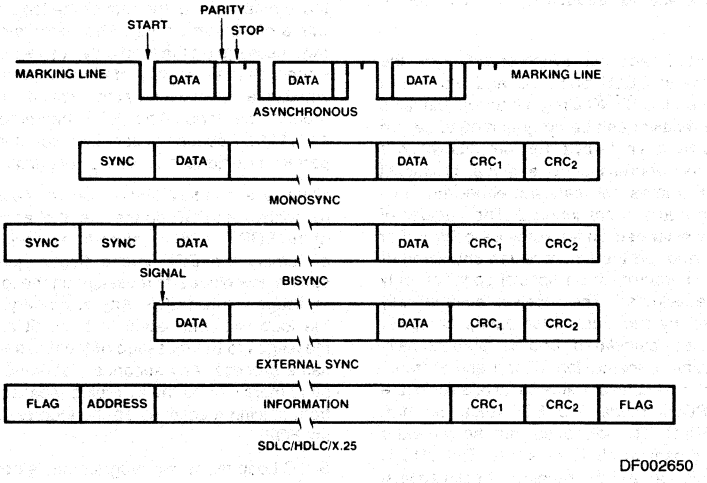


Figure 3. SCC Protocols

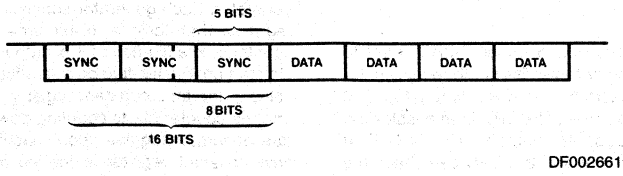


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols, such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in all synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there is no data or CRC to send in synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{SYNC}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1X mode. The parity options available in asynchronous modes are available in synchronous modes.

The SCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC LOOP MODE

The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

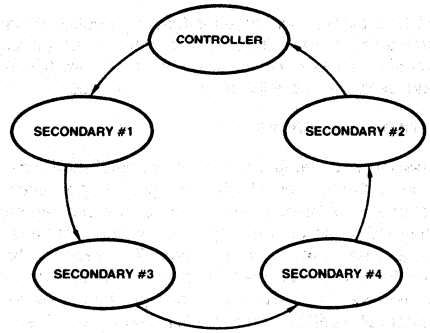


Figure 5. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary one of the EOP to a zero before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator

Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the flip-flop on the output is set in a High state; the value in the time constant register is loaded into the counter; and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero; the value in the time constant register is loaded into the counter; and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds.)

$$\text{baud rate} = \frac{1}{2\ (\text{time constant} + 2) \times (\text{BR clock period})}$$

Time-Constant Values for Standard Baud Rates at BR Clock = 3.9936MHz

Rate (Baud)	Time Constant (decimal notation)	Error
19200	102	–
9600	206	–
7200	275	0.12%
4800	414	–
3600	553	0.06%
2400	830	–
2000	996	0.04%
1800	1107	0.03%
1200	1662	–
600	3326	–
300	6654	–
150	13310	–
134.5	14844	0.0007%
110	18151	0.0015%
75	26622	–
50	39934	–

Digital Phase-Locked Loop

The SCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

Data Encoding

The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level, and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level, and a 0 is represented by a change in level. In FM_1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell, and a 0 is represented by no additional transition at the center of the bit cell. In FM_0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a 0. If the transition is 1/0, the bit is a 1.

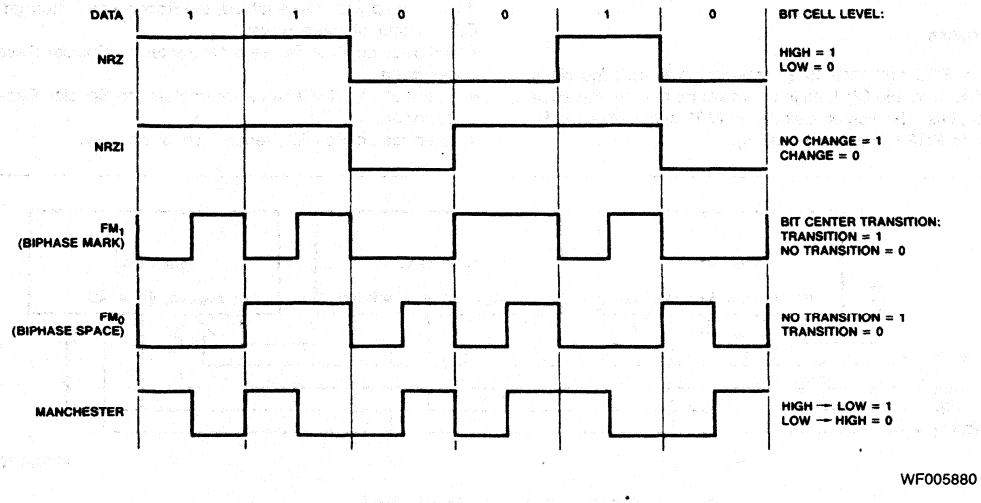


Figure 6. Data Encoding Methods

Auto Echo and Local Loopback

The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed, and the programmer is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The SCC is also capable of local loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data, and RxD is ignored (except to be echoed out via TxD). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in asynchronous, synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities

The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

Polling

All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

When a SCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 8 and 9).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

The other two bits are related to the Z-Bus interrupt priority chain (Figure 7). As a Z-Bus peripheral, the SCC may request an interrupt only when no higher priority device is requesting one; e.g., when IEI is HIGH. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is HIGH, the INT output is pulled LOW, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled LOW and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the Receive can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition.
- Interrupt on all Receive Characters or Special Receive condition.
- Interrupt on Special Receive condition only.

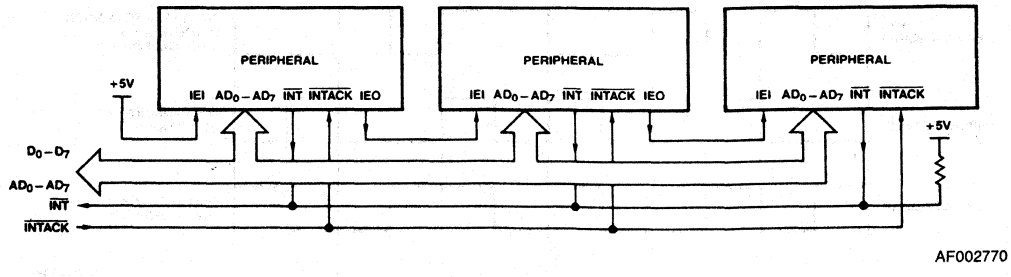


Figure 7. Z-Bus Interrupt Schedule

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode, End-of-Frame in SDLC mode, and optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, a zero count in the baud rate generator, the detection of a Break (asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{WAIT}/\overline{Request}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT}/\overline{REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC $\overline{REQUEST}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR}/\overline{REQUEST}$ line allows full-duplex operation under DMA control.

PROGRAMMING INFORMATION

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

The Z8030 and Z8530 differ in the way the system accesses these registers:

In the Z8030 all registers are directly addressable from the multiplexed Address Data bus. See Figure 10 and Figure 11 for timing. The Z8030 can operate in either of two modes: when bit 0 in Write Register 0 is reset (or after initialization with a hardware reset), Address lines AD_1 through AD_5 select the register to be read from or written into during Data Strobe $\overline{DS}$. (This is called left shift and is the natural Z8000 mode.) When bit 0 in Write Register 0 is set, Address lines AD_0 through AD_4 select the register to be read from or written into. (This is called right shift and is more natural for interfacing with other microprocessors.)

Table 2 describes the register addressing for both modes.

Channel A/Channel B selection is made either by AD_0 or by AD_5.

If Bit D_0 in WR0 is reset (or after hardware reset):
AD_5 selects the channel (0 = B, 1 = A)
(this is called "Select Shift Left Mode").

If Bits D_0 and D_1 in WR0 are set:
AD_0 selects the channel (0 = B, 1 = A)
(this is called "Select Shift Right Mode).

In the Z8530 only the four data registers (Read and Write for Channels A and B) are directly selected by a HIGH on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a LOW on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. If bit D3 in WR0 is 1 and bits 5 and 6 are 0, then bits 0, 1, 2 address the higher registers 8 through 15. If bits 4, 5, 6 contain a different code, bits 0, 1, 2 address the lower registers 0 through 7 as shown in Table 3.

TABLE 2. REGISTER ADDRESSING (Z8030 ONLY)

AD_4	AD_3	AD_2	AD_1	Write Register	Read Register
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	(0)
0	1	0	1	5	(1)
0	1	1	0	6	(2)
0	1	1	1	7	(3)
1	0	0	0	Data	Data
1	0	0	1	9	–
1	0	1	0	10	10
1	0	1	1	11	(15)
1	1	0	0	12	12
1	1	0	1	13	13
1	1	1	0	14	(10)
1	1	1	1	15	15

Writing to or reading from any register except RR0, WR0 and the Data Registers thus involves two operations:

First, write the appropriate code into WR0, then follow this by a write or read operation on the register thus specified. Bits 0 through 4 in WW0 are automatically cleared after this operation, so that WW0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made by the A/$\overline{B}$ input (HIGH = A, LOW = B)

In both Z8030 and Z8530, the system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set and, finally, receiver or transmitter enable.

TABLE 3. REGISTER ADDRESSING (Z8530 ONLY)

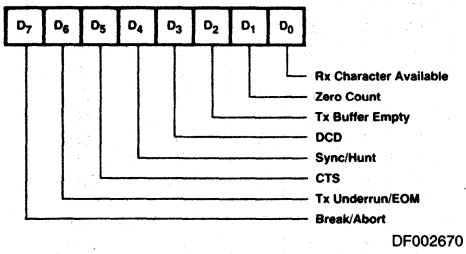
D/C "Point High" Code in WR0:		D_2 In WR0:	D_1 In WR0:	D_0 In WR0:	Write Register	Read Register
HIGH	Either way	X	X	X	Data	Data
LOW	Not true	0	0	0	0	0
LOW	Not true	0	0	1	1	1
LOW	Not true	0	1	0	2	2
LOW	Not true	0	1	1	3	3
LOW	Not true	1	0	0	4	(0)
LOW	Not true	1	0	1	5	(1)
LOW	Not true	1	1	0	6	(2)
LOW	Not true	1	1	1	7	(3)
LOW	True	0	0	0	Data	Data
LOW	True	0	0	1	9	–
LOW	True	0	1	0	10	10
LOW	True	0	1	1	11	(15)
LOW	True	1	0	0	12	12
LOW	True	1	0	1	13	13
LOW	True	1	1	0	14	(10)
LOW	True	1	1	1	15	15

Read Registers

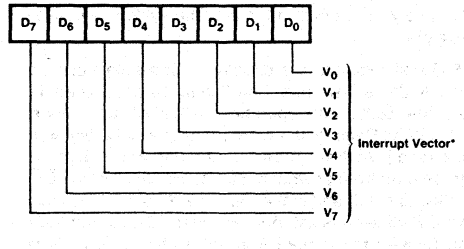
The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 8 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

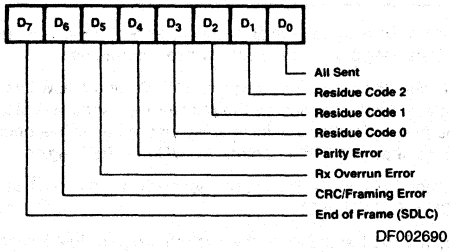
Read Register 0



Read Register 2



Read Register 1



Read Register 3

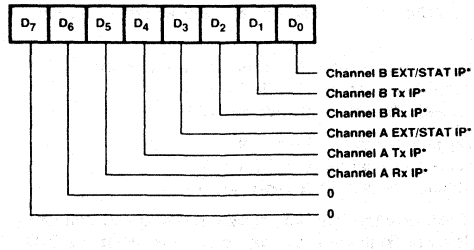


Figure 8. Read Register Bit Functions

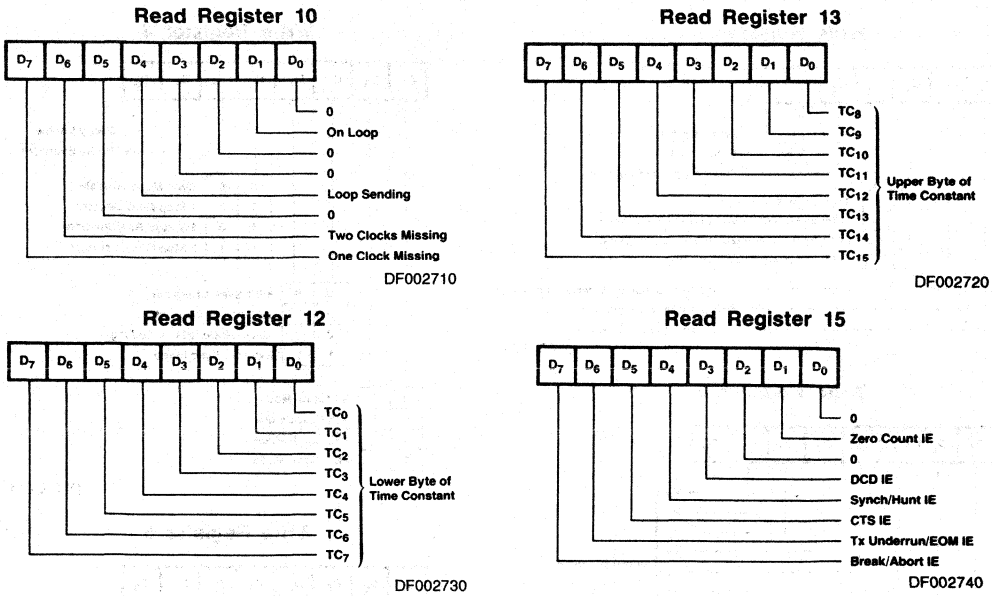


Figure 8. Read Register Bit Functions (Cont.)

Write Registers

The SCC contains 15 write registers (16 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 9 shows the format of each write register.

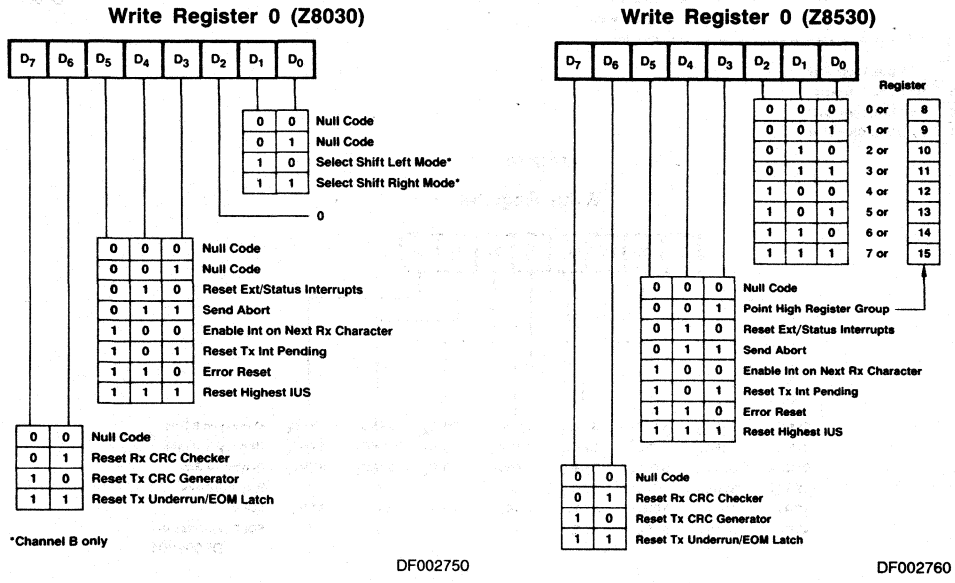
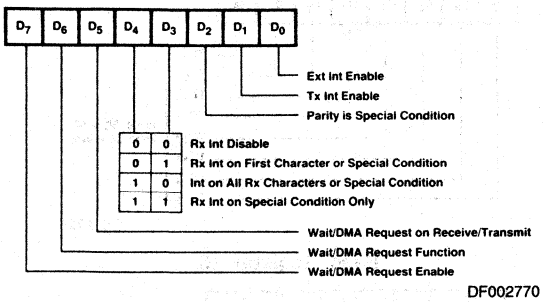


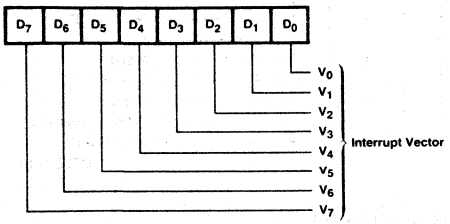
Figure 9. Write Register Bit Functions

Write Register 1



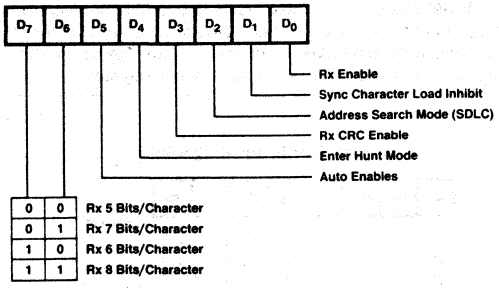
DF002770

Write Register 2



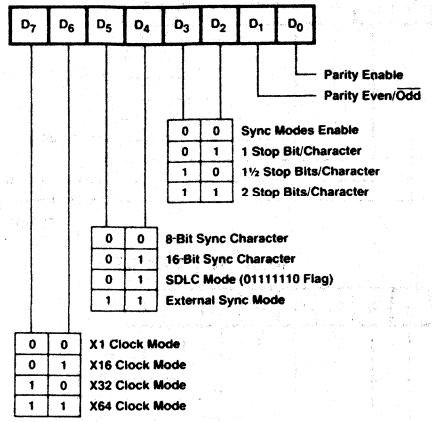
DF002780

Write Register 3



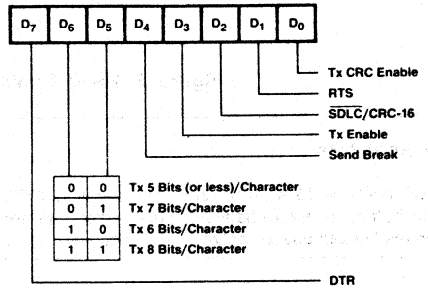
DF002790

Write Register 4



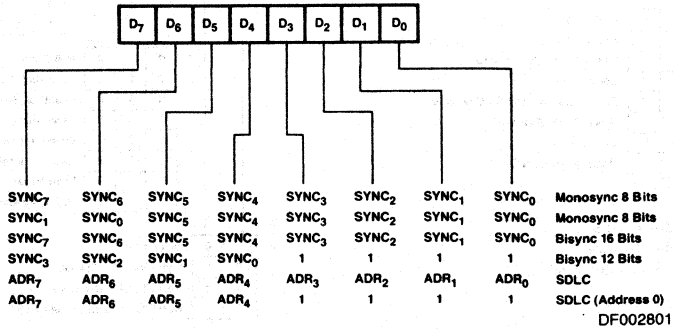
DF002811

Write Register 5



DF002820

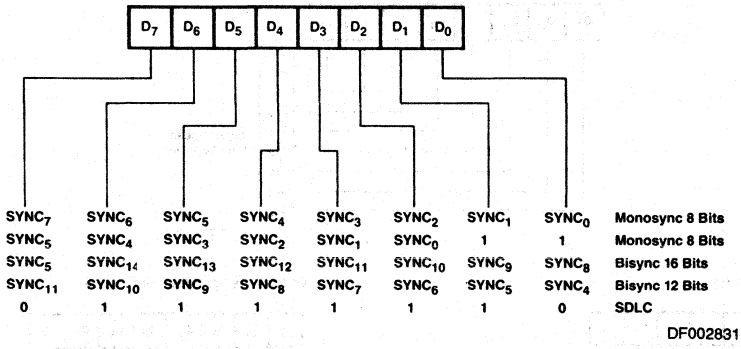
Write Register 6



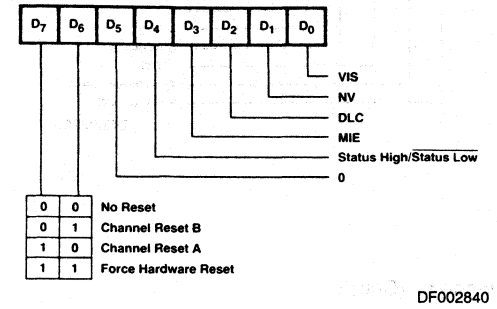
DF002801

Figure 9. Write Register Bit Functions (Cont.)

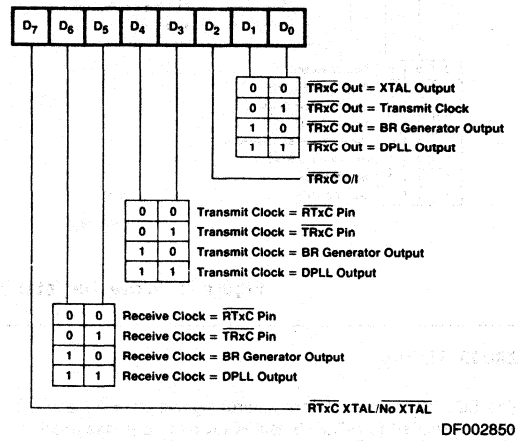
Write Register 7



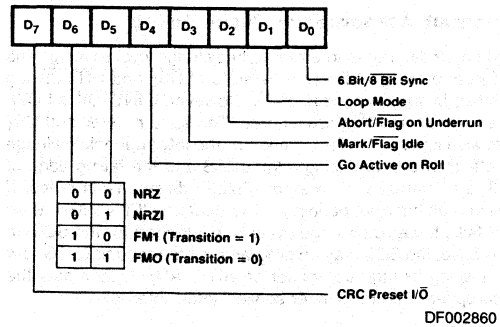
Write Register 9



Write Register 11



Write Register 10



Write Register 12

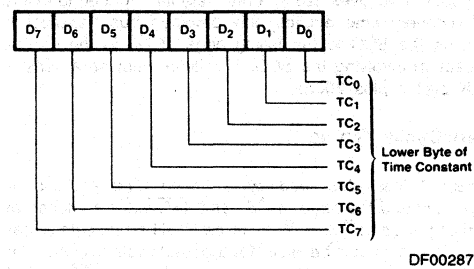


Figure 9. Write Register Bit Functions (Cont.)

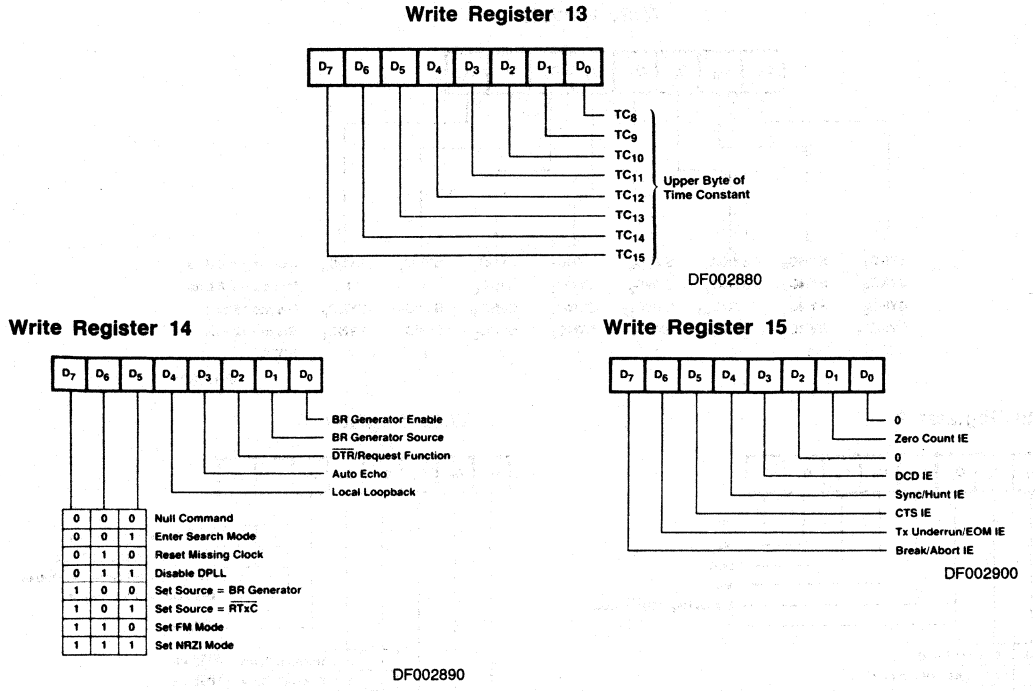


Figure 9. Write Register Bit Functions (Cont.)

Z8030 Timing

The SCC generates internal control signals from $\overline{AS}$ and $\overline{DS}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{AS}$ and $\overline{DS}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC to the falling edge of $\overline{DS}$ in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200ns.

Read Cycle Timing

Figure 10 illustrates read cycle timing. The address on $AD_0 - AD_7$ and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be HIGH to indicate a read cycle. CS_1 must also be HIGH for the read cycle to occur. The data bus drivers in the SCC are then enabled while $\overline{DS}$ is LOW.

Write Cycle Timing

Figure 11 illustrates write cycle timing. The address on $AD_0 - AD_7$ and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be LOW to indicate a write cycle. CS_1 must be HIGH for the write cycle to occur. $\overline{DS}$ Low strobes the data into the SCC.

Interrupt Acknowledge Cycle Timing

Figure 12 illustrates interrupt acknowledge cycle timing. The address on $AD_0 - AD_7$ and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. However, if $\overline{INTACK}$ is LOW, the address and $\overline{CS_0}$ are ignored. The state of R/$\overline{W}$ and CS_1 are also ignored for the duration of the interrupt acknowledge cycle. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is HIGH when $\overline{DS}$ falls, the acknowledge cycle was intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{DS}$ Low by placing its interrupt vector on $AD_0 - AD_7$. It then sets the appropriate interrupt-under-service latch internally.

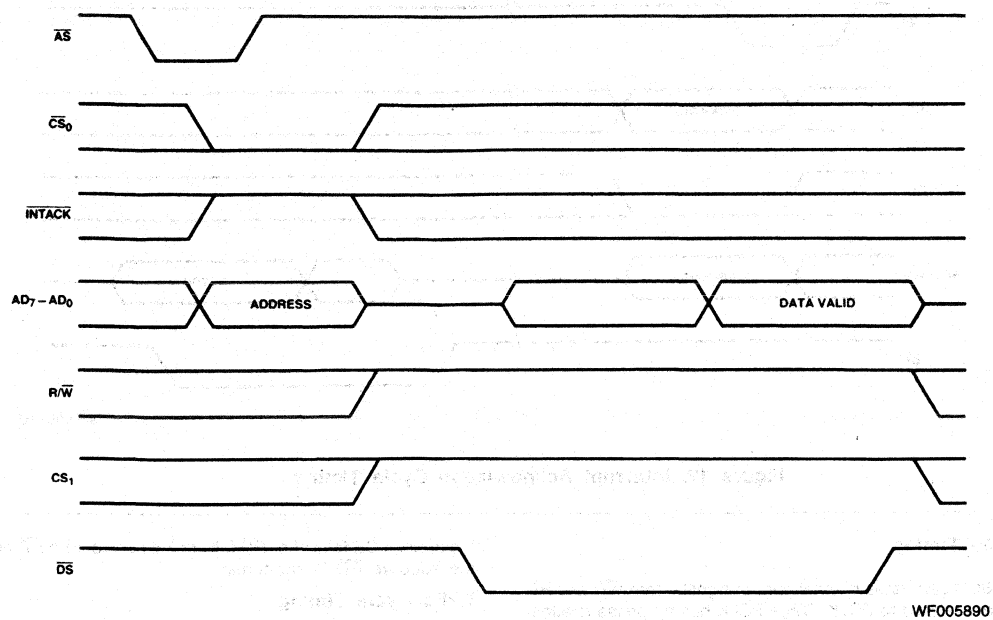


Figure 10. Read Cycle Timing

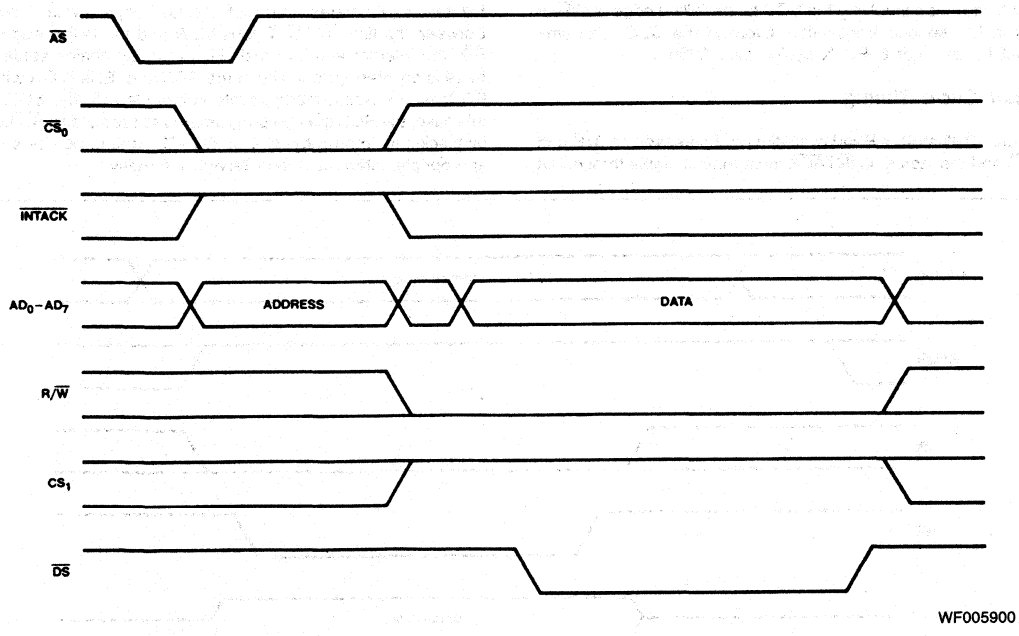


Figure 11. Write Cycle Timing

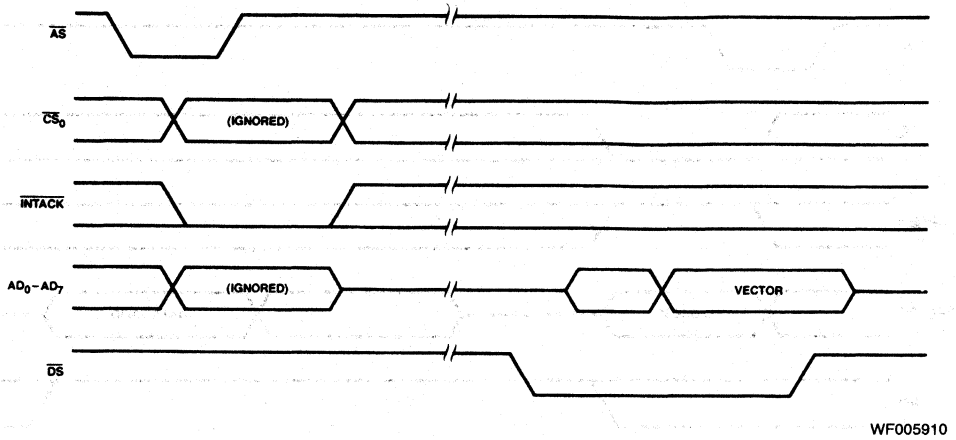


Figure 12. Interrupt Acknowledge Cycle Timing

Z8530 Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the SCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200ns.

Read Cycle Timing

Figure 13 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

Write Cycle Timing

Figure 14 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened

Interrupt Acknowledge Cycle Timing

Figure 15 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes LOW and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is HIGH when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on $D_0 - D_7$, and it then sets the appropriate Interrupt-Under-Service internally.

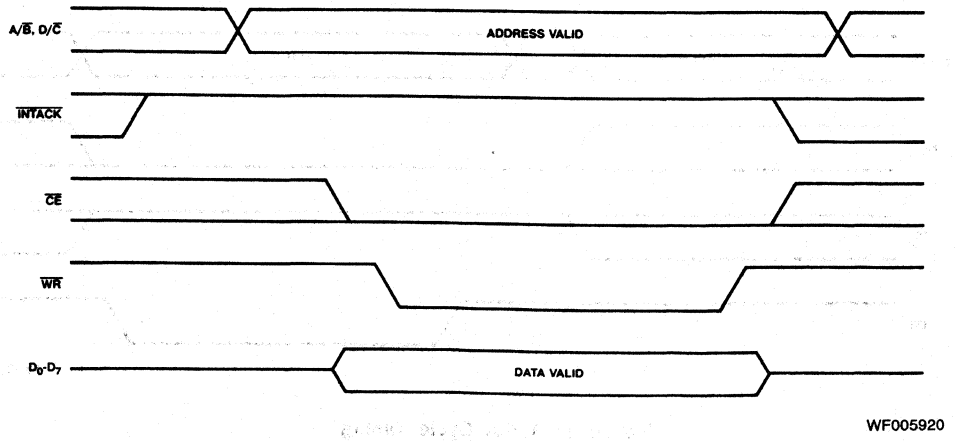


Figure 13. Read Cycle Timing

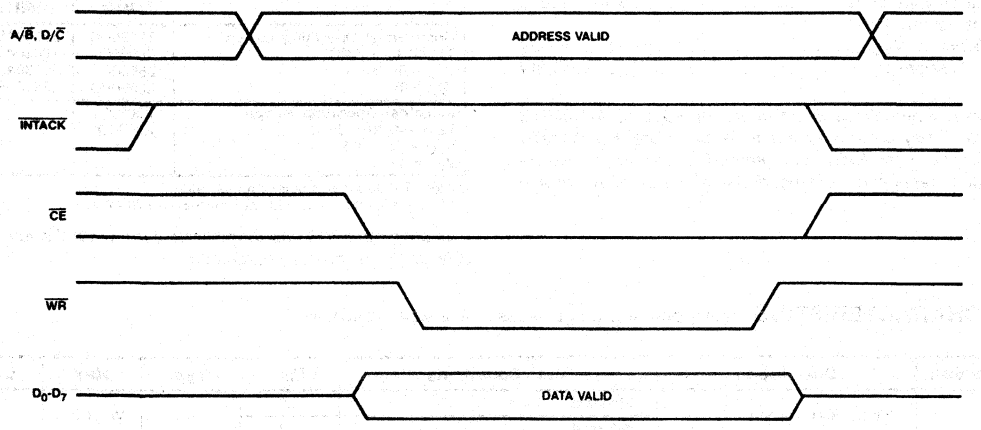


Figure 14. Write Cycle Timing

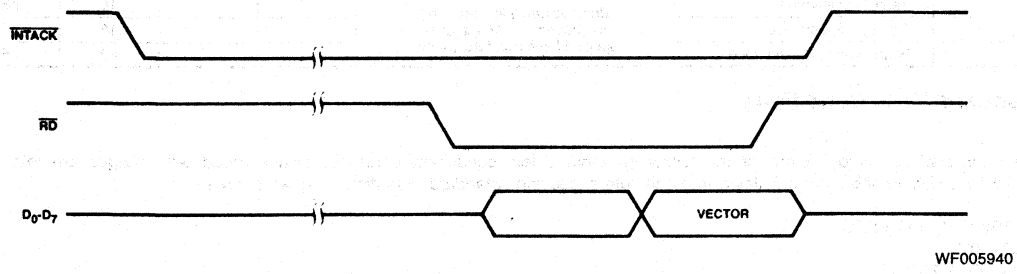


Figure 15. Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin
Relative to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.8W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	4MHz	6MHz
Commercial Operating Range T_A = 0 to +70°C V_{CC} = 5V ±5% V_{SS} = 0V	Z8030DC Z8030PC Z8530DC Z8530PC	Z8030ADC Z8030APC Z8530ADC Z8530APC
Industrial Operating Range T_A = −40 to +85°C V_{CC} = 5V ±10% V_{SS} = 0V	Z8030DI Z8530DI	

Notes: T_A denotes ambient temperature.
Add suffix B to indicate burn-in requirement.

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

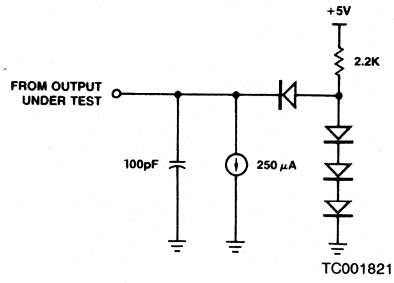
Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IH}	Input HIGH Voltage	Standard	2.0		V_{CC}+0.3	V
		Industrial	2.2			
V_{IL}	Input LOW Voltage		−0.3		0.8	V
V_{OH}	Output HIGH Voltage	I_{OH} = −250μA	2.4			V
V_{OL}	Output LOW Voltage	I_{OL} = +2.0mA			0.4	V
I_{IL}	Input Leakage	0.4 ≤ V_{IN} ≤ +2.4V			±10.0	μA
I_{OL}	Output Leakage	0.4 ≤ V_{OUT} ≤ +2.4V			±10.0	μA
I_{CC}	V_{CC} Supply Current				250	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.			10	pF
C_{OUT}	Output Capacitance				15	pF
$C_{I/O}$	Bidirectional Capacitance				20	pF

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

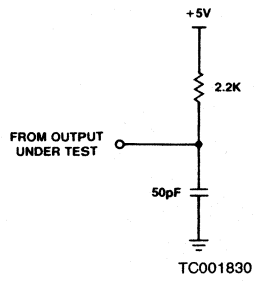
+4.75V ≤ V_{CC} ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ 70°C

Standard Test Load



TC001821

Open Drain Test Load



TC001830

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

GENERAL TIMING

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TdPC(REQ)	PCLK ↓ to W/REQ Valid Delay		250		250	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350	ns
3	TsRXC(PC)	RxC ↑ to PCLK ↑ Set-up Time (Notes 1, 4)	80	T_{WPCL}	70	T_{WPCL}	ns
4	TsRXD(RXCr)	RxD to RxC ↑ Set-up Time (XI Mode) (Note 1)	0		0		ns
5	ThRXD(RXCr)	RxD to RxC ↑ Hold Time (XI Mode) (Note 1)	150		150		ns
6	TsRXD(RXCf)	RxD to RxC ↓ Set-up Time (XI Mode) (Notes 1, 5)	0		0		ns
7	ThRXD(RXCf)	RxD to RxC ↓ Hold Time (XI Mode) (Notes 1, 5)	150		150		ns
8	TsSY(RXC)	SYNC to RxC ↑ Set-up Time (Note 1)	−200		−200		ns
9	ThSY(RXC)	SYNC to RxC ↑ Hold Time (Note 1)	3TcPC +200		3TcPC +200		ns
10	TsTXC(PC)	TxC ↓ to PCLK ↑ Set-up Time (Notes 2, 4)	0		0		ns
11	TdTXCf(TXD)	TxC ↓ to TxD Delay (XI Mode) (Note 2)		300		230	ns
12	TdTXCr(TXD)	TxC ↑ to TxD Delay (XI Mode) (Notes 2, 5)		300		230	ns
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)		200		200	ns
14	TwRTXh	RTxC High Width (Note 6)	180		180		ns
15	TwRTXl	RTxC Low Width (Note 6)	180		180		ns
16	TcRTX	RTxC Cycle Time (Note 6)	400		400		ns
17	TcRTXX	Crystal Oscillator Period (Note 3)	250	1000	250	1000	ns
18	TwTRXh	TRxC High Width (Note 6)	180		180		ns
19	TwTRXl	TRxC Low Width (Note 6)	180		180		ns
20	TcTRX	TRxC Cycle Time (Note 6)	400		400		ns
21	TwEXT	DCD or CTS Pulse Width	200		200		ns
22	TwSY	SYNC Pulse Width	200		200		ns

Notes:
1. RxC is RTxC or TRxC, whichever is supplying the receive clock.
2. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
3. Both RTxC and SYNC have 30pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC and PCLK or TxC and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.

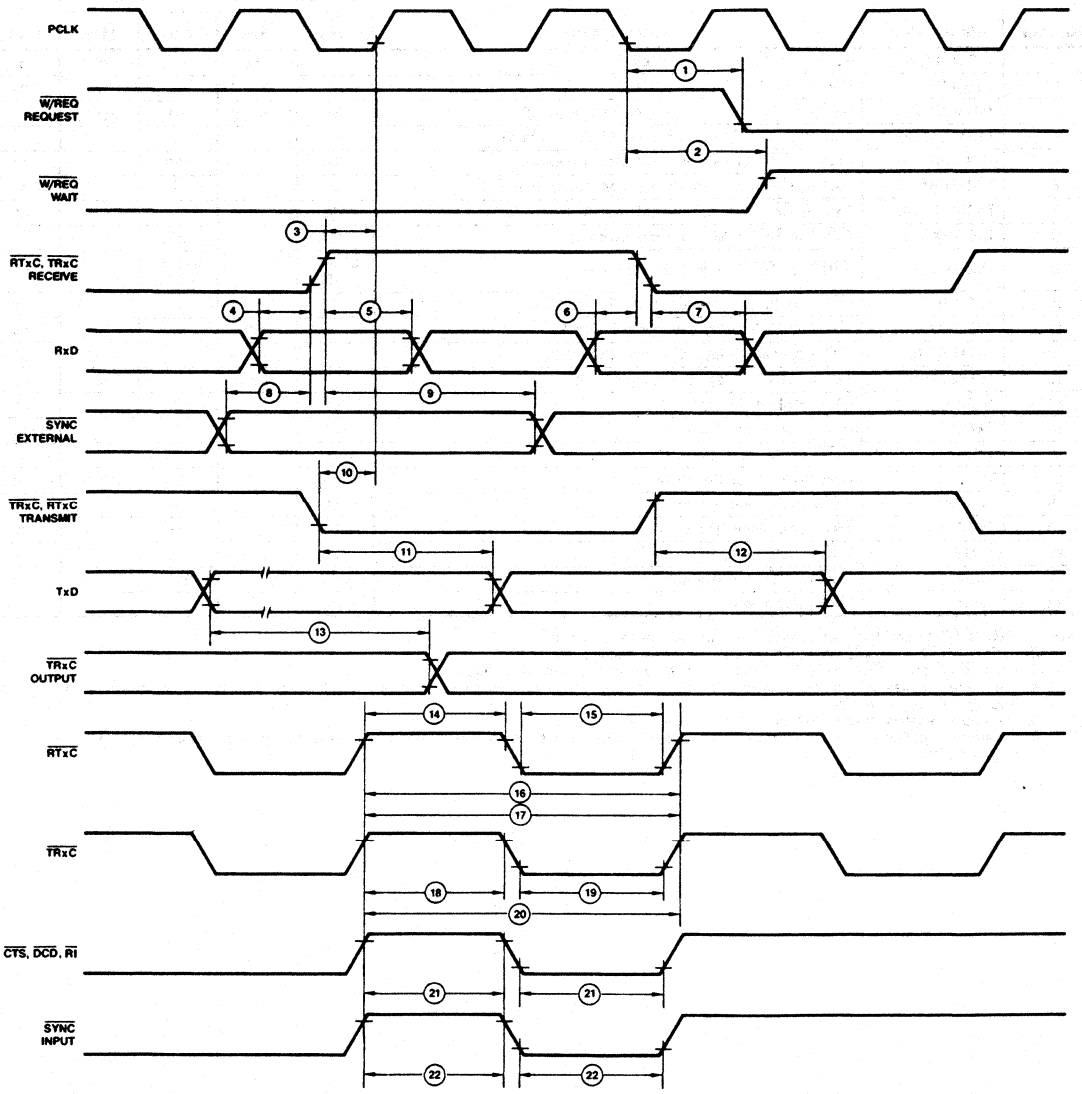


Figure 16. General Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

SYSTEM TIMING (Z8030)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TdRXC(REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ Wait Inactive Delay (Notes 1, 2)	8	12	8	12	TcPC
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPC
4	TdRXC(INT)	$\overline{RxC}$ ↑ $\overline{INT}$ Valid Delay (Notes 1, 2)	8 +2	12 +3	8 +2	12 +3	TcPC $\overline{AS}$
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPC
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	8	5	8	TcPC
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPC
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	4 +2	6 +3	4 +2	6 +3	TcPC $\overline{AS}$
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	3	2	3	TcPC
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	3	2	3	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

SYSTEM TIMING (Z8530)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TdRXC(REQ)	$\overline{RxC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	12	8	12	TcPC
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2)	4	7	4	7	TcPC
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPC
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPC
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	8	5	8	TcPC
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPC
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPC
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPC
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

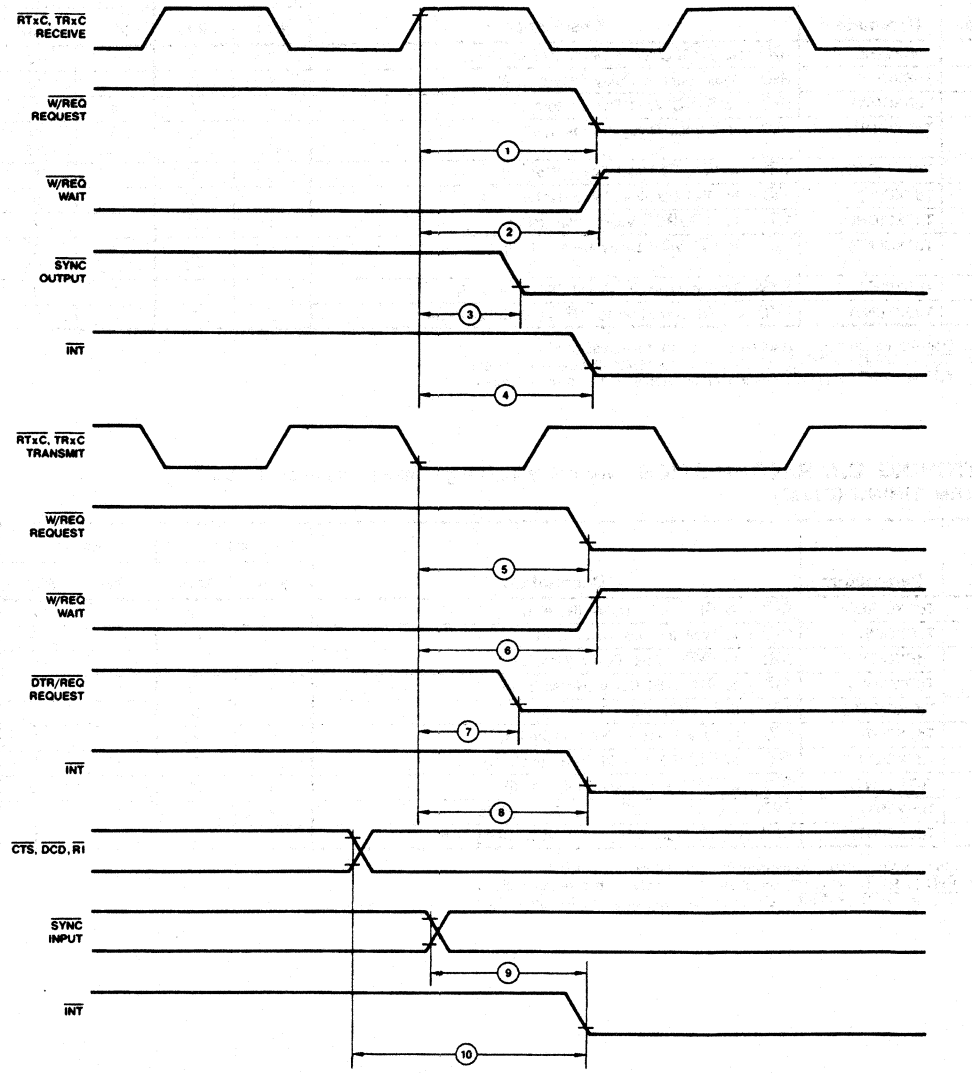


Figure 17. System Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

READ AND WRITE TIMING (Z8030)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TwAS	$\overline{AS}$ LOW Width	70		50		ns
2	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		ns
3	TsCS0(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Set-up Time (Note 1)	0		0		ns
4	ThCS0(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Hold Time (Note 1)	60		40		ns
5	TsCS1(DS)	CS_1 to $\overline{DS}$ ↓ Set-up Time (Note 1)	100		80		ns
6	ThCS1(DS)	CS_1 to $\overline{DS}$ ↑ Hold Time (Note 1)	55		40		ns
7	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Set-up Time	0		0		ns
8	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		250		ns
9	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Set-up Time	100		80		ns
10	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		ns
11	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Set-up Time	0		0		ns
12	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	60		40		ns
13	TwDSl	$\overline{DS}$ LOW Width	390		250		ns
14	TrC	Valid Access Recovery Time (Note 2)	6TcPC +200		6TcPC +130		ns
15	TsA(AS)	Address to $\overline{AS}$ ↑ Set-up Time (Note 1)	30		10		ns
16	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time (Note 1)	50		30		ns
17	TsDW(DS)	Write Data to $\overline{DS}$ ↓ Set-up Time	30		20		ns
18	ThDW(DS)	Write Date to $\overline{DS}$ ↑ Hold Time	30		20		ns
19	TdDS(DA)	$\overline{DS}$ ↓ to Data Active Delay	0		0		ns
20	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		ns
21	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	ns
22	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Valid Delay		520		335	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Parameter applies only between transactions involving the Z8030.

00970D

Refer to page 7-1 for Essential Information on Military Devices

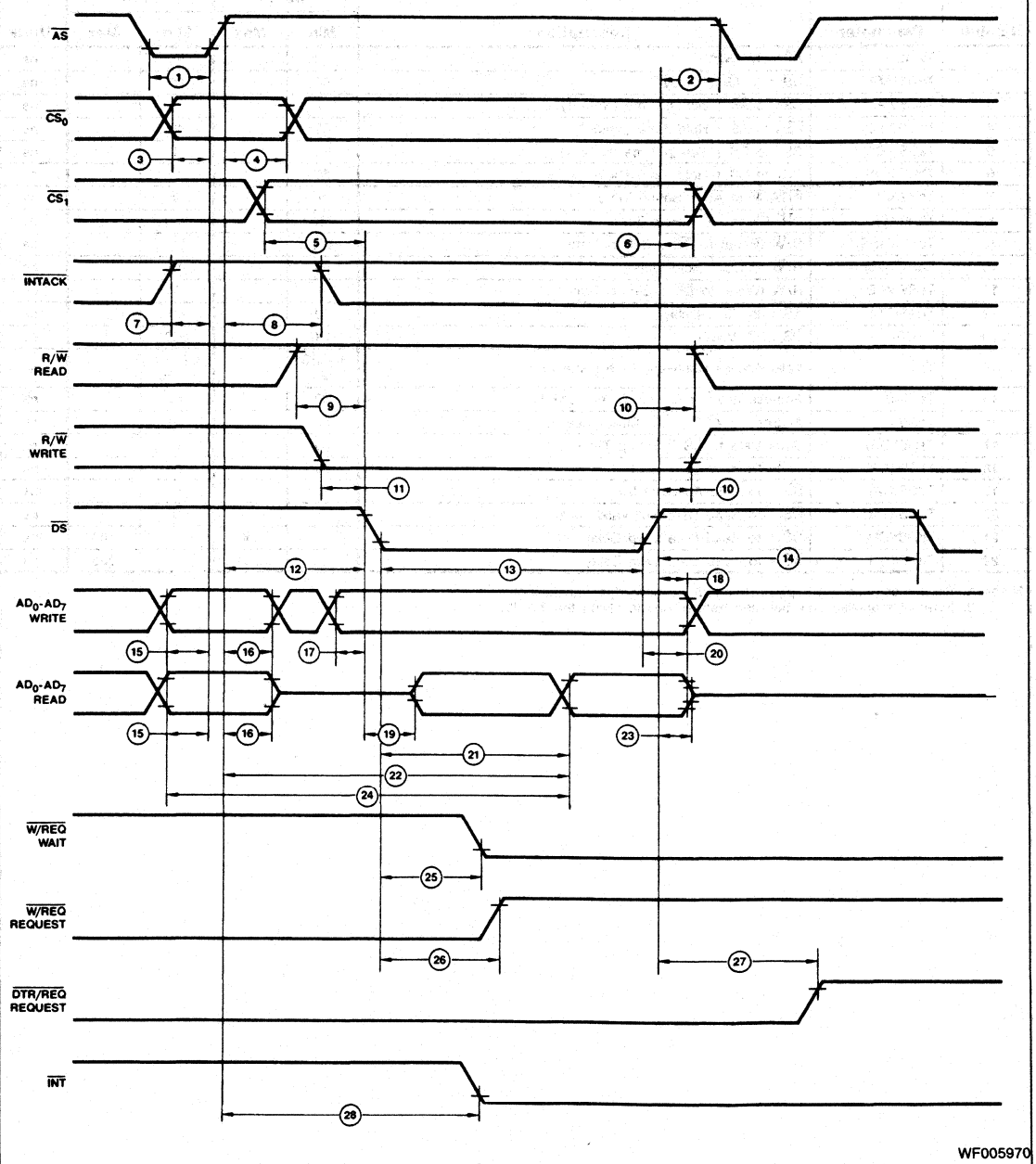


Figure 18. Read and Write Timing (Z8030)

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

READ AND WRITE TIMING (Z8530)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TwPCl	PCLK Low Width	105	2000	70	1000	ns
2	TwPCh	PCLK High Width	105	2000	70	1000	ns
3	TfPC	PCLK Fall Time		20		10	ns
4	TrPC	PCLK Rise Time		20		15	ns
5	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Set-up Time	80		80		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Set-up Time	80		80		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TslA(PC)	$\overline{INTACK}$ to PCLK ↑ Set-up Time	0		0		ns
11	TslAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Set-up Time (Note 1)	200		160		ns
12	ThlA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
13	TslAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Set-up Time (Note 1)	200		160		ns
14	ThlA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
15	ThlA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Set-up Time	0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Set-up Time	100		70		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Set-up Time (Note 1)	0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Set-up Time (Note 1)	100		70		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	390		250		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70		45	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ±0.5V change in the output with a maximum DC load and minimum AC load.

00970D

Refer to page 7-1 for Essential Information on Military Devices

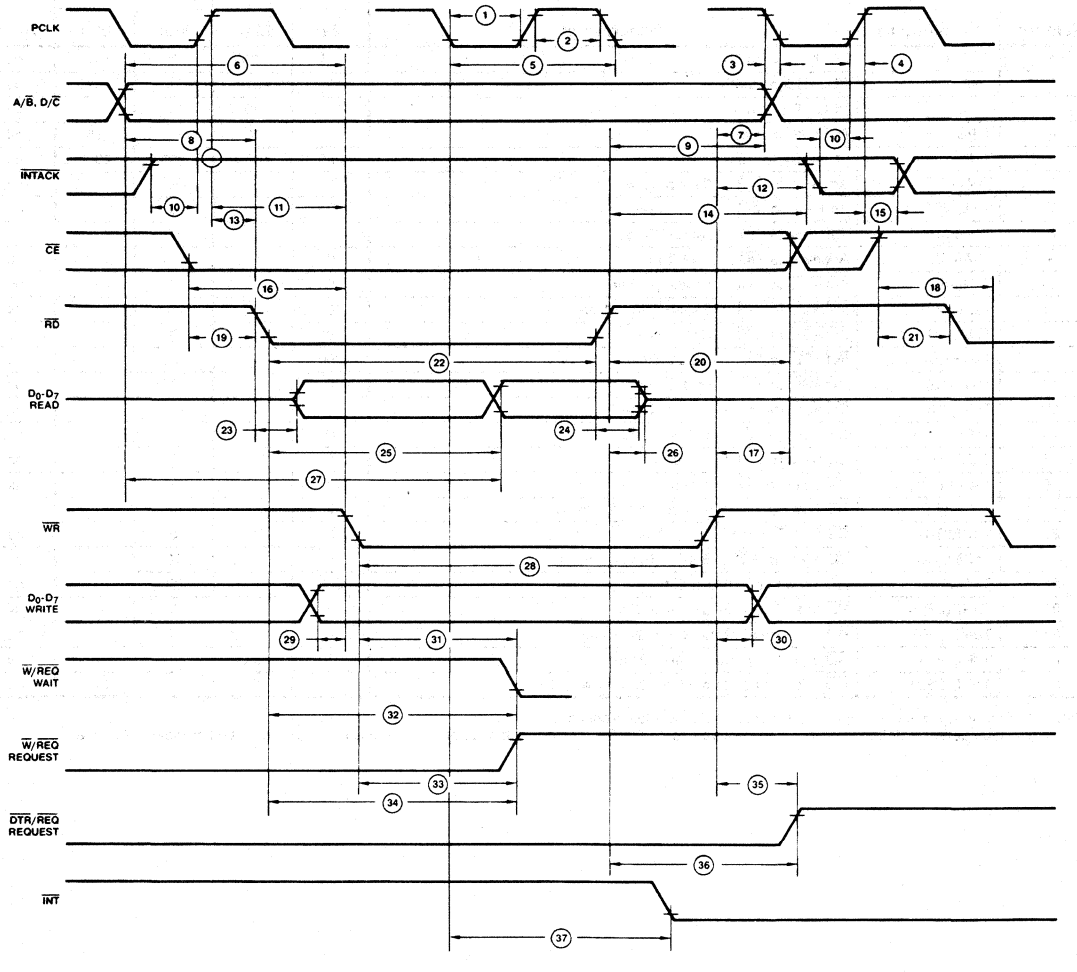


Figure 19. Read and Write Timing (Z8530)

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING (Z8030)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
23	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay (Note 3)		70		45	ns
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		570		420	ns
25	TdDS(W)	$\overline{DS}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
26	TdDSf(REQ)	$\overline{DS}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
27	TdDSr(REQ)	$\overline{DS}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC + 300		5TcPC + 250	ns
28	TdAS(INT)	$\overline{AS}$ ↑ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
29	TdAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
30	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		250		ns
31	TdDSA(DR)	$\overline{DS}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180	ns
32	TsIEI(DSA)	IEI to $\overline{DS}$ ↓ (Acknowledge) Set-up Time	120		100		ns
33	ThIEI(DSA)	IEI to $\overline{DS}$ ↑ (Acknowledge) Hold Time	0		0		ns
34	TdIEI(IEO)	IEI to IEO Delay		120		100	ns
35	TdAS(IEO)	$\overline{AS}$ ↑ to IEO Delay (Note 6)		250		250	ns
36	TdDSA(INT)	$\overline{DS}$ ↓ (Acknowledge) to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
37	TdDS(ASQ)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay for No Reset	30		15		ns
38	TdASQ(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay for No Reset	30		30		ns
39	TwRES	$\overline{AS}$ and $\overline{DS}$ Coincident Low for Reset (Note 7)	250		250		ns
40	TwPCl	PCLK Low Width	105	2000	70	1000	ns
41	TwPCh	PCLK High Width	105	2000	70	1000	ns
42	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
43	TrPC	PCLK Rise Time		20		15	ns
44	TfPC	PCLK Fall Time		20		10	ns

Notes: 3. Float delay is defined as the time required for a ±0.5V change in the output with a maximum DC load and minimum AC load.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any Z8030 in the daisy chain, TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain, TsIEI(DSA) for the Z8030, and TdIEIf(IEO) for each device separating them in the daisy chain.
6. Parameter applies only to a Z8030 pulling $\overline{INT}$ LOW at the beginning of the Interrupt Acknowledge transaction.
7. Internal circuitry allows for the reset provided by the Z8 to be recognized as a reset by the Z8030.

All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0."

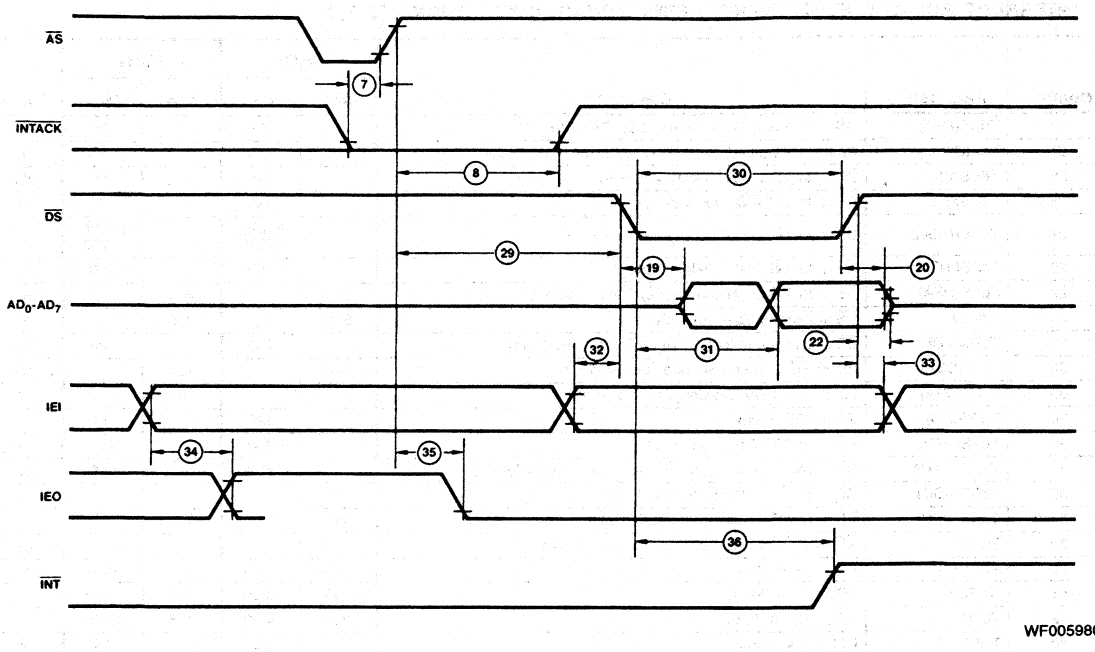


Figure 20. Interrupt Acknowledge Timing (Z8030)

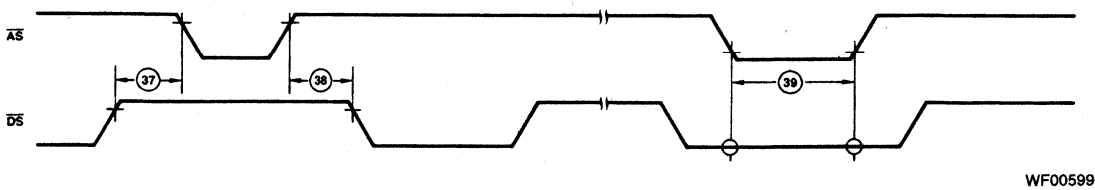


Figure 21. Reset Timing (Z8030)

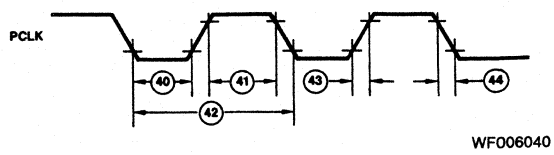


Figure 22. Cycle Timing (Z8030)

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING (Z8530)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		590		420	ns
28	TwWRl	$\overline{WR}$ Low Width	390		250		ns
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Set-up Time	0		0		ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	285		250		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		190		180	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Set-up Time	120		100		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		15		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		30		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		250		ns
49	Trc	Valid Access Recovery Time (Note 3)	6TcPC +200		6TcPC +130		ns

Notes: 3. Parameter applies only between transactions involving the Z8530.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any Z8530 in the daisy chain, TdIAI(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the Z8530, and TdIEIf(IEO) for each device separating them in the daisy chain.

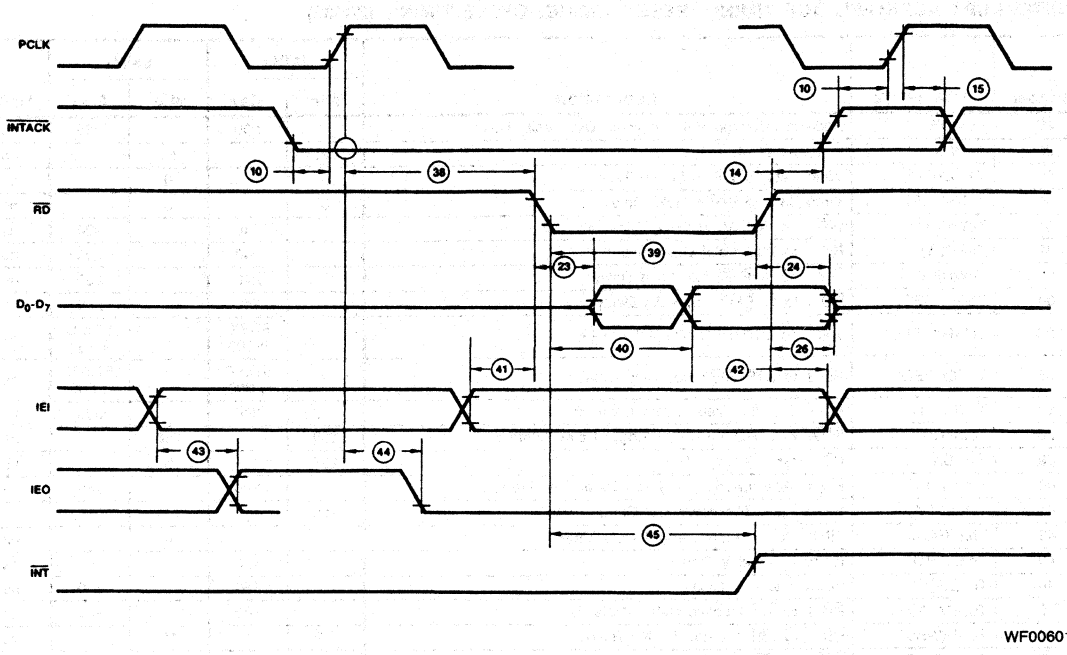


Figure 23. Interrupt Acknowledge Timing (Z8530)

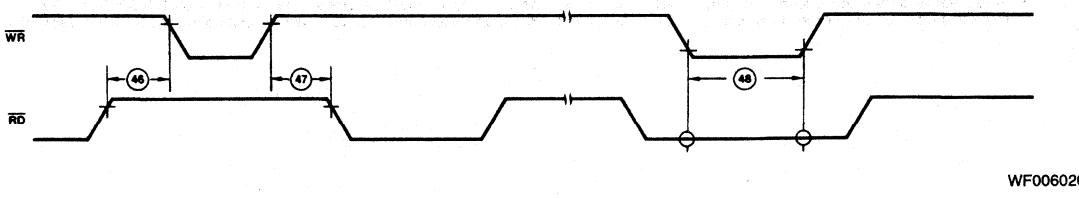


Figure 24. Reset Timing (Z8530)

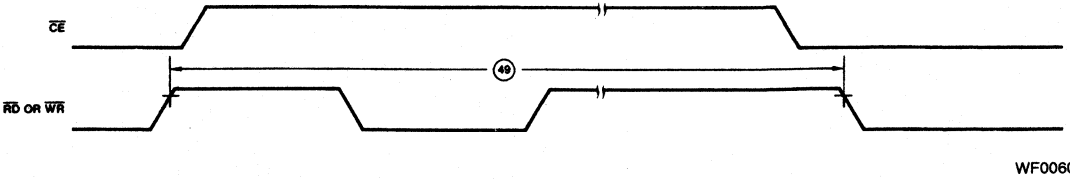


Figure 25. Cycle Timing (Z8530)

Z8031 • Z8531 (ASCC)

Asynchronous Serial Communications Controller

DISTINCTIVE CHARACTERISTICS

- **Two 1M.bps full duplex serial channels –** Each channel has independent oscillator, band-rate generator, and PLL for clock recovery, dramatically reducing the need for external components.
- **Programmable protocols –** NRZ, NRZI, and FM data encoding supported under program control.
- **Programmable Asynchronous Modes –** 5 to 8 bit characters with programmable stop bits, clock break detect, and error conditions.
- **Z8000* compatible –** The Z8031 interfaces directly to the Z8000 CPU bus and to the Z8000 interrupt structure.
- **Compatible with non-multiplexed bus –** The Z8531 interfaces easily to most other CPUs.

GENERAL DESCRIPTION

Asynchronous Serial Communications Controllers are dual-channel communications peripherals designed for use with 8- and 16-bit microprocessors. They function as serial-to-parallel, and parallel-to-serial converter/controllers, and contain a variety of new, sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops and crystal oscillators, to dramatically reduce the need for external circuitry.

Both channels have facilities for modem control; in cases where these controls aren't needed, they can be used for general purpose I/O.

The Z8031 is directly compatible with the Z8000 and 8086 CPUs, while the Z8531 is designed for non-multiplexed buses, and is easily interfaced with most other CPUs such as 8080, Z80, 6800, 68000 and Multibus.**

BLOCK DIAGRAM

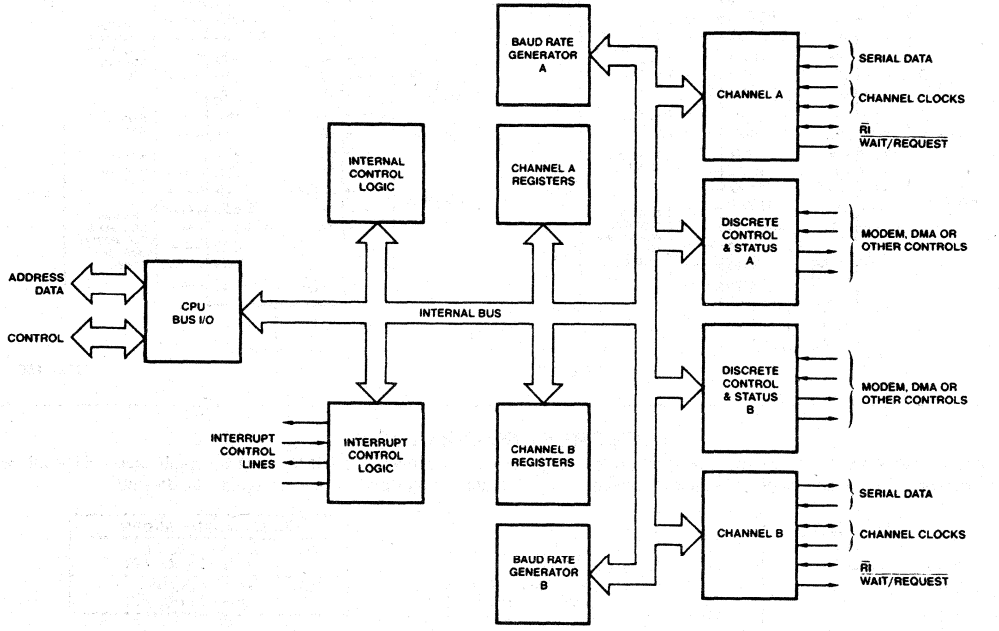


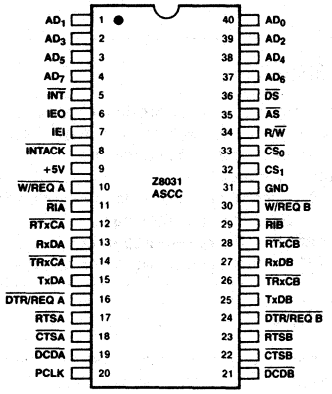
Figure 1.

03818C

Refer to page 7-1 for Essential Information on Military Devices

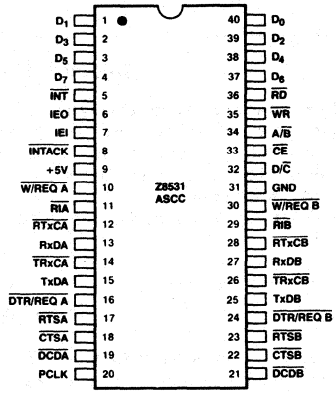
CONNECTION DIAGRAM
Top View

D-40-2, P-40-1



CD005080

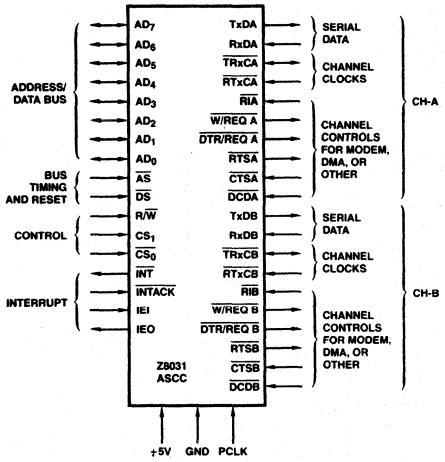
D-40-2, P-40-1



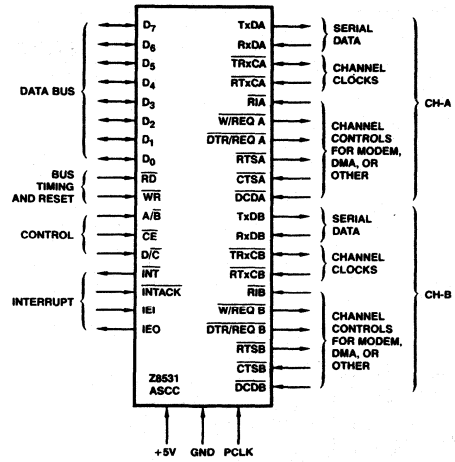
CD005090

Note: Pin 1 is marked for orientation

LOGIC SYMBOL



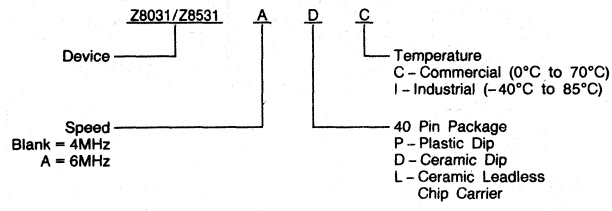
LS001150



LS001160

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations		
Z8031	4MHz 6MHz	DC, PC ADC, APC
Z8531	4MHz 6MHz	DC, PC, ADC, APC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

03818C

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION for Z8031

Pin No.	Name	I/O	Description
9	V_{CC}		+ 5V Power Supply
31	GND		Ground
1-4, 37-40	$AD_0 - AD_7$	I/O	Address/Data Bus (bidirectional, active High, three-state). These multiplexed lines carry register addresses to the ASCC as well as data or control information to and from the ASCC.
35	$\overline{AS}$	I	Address Strobe (active Low). Addresses on $AD_0 - AD_7$ are latched by the rising edge of this signal.
33	$\overline{CS}_0$	I	Chip Select 0 (active Low). This signal is latched concurrently with the addresses on $AD_0 - AD_7$ and must be active for the intended bus transaction to occur.
32	CS_1	I	Chip Select 1 (active High). This second select signal must also be active before the intended bus transaction can occur. CS_1 must remain active throughout the transaction.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear to Send (active Low). If these pins are programmed as Auto Enables, a Low on these inputs enables their respective transmitter. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The ASCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose inputs pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The ASCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
36	$\overline{DS}$	I	Data Strobe (active Low). This signal provides timing for the transfer of data into and out of the ASCC. If AS and DS coincide, this is interpreted as a reset.
16, 24	$\overline{DTR}/\overline{REQA}$, $\overline{DTR}/\overline{REQB}$	O	Data Terminal Ready/Request (active Low).These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is High only if IEI is High and the CPU is not servicing an ASCC interrupt or the ASCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the ASCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the ASCC interrupt daisy chain settles. When $\overline{DS}$ becomes active, the ASCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of $\overline{AS}$.
20	PCLK	I	Clock. This is the master ASCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock, although the frequency of this clock must be at least 90% of the CPU clock frequency for a Z8000. PCLK is a TTL level signal.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{RI}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request to Send (active Low). When the Request to Send (RTS) bit in Write Register 5 (Figure 7) is set, the RTS signal goes Low. When the $\overline{RTS}$ bit is reset and Auto Enable is on, the signal goes High after the transmitter is empty. With Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
34	$R/\overline{W}$	I	Read/Write. This signal specifies whether the operation to be performed is read or a write.
11, 29	$\overline{RIA}$, $\overline{RIB}$	I	Ring Indicator (active Low). These pins can act either as inputs or as part of the crystal oscillator circuit. In normal operation (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Ring Indicator status bits in Read Register 0 (Figure 6) but have no other function.
15, 25	TxDA, TxDB	O	Transmit/Receive Clocks (active Low). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
10, 30	$\overline{W}/\overline{REQA}$, $\overline{W}/\overline{REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the ASCC data rate. The reset state is Wait.

PIN DESCRIPTIONfor Z8531

Pin No.	Name	I/O	Description
9	V_{CC}		+ 5V Power Supply
31	GND		Ground
34	A/$\overline{B}$	I	Channel A/Channel B Select. This signal selects the channel in which the read or write operation occurs.
33	$\overline{CE}$	I	Chip Enable (active Low). This signals selects the ASCC for a read or write operation.
18, 22	$\overline{CTSA}$, $\overline{CTSB}$	I	Clear to Send (active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The ASCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.
32	D/$\overline{C}$	I	Data/Control Select. This signal defines the type of information transferred to or from the ASCC. A High means data is transferred; a Low indicates a command.
19, 21	$\overline{DCDA}$, $\overline{DCDB}$	I	Data Carrier Detect (active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose inputs pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The ASCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
1-4, 37-40	$D_0 - D_7$	I/O	Data Bus (bidirectional, three-state). These lines carry data and commands to and from the ASCC.
16, 24	$\overline{DTR}$/$\overline{REQA}$, $\overline{DTR}$/$\overline{REQB}$	O	Data Terminal Ready/Request (active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.
7	IEI	I	Interrupt Enable In (active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.
6	IEO	O	Interrupt Enable Out (active High). IEO is High only if IEI is High and the CPU is not servicing an ASCC interrupt or the ASCC is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.
5	$\overline{INT}$	O	Interrupt Request (open-drain, active Low). This signal is activated when the ASCC requests an interrupt.
8	$\overline{INTACK}$	I	Interrupt Acknowledge (active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the ASCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the ASCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of PCLK.
20	PCLK	I	Clock. This is the master ASCC clock used to synchronize internal signals. PCLK is a TTL level signal.
36	$\overline{RD}$	I	Read (active Low). This signal indicates a read operation and when the ASCC is selected, enables the ASCC's bus drivers. During the interrupt acknowledge cycle, this signal gates the interrupt vector onto the bus if the ASCC is the highest priority device requesting an interrupt.
13, 27	RxDA, RxDB	I	Receive Data (active High). These input signals receive serial data at standard TTL levels.
12, 28	$\overline{RTxCA}$, $\overline{RTxCB}$	I	Receive/Transmit Clocks (active Low). These pins can be programmed in several different modes of operation. In each channel $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock of the digital phase-locked loop. These pins can also be programmed for use with the respective $\overline{RI}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.
17, 23	$\overline{RTSA}$, $\overline{RTSB}$	O	Request to Send (active Low). When the Request to Send (RTS) bit in Write Register 5 (Figure 7) is set, the $\overline{RTS}$ signal goes Low. When the RTS bit is reset in the asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. With Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
15, 25	TxDA, TxDB	O	Transmit Data (active High). These output signals transmit serial data at standard TTL levels.
14, 26	$\overline{TRxCA}$, $\overline{TRxCB}$	I/O	Transmit/Receive Clocks (active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.
35	$\overline{WR}$	I	Write (active Low). When the ASCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.
10, 30	$\overline{W/REQA}$, $\overline{W/REQB}$	O	Wait/Request (open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the ASCC data rate. The reset state is Wait.

ARCHITECTURE

The ASCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to the Z8000 CPU (Z8031) or to a non-multiplexed CPU bus (Z8531). Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 1).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modern control.

The register set for each channel includes ten control (write) registers, two synchronous character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write-only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the Interrupt Pending bits (A only).

The registers for each channel are designated as follows: WR0–WR15 – Write Registers 0 through 15. RR0–RR3, RR10, RR12, RR13, RR15 – Read Registers 0 through 3, 10, 12, 13, 15.

The following table lists the functions assigned to each read or write register. The ASCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

DATA PATH

The transmit and receive data path illustrated in Figure 2 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in asynchronous modes also determines the data path).

The transmitter has an 8-bit transmit data buffer register loaded from the internal data bus and an 11-bit transmit shift register that can be loaded from the transmit data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

TABLE 1. READ AND WRITE REGISTER FUNCTIONS

READ REGISTER FUNCTIONS	
RR0	Transmit/Receive buffer status and External status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/Status interrupt information

WRITE REGISTER FUNCTIONS	
WR0	CRC initialize, initialization commands for the various modes, shift right/shift left command
WR1	Transmit/Receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync characters or SDLC address field
WR7	Sync character or SDLC flag
WR8	Transmit buffer
WR9	Master interrupt control and reset (accessed through either channel)
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits
WR15	External/Status interrupt control

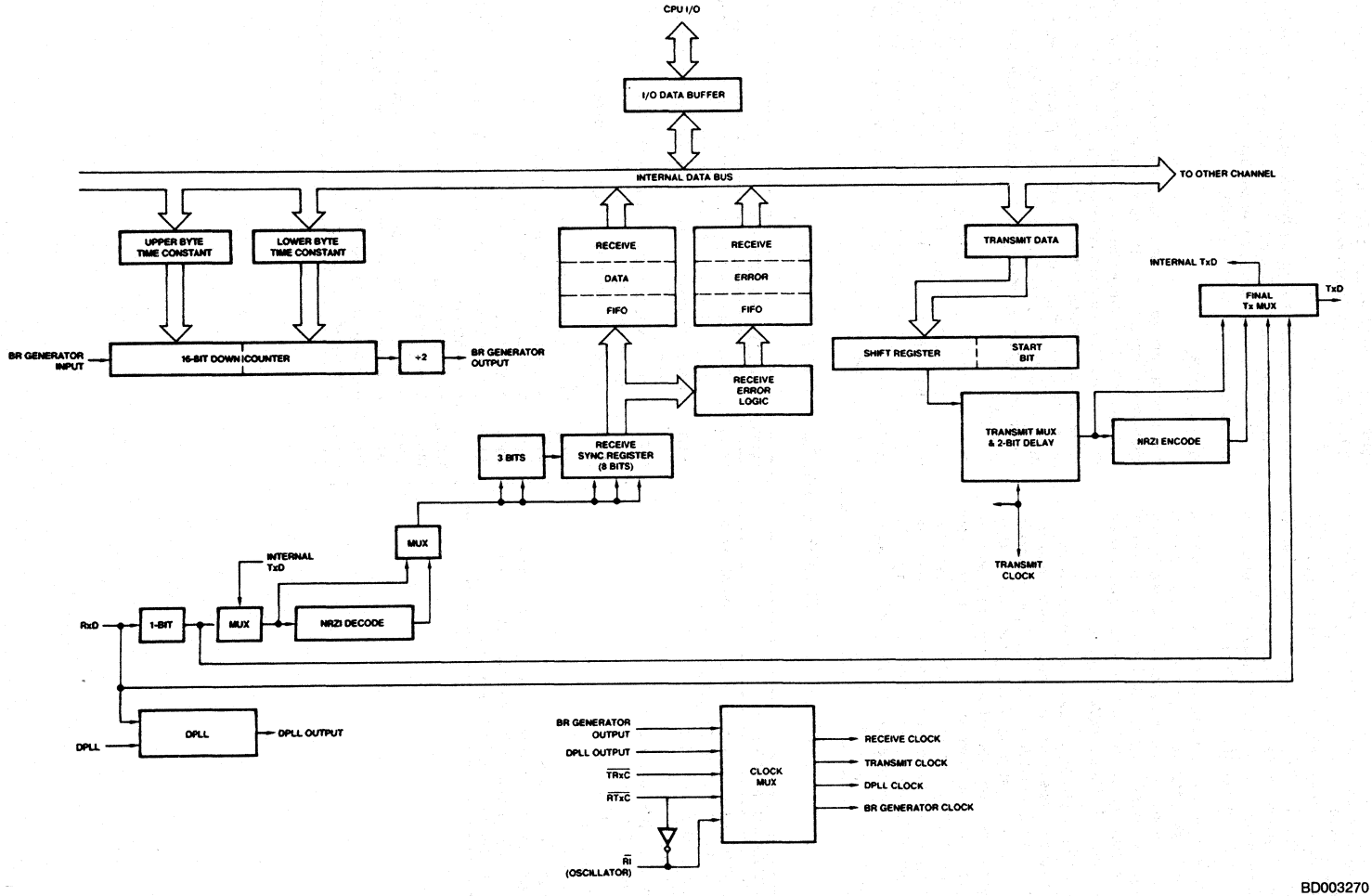


Figure 2. Data Path

Refer to page 7-1 for Essential Information on Military Devices

DETAILED DESCRIPTION

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, it interacts with the CPU and provides vectored interrupts and handshaking signals.

DATA COMMUNICATIONS CAPABILITIES

The ASCC provides two independent full-duplex channels programmable for use in any common asynchronous data-communication protocol. Figure 3 and the following description briefly detail this protocol.

Asynchronous Modes

Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 14). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The ASCC does not require symmetric transmit and receive clock signals – a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs.

BAUD RATE GENERATOR

Each channel in the ASCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching zero, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the digital phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate. (The baud rate is in bits/second and the BR clock period is in seconds.)

$$\text{baud rate} = \frac{1}{2\ (\text{time constant} + 2) \times (\text{BR clock period})}$$

Time-Constant Values for Standard Baud Rates at BR Clock = 3.9936MHz

Rate (Baud)	Time Constant (decimal notation)	Error
19200	102	–
9600	206	–
7200	275	0.12%
4800	414	–
3600	553	0.06%
2400	830	–
2000	996	0.04%
1800	1107	0.03%
1200	1662	–
600	3326	–
300	6654	–
150	13310	–
134.5	14844	0.0007%
110	18151	0.0015%
75	26622	–
50	39934	–

DIGITAL PHASE-LOCKED LOOP

The ASCC contains a digital phase-locked loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the ASCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming data stream for edges (either 1/0 or 0/1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15/16 counting transition.

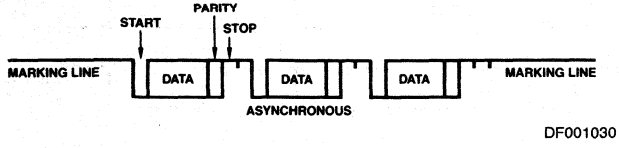


Figure 3. ASCC Protocols

The 32X clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the ASCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

DATA ENCODING

The ASCC may be programmed to encode and decode the serial data in four different ways. In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM_1 (more properly, bi-phase mark) a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM_0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ASCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0/1, the bit is a 0. If the transition is 1/0, the bit is a 1.

AUTO ECHO AND LOCAL LOOPBACK

The ASCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes, but works in synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The ASCC is also capable of local loopback. In this mode, TxD is RxD just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works with NRZ, NRZI or FM coding of the data stream.

I/O INTERFACE CAPABILITIES

The ASCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

POLLING

All interrupts are disabled. Three status registers in the ASCC are automatically updated whenever any function is performed. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

INTERRUPTS

When an ASCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 6 and 7).

To speed interrupt response time, the ASCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the ASCC (Transmit, Receive and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write-only.

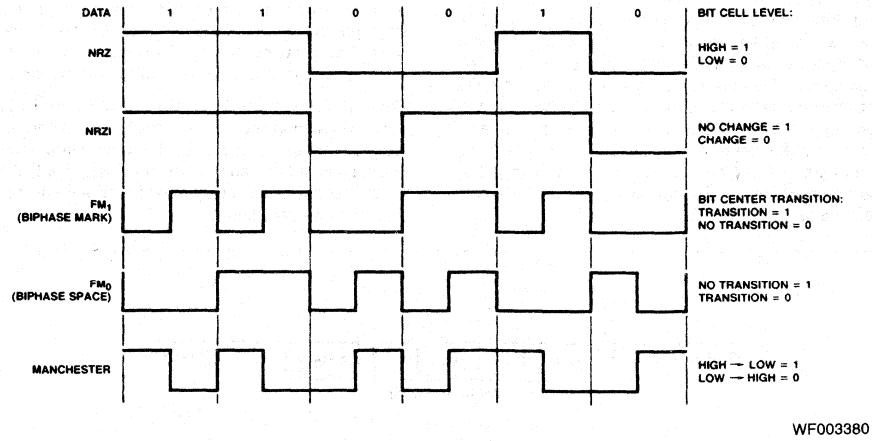


Figure 4. Data Encoding Methods

The other two bits are related to the Z-Bus interrupt priority chain (Figure 5). As a Z-Bus peripheral, the ASCC may request an interrupt only when no higher-priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the ASCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the INT output is pulled Low, requesting an interrupt. In the ASCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ASCC and external to the ASCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the ASCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive and External/Status interrupts. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive condition.
- Interrupt on all Receive Characters or Special Receive condition.
- Interrupt on Special Receive condition only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in asynchronous mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt-Acknowledge cycle. In Interrupt on first Receive Character, an interrupt can occur from Special Receive conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{RI}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (asynchronous mode).

CPU/DMA BLOCK TRANSFER

The ASCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{WAIT/REQUEST}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT/REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the ASCC $\overline{REQUEST}$ output indicates that the ASCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ line indicates that the ASCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR/REQUEST}$ line allows full-duplex operation under DMA control.

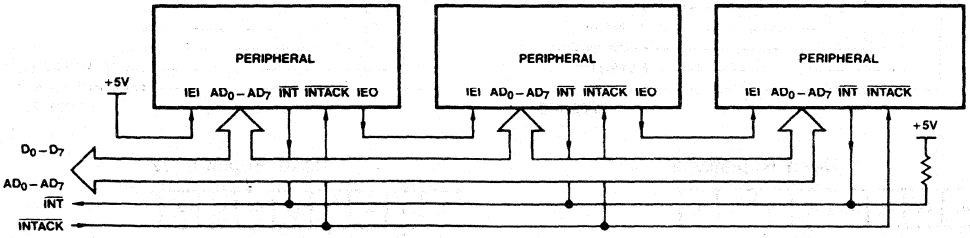


Figure 5. Z-Bus Interrupt Schedule

PROGRAMMING INFORMATION (Z8031)

The Z8031 contains 11 write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

All of the registers in the Z8031 are directly addressable. How the Z8031 decodes the address placed on the address/data bus at the beginning of a Read or Write cycle is controlled by a command issued in WR0B. In the shift right mode, the channel select $\overline{A/B}$ is taken from AD_0 and the state of AD_5 is ignored. In the shift left mode, $\overline{A/B}$ is taken from AD_5 and the state of AD_0 is ignored. AD_7 and AD_6 are always ignored as address bits and the register address itself occupies AD_4–AD_1.

The system program first issues a series of commands to initialize the basic mode of operation. For example, the character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

PROGRAMMING INFORMATION (Z8531)

The Z8531, register addressing is direct for the data registers only, which are selected by a High on the D/$\overline{C}$ pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word

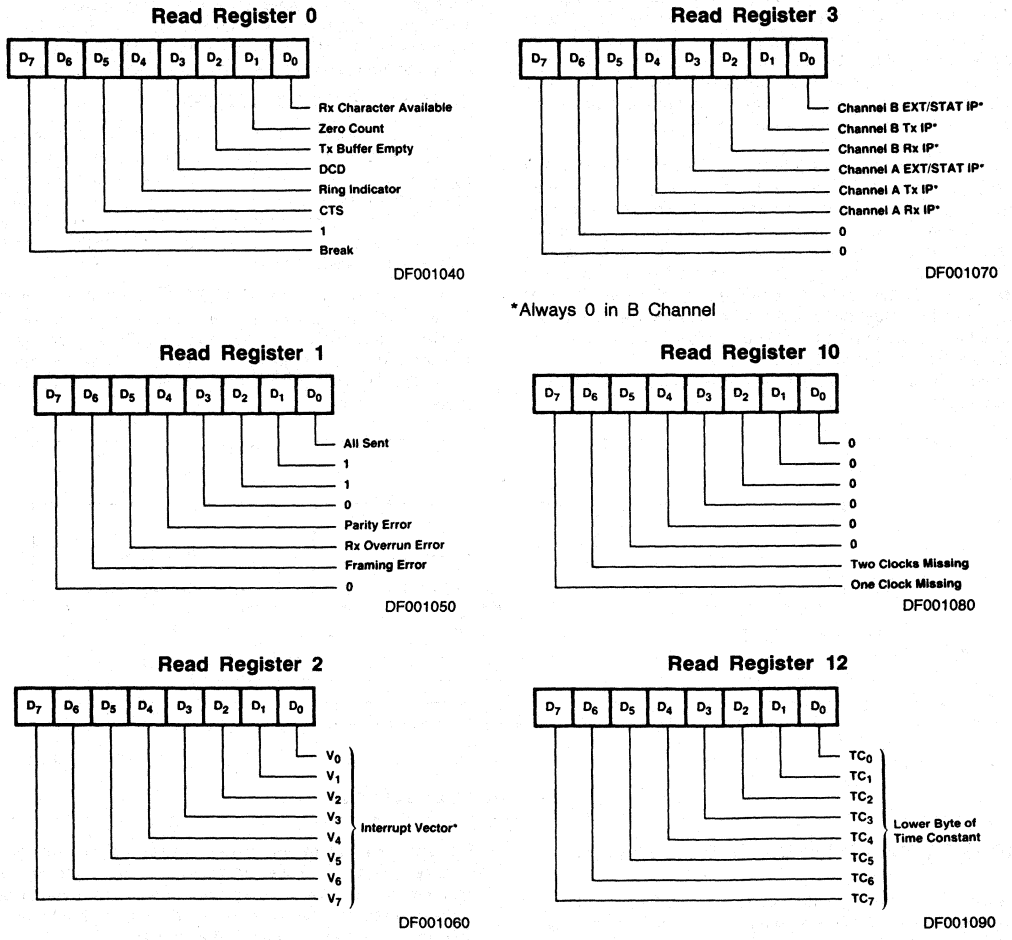
for the selected register, and if the second operation is read, the selected read register is accessed. All of the registers in the Z8531, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

The system program first issues a series of commands to initialize the basic mode of operation. For example, the character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

READ REGISTERS

The ASCC contains 8 read registers (actually 9, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10 and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 6 shows the formats for each read register.

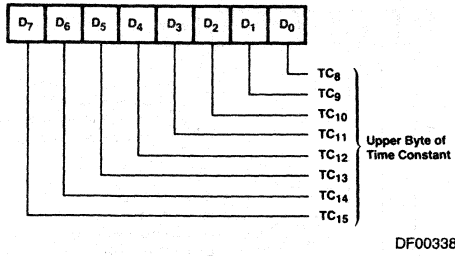
The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).



*Modified in B Channel

Figure 6. Read Register Bit Functions

Read Register 13



Read Register 15

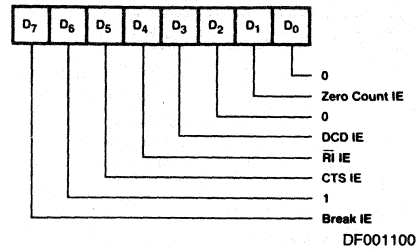
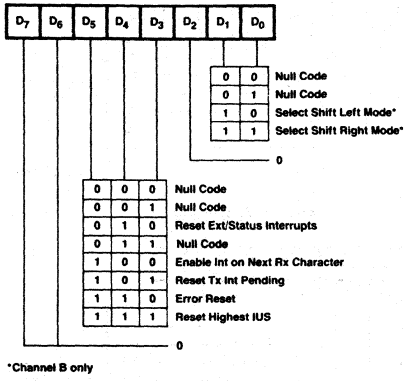


Figure 6. Read Register Bit Functions (Cont.)

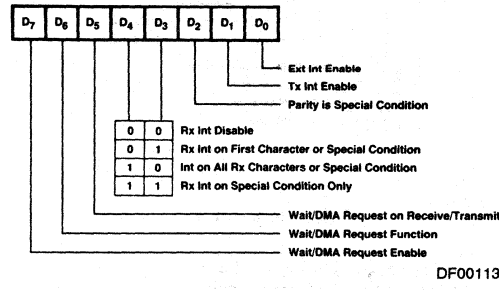
WRITE REGISTERS

The ASCC contains 11 write registers (12 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 7 shows the format of each write register.

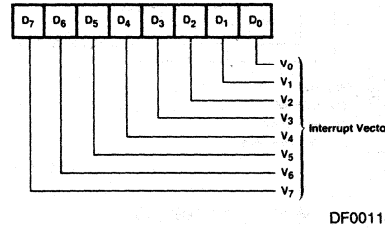
Write Register 0 (Z8031)



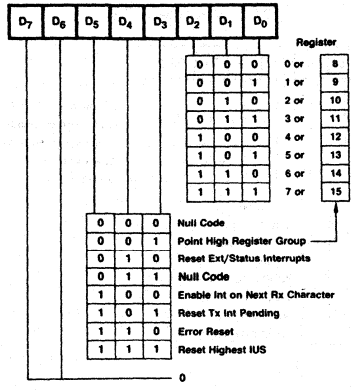
Write Register 1



Write Register 2



Write Register 0 (Z8531)



Write Register 3

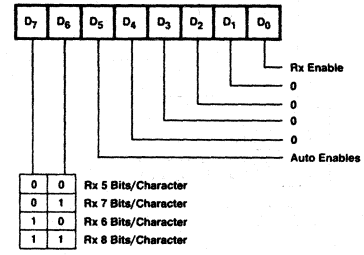
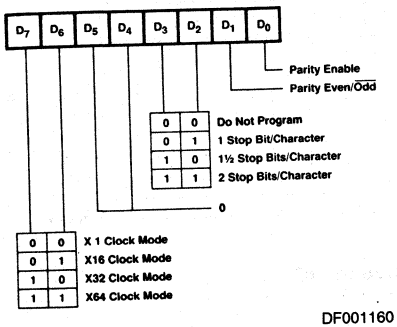


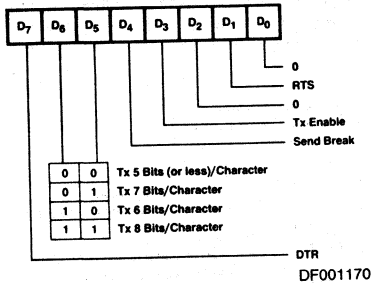
Figure 7. Write Register Bit Functions

Write Register 4



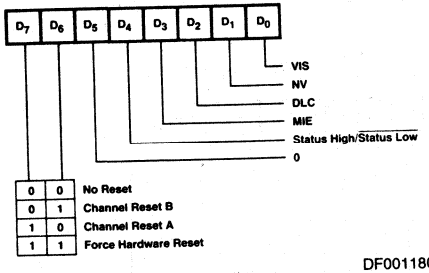
DF001160

Write Register 5



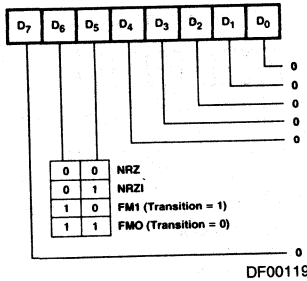
DF001170

Write Register 9



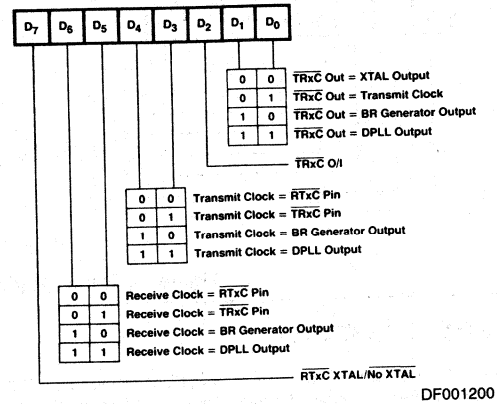
DF001180

Write Register 10



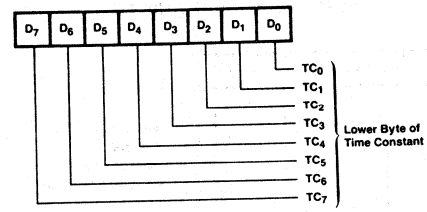
DF001190

Write Register 11



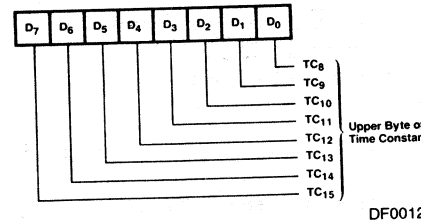
DF001200

Write Register 12



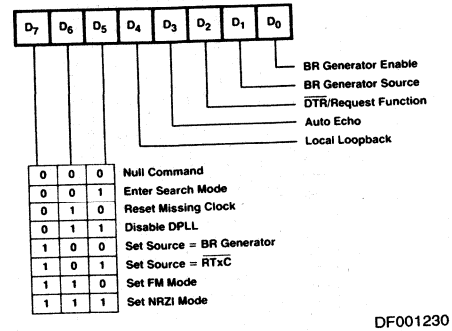
DF001210

Write Register 13



DF001220

Write Register 14



DF001230

Figure 7. Write Register Bit Functions (Cont.)

Write Register 15

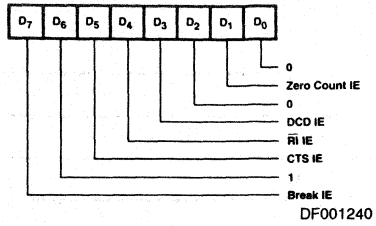


Figure 7. Write Register Bit Functions (Cont.)

Z8031 TIMING

The ASCC generates internal control signals from $\overline{AS}$ and $\overline{DS}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{AS}$ and $\overline{DS}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ASCC to the falling edge of $\overline{DS}$ in the second transaction involving the ASCC. This time must be at least 6 PCLK cycles plus 200ns.

READ CYCLE TIMING

Figure 8 illustrates read cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/W must be High to indicate a read cycle. CS_1 must also be High for the read cycle to occur. The data bus drivers in the ASCC are then enabled while $\overline{DS}$ is Low.

WRITE CYCLE TIMING

Figure 9 illustrates write cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be Low to indicate a write cycle. CS_1 must be High for the write cycle to occur. $\overline{DS}$ Low strobes the data into the ASCC.

INTERRUPT ACKNOWLEDGE CYCLE TIMING

Figure 10 illustrates interrupt acknowledge cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. However, if $\overline{INTACK}$ is Low, the address and $\overline{CS_0}$ are ignored. The state of R/$\overline{W}$ and CS_1 are also ignored for the duration of the interrupt acknowledge cycle. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ASCC and IEI is High when DS falls, the acknowledge cycle was intended for the ASCC. In this case, the ASCC may be programmed to respond to $\overline{DS}$ Low by placing its interrupt vector on AD_0–AD_7. It then sets the appropriate interrupt-under-service latch internally.

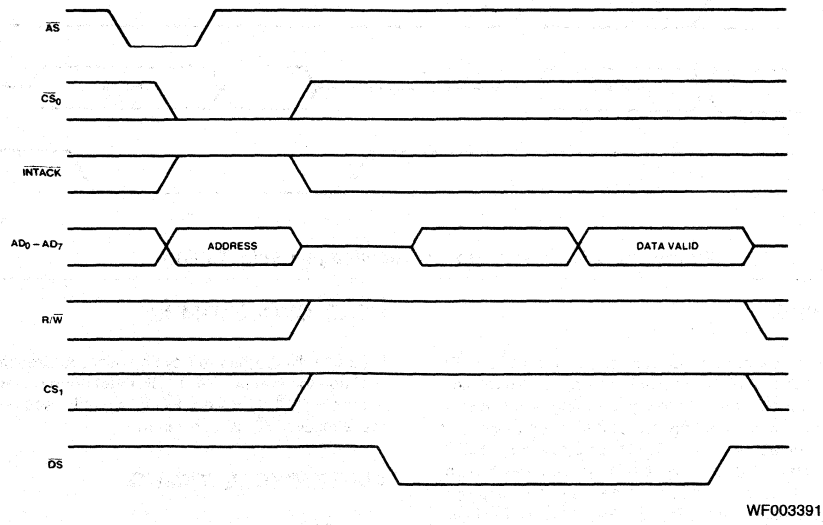


Figure 8. Z8031 Read Cycle Timing

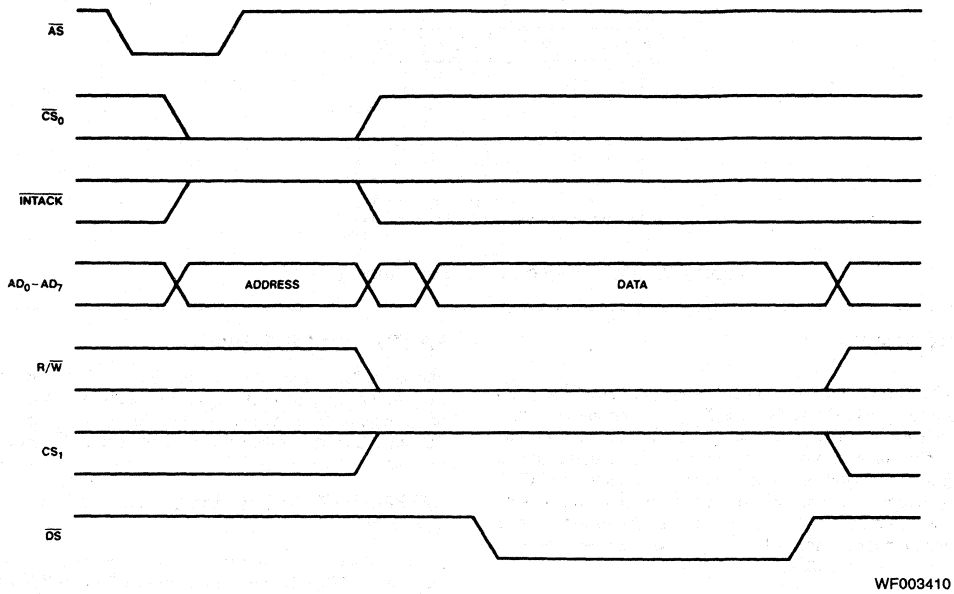


Figure 9. Z8031 Write Cycle Timing

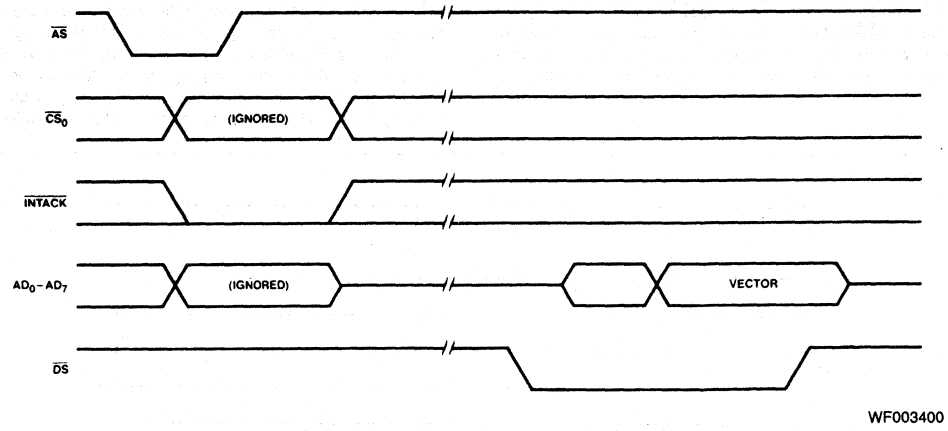


Figure 10. Z8031 Interrupt Acknowledge Cycle Timing

Z8531 TIMING

The ASCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ASCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the ASCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the ASCC. This time must be at least 6 PCLK cycles plus 200ns.

READ CYCLE TIMING

Figure 11 illustrates read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

WRITE CYCLE TIMING

Figure 12 illustrates write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened.

INTERRUPT ACKNOWLEDGE CYCLE TIMING

Figure 13 illustrates interrupt acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ASCC and IEI is High when $\overline{RD}$ falls, the acknowledge cycle is intended for the ASCC. In this case, the ASCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on D_0–D_7; it then sets the appropriate interrupt-under-service latch internally.

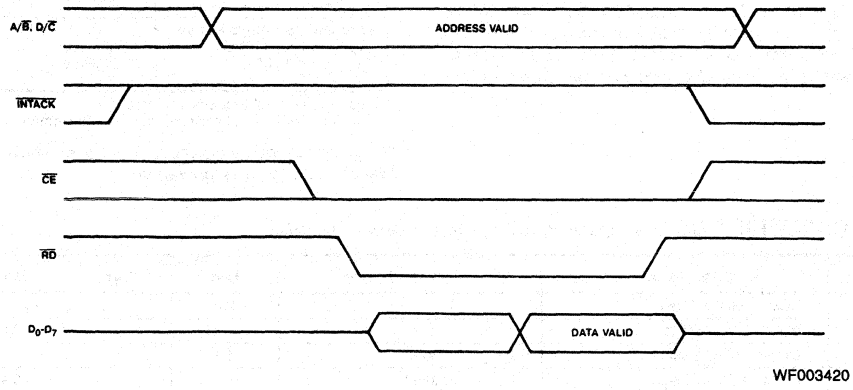


Figure 11. Z8531 Read Cycle Timing

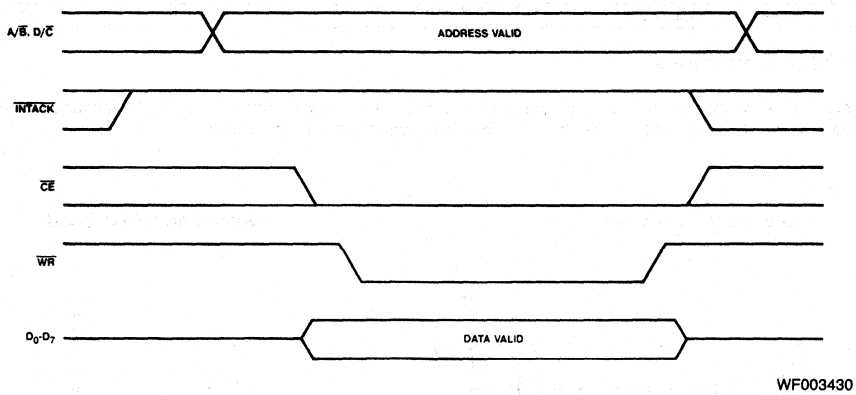


Figure 12. Z8531 Write Cycle Timing

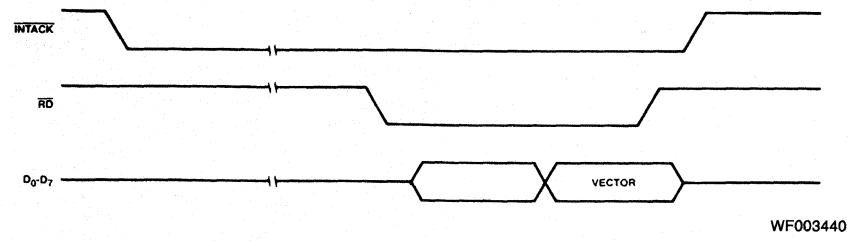


Figure 13. Z8531 Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin Relative to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 1.8W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	4MHz	6MHz
Commercial Operating Range T_A = 0 to 70°C V_{CC} = 5V ±5% V_{SS} = 0V	Z8031DC Z8031PC Z8531DC Z8531PC	Z8031ADC Z8031APC Z8531ADC Z8531APC
Industrial Operating Range T_A = −40 to +85°C V_{CC} = 5V ±10% V_{SS} = 0V	Z8031DI Z8531DI	

Notes: T_A denotes ambient temperature.
Add suffix B to indicate burn-in requirement.

Operating ranges define those limits over which the functionality of the device is guaranteed.

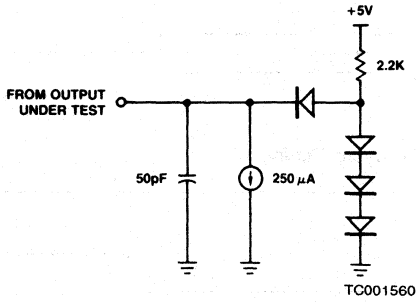
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IH}	Input HIGH Voltage		2.0		V_{CC} + 0.3	V
V_{IL}	Input LOW Voltage		−0.3		0.8	V
V_{OH}	Output HIGH Voltage	I_{OH} = −250µA	2.4			V
V_{OL}	Output LOW Voltage	I_{OL} = +2.0mA			0.4	V
I_{IL}	Input Leakage	0.4 ≤ V_{IN} +2.4V			±10.0	µA
I_{OL}	Output Leakage	0.4 ≤ V_{OUT} ≤ +2.4V			±10.0	µA
I_{CC}	V_{CC} Supply Current				250	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.			10	pF
C_{OUT}	Output Capacitance				15	pF
$C_{I/O}$	Bidirectional Capacitance				20	pF

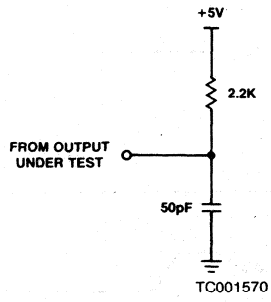
Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

Standard Test Load



Open Drain Test Load



SWITCHING CHARACTERISTICS over operating range unless otherwise specified
SYSTEM TIMING
Z8531

Number	Parameters	Description	4MHz Min	4MHz Max	6MHz Min	6MHz Max	Units
1	TdRXC(REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 2)	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay (Notes 1, 2)	8	12	8	12	TcPC
3	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2)	10	16	10	16	TcPC
4	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3)	5	8	5	8	TcPC
5	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3)	5	8	5	8	TcPC
6	TdTXC(DRQ)	$\overline{TxC}$ ↓ $\overline{DTR/REQ}$ Valid Delay (Note 3)	4	7	4	7	TcPC
7	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3)	6	10	6	10	TcPC
8	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1)	2	6	2	6	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

*Timings are preliminary and subject to change.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
SYSTEM TIMING
Z8031

Number	Parameters	Description	4MHz Min	4MHz Max	6MHz Min	6MHz Max	Units
1	TdRXC(REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay	8	12	8	12	TcPC
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay	8	12	8	12	TcPC
3	TdRXC(INT)	$\overline{RxC}$ ↑ $\overline{INT}$ Valid Delay	8	12	8	12	TcPC
			+2	+3	+2	+3	$\overline{AS}$
4	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay	5	8	5	8	TcPC
5	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay	5	8	5	8	TcPC
6	TdTXC(DRQ)	$\overline{TxC}$ ↓ $\overline{DTR/REQ}$ Valid Delay	4	7	4	7	TcPC
7	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay	4	6	4	6	TcPC
			+2	+3	+2	+3	$\overline{AS}$
8	TdEXT(INT)	$\overline{DCD}$, $\overline{RI}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay	2	3	2	3	TcPC

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

*Timings are preliminary and subject to change.

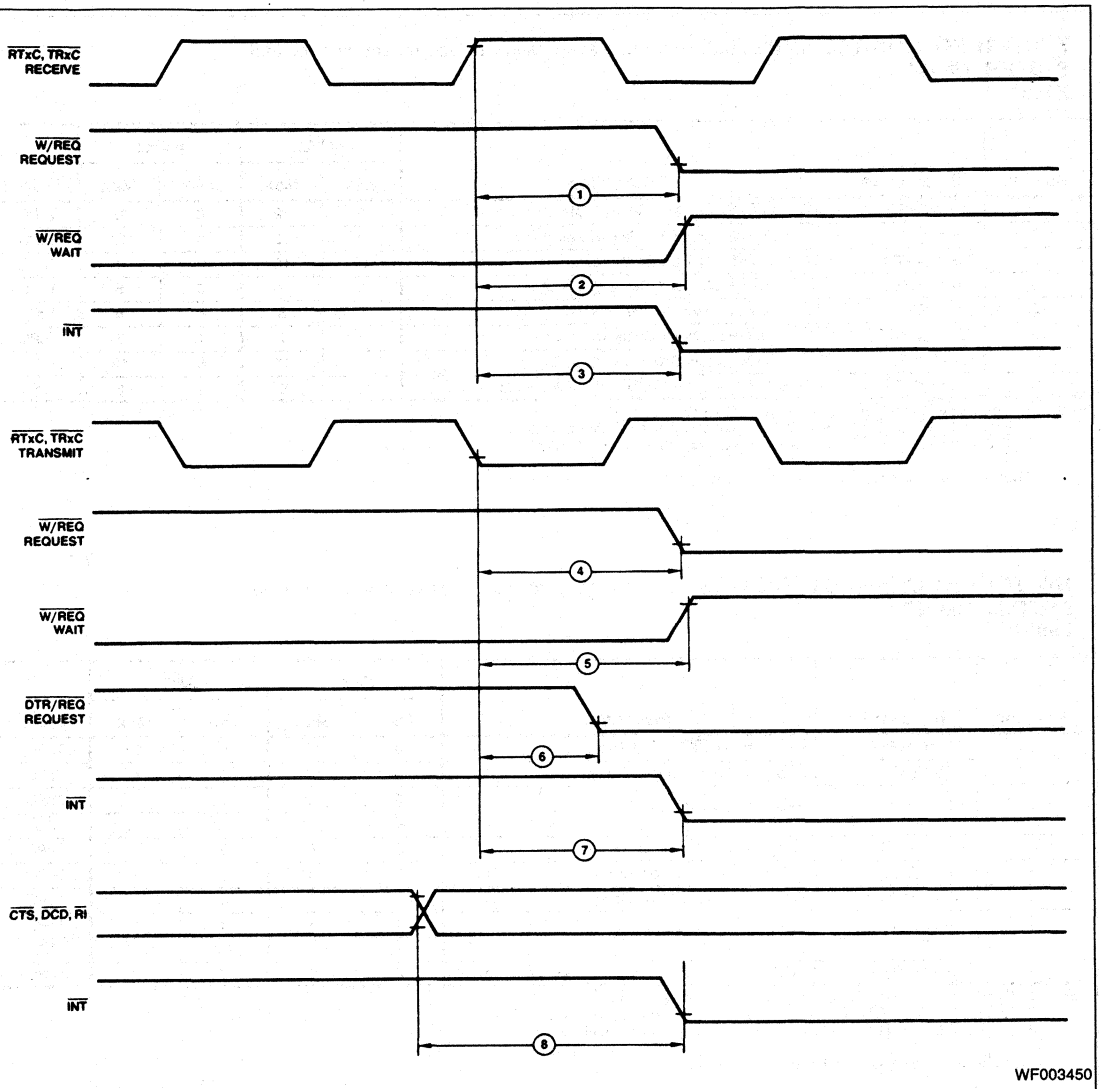


Figure 14. System Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
GENERAL TIMING (See Figure 15)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TdPC(REQ)	PCLK ↓ to W/$\overline{REQ}$ Valid Delay		250		250	ns
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350	ns
3	TsRXC(PC)	$\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4)	80		80		ns
4	TsRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Setup Time (X1 Mode)(Note 1)	0		0		ns
5	ThRXD(RXCr)	RxD to $\overline{RxC}$ ↑ Hold Time (X1 Mode)(Note 1)	150		150		ns
6	TsRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Setup Time (X1 Mode)(Notes 1, 5)	0		0		ns
7	ThRXD(RXCf)	RxD to $\overline{RxC}$ ↓ Hold Time (X1 Mode)(Notes 1, 5)	150		150		ns
8	TsTXC(PC)	$\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4)	0		0		ns
9	TdTXCf(TXD)	$\overline{TxC}$ ↓ to TxD Delay (X1 Mode)(Note 2)		300		300	ns
10	TdTXCr(TXD)	$\overline{TxC}$ ↑ to TxD Delay (X1 Mode)(Notes 2, 5)		300		300	ns
11	TdTXD(TRX)	TxD to $\overline{TRxC}$ Delay (Send Clock Echo)		200		200	ns
12	TwRTXh	$\overline{RTxC}$ High Width	180		180		ns
13	TwRTXl	$\overline{RTxC}$ Low Width	180		180		ns
14	TcRTX	$\overline{RTxC}$ Cycle Time	400		400		ns
15	TcRTXX	Crystal Oscillator Period (Note 3)	250	1000	250	1000	ns
16	TwTRXh	$\overline{TRxC}$ High Width	180		180		ns
17	TwTRXl	$\overline{TRxC}$ Low Width	180		180		ns
18	TcTRX	$\overline{TRxC}$ Cycle Time	400		400		ns
19	TwEXT	$\overline{DCD}$ or $\overline{CTS}$ Pulse Width	200		200		ns

Notes: 1. RxC is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. TxC is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{RI}$ have 30pF capacitors to the ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{RxC}$ and $\overline{PCLK}$ or $\overline{TxC}$ and $\overline{PCLK}$ is required.
5. Parameter applies only to FM encoding/decoding.

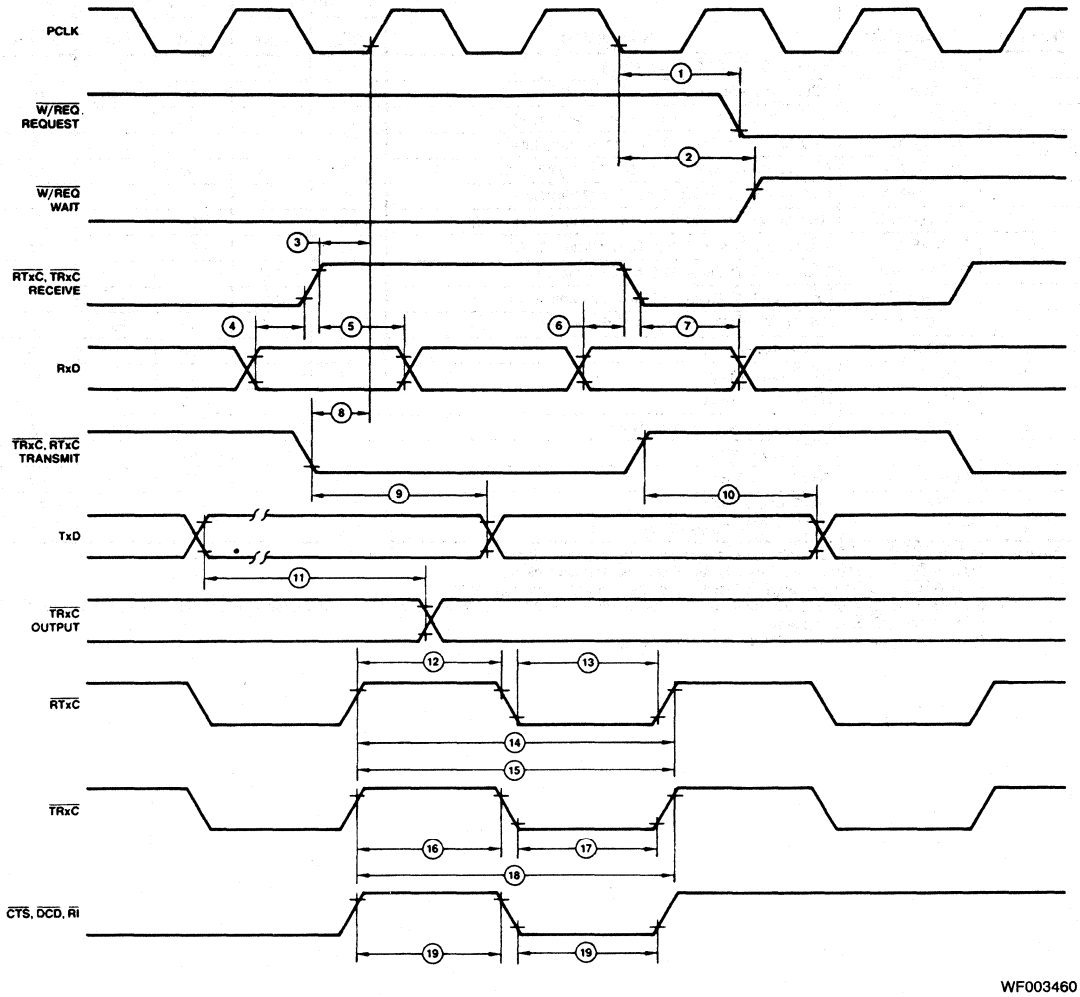


Figure 15. General Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8031 READ AND WRITE TIMING (see Figure 16)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TwAS	$\overline{AS}$ LOW Width	70		50		ns
2	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		ns
3	TsCS0(AS)	$\overline{CS}_0$ to $\overline{AS}$ ↑ Setup Time (Note 1)	0		0		ns
4	ThCS0(AS)	$\overline{CS}_0$ to $\overline{AS}$ ↑ Hold Time (Note 1)	60		40		ns
5	TsCS1(DS)	CS_1 to $\overline{DS}$ ↓ Setup Time (Note 1)	100		80		ns
6	ThCS1(DS)	CS_1 to $\overline{DS}$ ↑ Hold Time (Note 1)	55		40		ns
7	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	0		0		ns
8	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		250		ns
9	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Setup Time	100		80		ns
10	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		ns
11	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Setup Time	0		0		ns
12	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	60		40		ns
13	TwDSl	$\overline{DS}$ LOW Width	390		250		ns
14	TrC	Valid Access Recovery Time (Note 2)	6TcPC +200		6TcPC +130		ns
15	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time (Note 1)	30		10		ns
16	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time (Note 1)	50		30		ns
17	TsDW(DS)	Write Data to $\overline{DS}$ ↓ Setup Time	30		20		ns
18	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		20		ns
19	TdDS(DA)	$\overline{DS}$ ↓ to Data Active Delay	0		0		ns
20	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		ns
21	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	ns
22	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Valid Delay		520		335	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Parameter applies only between transactions involving the 8030.

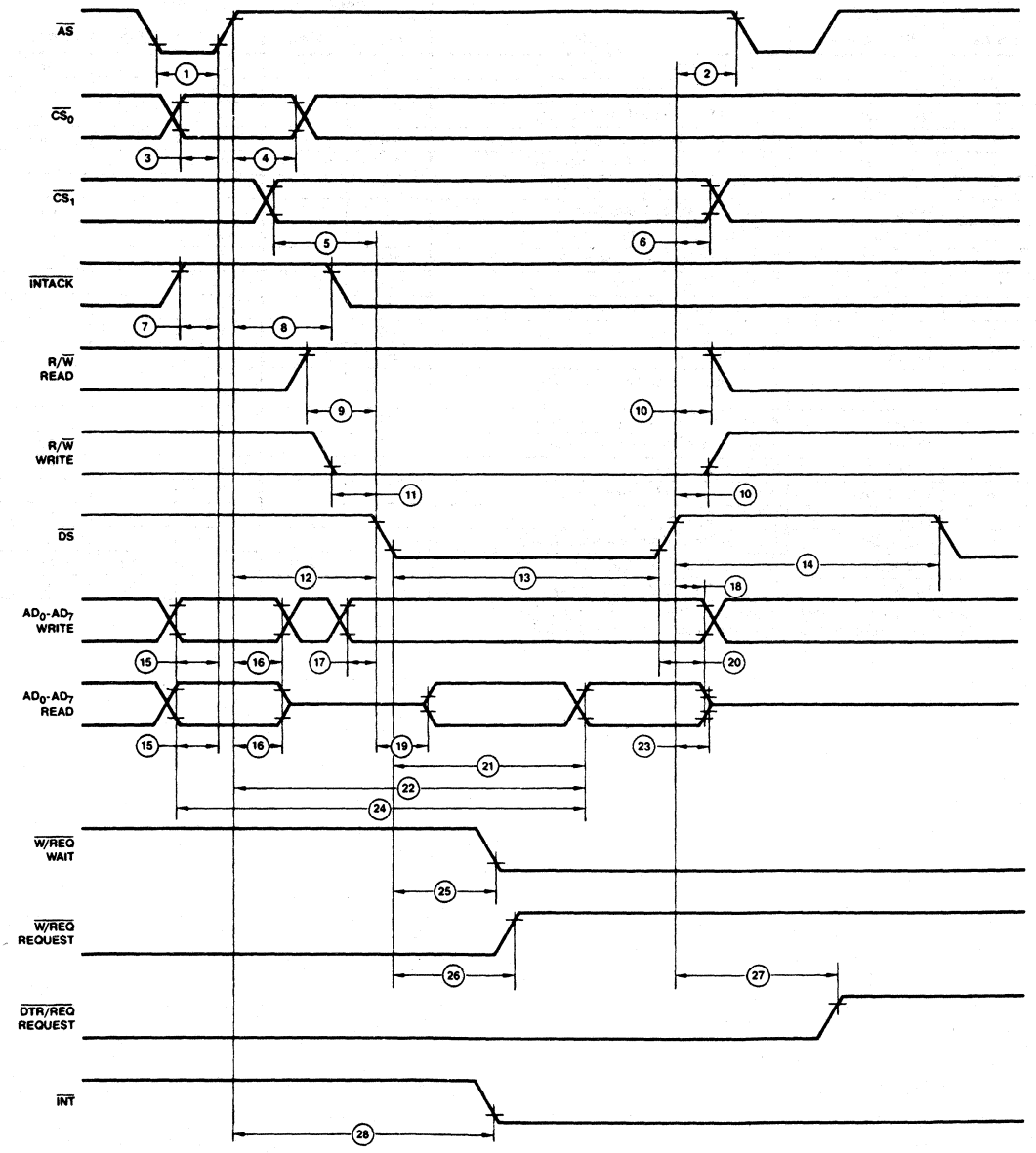


Figure 16. Z8031 Read and Write Timing

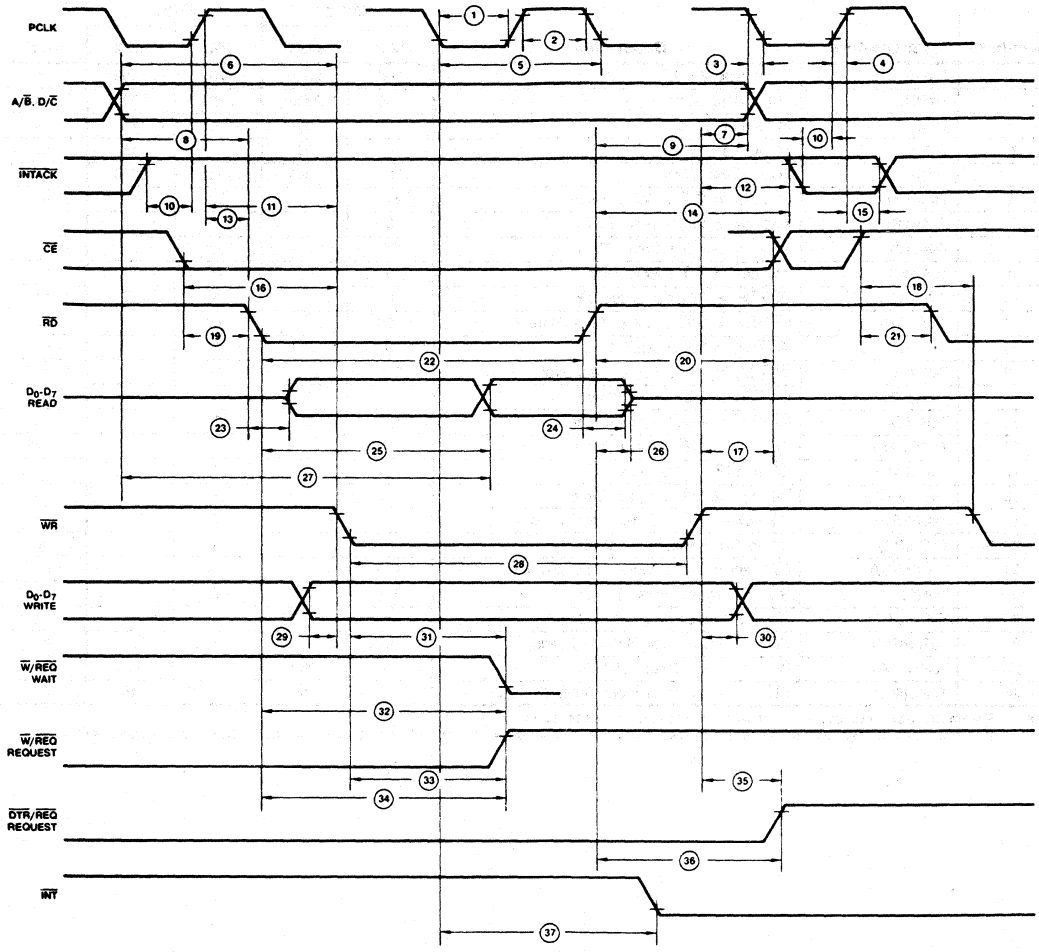
SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8531 READ AND WRITE TIMING (see Figure 20)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TwPCl	PCLK Low Width	105	2000	70	1000	ns
2	TwPCh	PCLK High Width	105	2000	70	1000	ns
3	TfPC	PCLK Fall Time		20		10	ns
4	TrPC	PCLK Rise Time		20		15	ns
5	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		80		ns
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		80		ns
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TslA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	0		0		ns
11	TslAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1)	200		200		ns
12	ThlA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
13	TslAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	200		200		ns
14	ThlA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
15	ThlA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		ns
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		ns
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		70		ns
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		ns
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		ns
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note 1)	100		70		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	390		250		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180	ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70		45	ns

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ±0.5V change in the output with a maximum D.C. load and minimum A.C. load.

03818C

Refer to page 7-1 for Essential Information on Military Devices



WF003510

Figure 20. Z8531 Read and Write Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z8031 INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING (see Figures 17, 18, 19)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
23	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay (Note 3)		70		45	ns
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		570		420	ns
25	TdDS(W)	$\overline{DS}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
26	TdDSf(REQ)	$\overline{DS}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
27	TdDSr(REQ)	$\overline{DS}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
28	TdAS(INT)	$\overline{AS}$ ↑ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
29	TdAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
30	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		250		ns
31	TdDSA(DR)	$\overline{DS}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180	ns
32	TsIEI(DSA)	IEI to $\overline{DS}$ ↓ (Acknowledge) Setup Time	120		100		ns
33	ThIEI(DSA)	IEI to $\overline{DS}$ ↑ (Acknowledge) Hold Time	0		0		ns
34	TdIEI(IEO)	IEI to IEO Delay		120		100	ns
35	TdAS(IEO)	$\overline{AS}$ ↑ to IEO Delay (Note 6)		250		250	ns
36	TdDSA(INT)	$\overline{DS}$ ↓ (Acknowledge) to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
37	TdDS(ASQ)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay for No Reset	30		15		ns
38	TdASQ(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay for No Reset	30		30		ns
39	TwRES	$\overline{AS}$ and $\overline{DS}$ Coincident Low for Reset (Note 7)	250		250		ns
40	TwPCl	PCLK Low Width	105	2000	70	1000	ns
41	TwPCh	PCLK High Width	105	2000	70	1000	ns
42	TcPC	PCLK Cycle Time	250	4000	165	2000	ns
43	TrPC	PCLK Rise Time		20		15	ns
44	TfPC	PCLK Fall Time		20		10	ns

Notes: 3. Float delay is defined as the time required for a ±0.5V change in the output with a maximum D.C. load and minimum A.C. load.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any 8031 in the daisy chain, TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain, TsIEI(DSA) for the 8031, and TdIEI(IEO) for each device separating them in the daisy chain.
6. Parameter applies only to a 8031 pulling $\overline{INT}$ Low at the beginning of the Interrupt Acknowledge transaction.
7. Internal circuitry allows for the reset provided by the Z8 to be recognized as a reset by the 8031.

*Timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic.

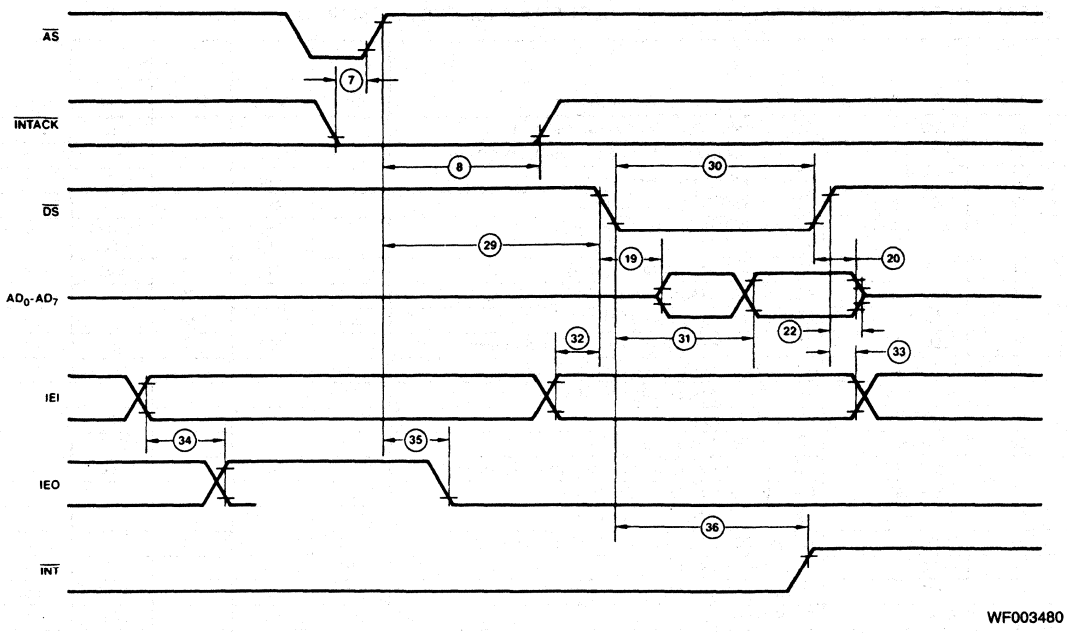


Figure 17. Z8031 Interrupt Acknowledge Timing

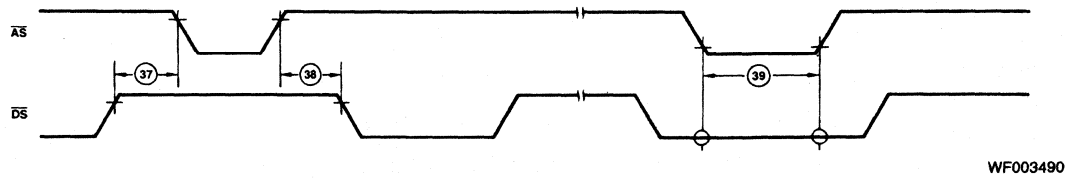


Figure 18. Z8031 Reset Timing

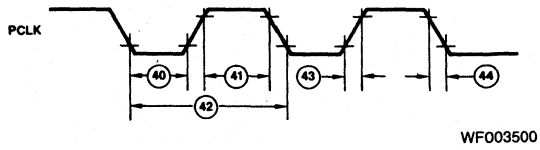


Figure 19. Z8031 Cycle Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8531 INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING (see Figures 21, 22, 23)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		590		420	ns
28	TwWRI	$\overline{WR}$ Low Width	390		250		ns
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	0		0		ns
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		ns
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay (Note 4)		240		200	ns
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200	ns
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
36	TdRDr(REQ)	$\overline{RD}$↑ to $\overline{DTR/REQ}$ Not Valid Delay		5TcPC +300		5TcPC +250	ns
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay (Note 4)		500		500	ns
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	285		250		ns
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		190		180	ns
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		100		ns
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		ns
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100	ns
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250	ns
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 4)		500		500	ns
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		15		ns
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		30		ns
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		250		ns
49	Trc	Valid Access Recovery Time (Note 3)	6TcPC +200		6TcPC +130		ns

Notes: 3. Parameter applies only between transactions involving the ASCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.

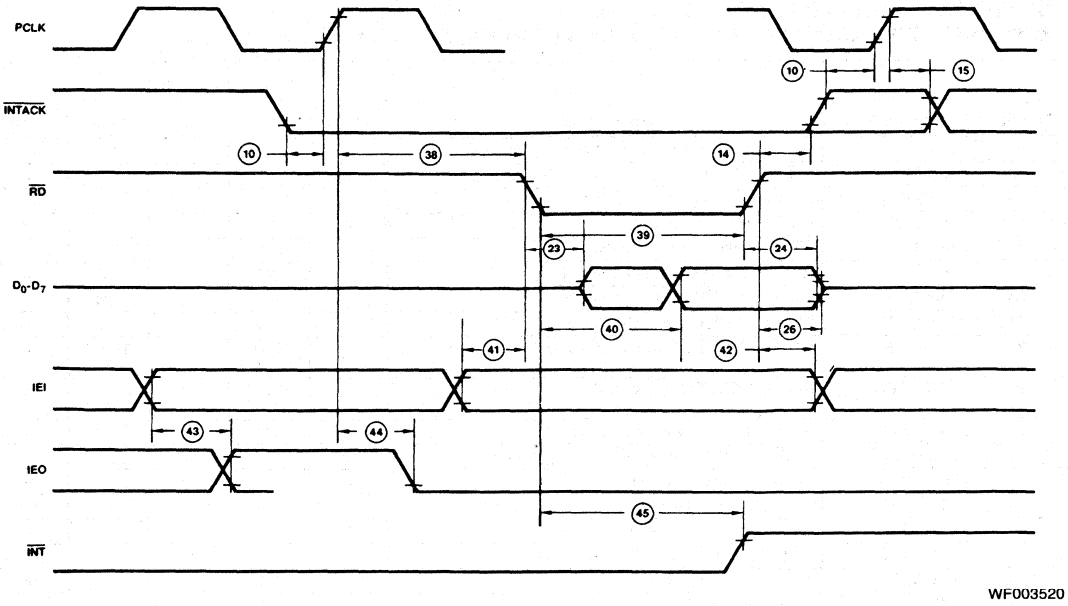


Figure 21. Z8531 Interrupt Acknowledge Timing

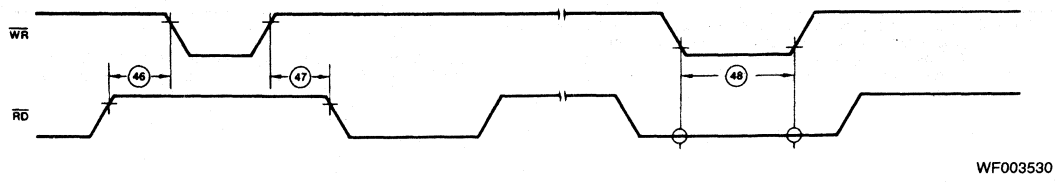


Figure 22. Z8531 Reset Timing

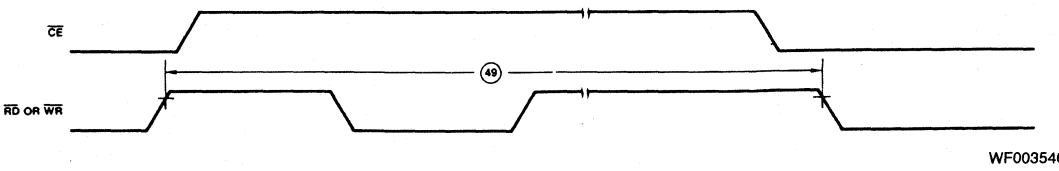


Figure 23. Z8531 Cycle Timing

Z8036/8536

Counter/Timer and Parallel I/O Unit

DISTINCTIVE CHARACTERISTICS

- Two independent 8-bit, double-buffered, bidirectional I/O ports plus a 4-bit special purpose I/O port. I/O ports feature programmable polarity, programmable direction (Bit mode), "pulse catchers" and programmable open-drain outputs.
- Four handshake modes, including 3-wire (like the IEEE-488).
- REQUEST/$\overline{\text{WAIT}}$ signal for high-speed data transfer.
- Flexible pattern-recognition logic, programmable as a 16-vector interrupt controller.
- Three independent 16-bit counter/timers with up to four external access lines per counter/timer (count input, output, gate, and trigger), and three output duty cycles (pulsed, one-shot, and square-wave), programmable as retriggerable or nonretriggerable.
- Easy to use since all registers are read/write and directly addressable.

GENERAL DESCRIPTION

The Z8036* CIO Counter/Timers and Parallel I/O elements are general-purpose peripheral circuits, satisfying most counter/timer and parallel I/O needs encountered in system designs. These versatile devices contain three I/O ports and three counter/timers. Many programmable options tailor their configuration to specific applications.

The use of these devices is simplified by making all internal registers (command, status, and data) readable and (except for status bits) writable. Each register is given its own unique address so that it can be accessed directly on the Z8036. The Z8036 is directly Z-Bus compatible.

CIO BLOCK DIAGRAM

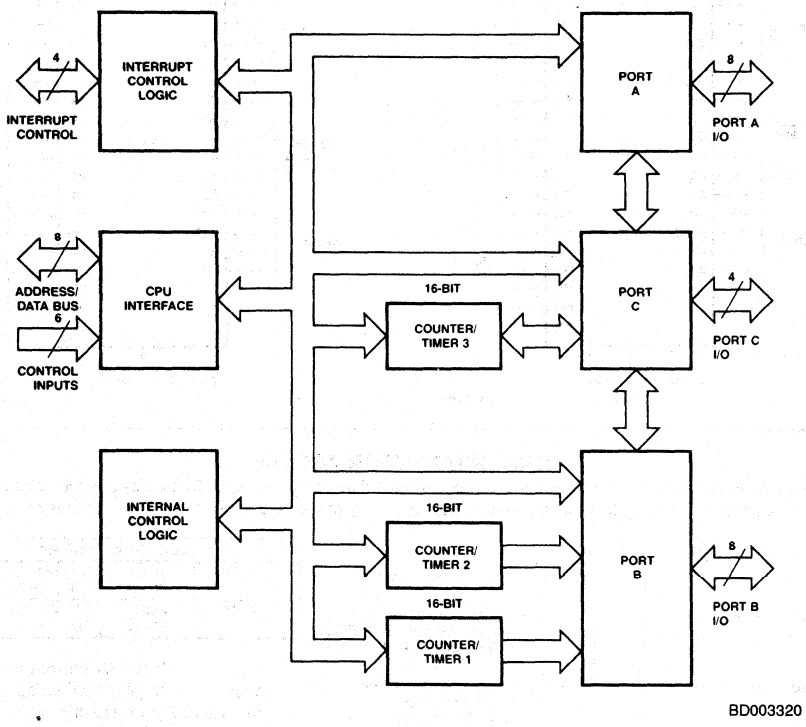


Figure 1.

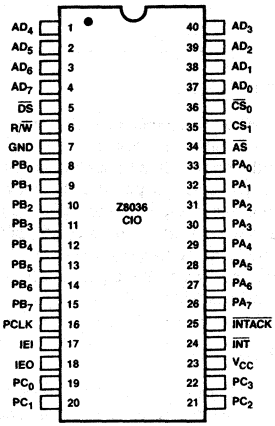
*Z8000 is a trademark of Zilog, Inc.

00958C

Refer to page 7-1 for Essential Information on Military Devices

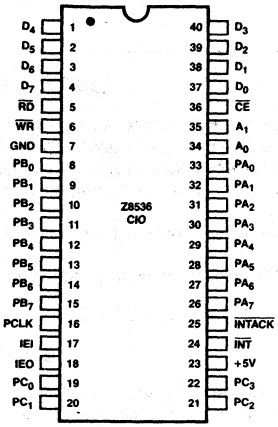
CONNECTION DIAGRAM
Top View

D-40-1, P-40-1



CD005140

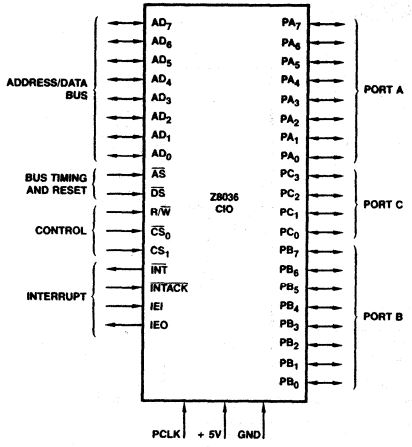
D-40-1, P-40-1



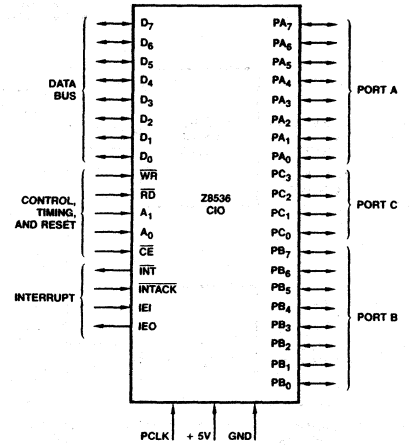
CD005150

Note: Pin 1 is marked for orientation

LOGIC SYMBOL



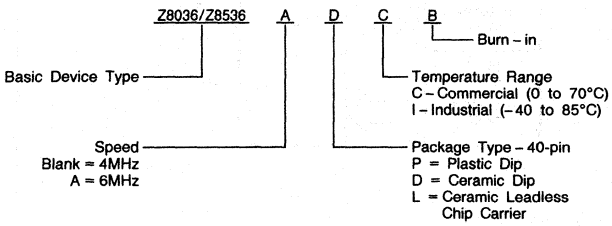
LS001180



LS001190

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations		
Z8036	4MHz 6MHz	DC, PC, DI, ADC, APC
Z8536	4MHz 6MHz	DC, PC, DI, ADC, APC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

00958C

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION

Pin No.	Name	I/O	Description
23	V_{CC}		+5V Power Supply
7	GND		Ground
17	IEI	I	Interrupt Enable In. IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device on the system bus. A HIGH IEI indicates to the CIO that no other higher priority device has an interrupt under service or is requesting an interrupt.
18	IEO	O	Interrupt Enable Out. IEO is normally connected to the next lower priority device's IEI input and inhibits interrupts from lower priority devices. IEO is HIGH only if IEI is HIGH and the CIO has not requested an interrupt.
24	$\overline{INT}$	O	Interrupt Request (Open Drain). This signal is active LOW when the CIO is requesting an interrupt and stays active until the end of the interrupt acknowledge sequence.
25	$\overline{INTACK}$	I	Interrupt Acknowledge. This signal, when active LOW, indicates to the CIO that an interrupt acknowledge cycle is in progress. $\overline{INTACK}$ is sampled while $\overline{AS}$ is LOW.
33-26	PA_0-PA_7	I/O	Port A I/O lines (Bidirectional, 3-state, or Open Drain). These eight I/O lines transfer information between Port A and external devices.
8-15	PB_0-PB_7	I/O	Port B I/O lines (Bidirectional, 3-state, or Open Drain). These eight I/O lines transfer information between Port B and external devices. The lines can also be programmed to provide external access to Counters/Timers 1 and 2.
19-22	PC_0-PC_3	I/O	Port C I/O lines (Bidirectional, 3-state, or Open Drain). These four I/O lines provide Handshake, WAIT and REQUEST lines for Ports A and B, or provide external access to Counter/Timer 3 or access to Port C.
16	PCLK	I	Peripheral Clock. PCLK may be synchronous or asynchronous to the CPU's clock and may be of lower frequency than the CPU's clock. It is used with timers and Request/Wait logic. The input is TTL compatible.
Z8036 Only			
37-40, 1-4	AD_0-AD_7	I/O	Address/Data Bus (Bidirectional, 3-state). Multiplexed address/data lines for transfers between the CPU and CIO.
34	$\overline{AS}$*	I	Address Strobe. Register addresses on AD_0 -AD_7 lines, $\overline{INTACK}$, and $\overline{CS_0}$ are sampled while $\overline{AS}$ is LOW, and latched while $\overline{AS}$ is HIGH.
36, 35	$\overline{CS_0}$ and $\overline{CS_1}$	I	Chip Select 0 and Chip Select 1. Chip Select 0 and Chip Select 1 must be LOW and HIGH, respectively, to select the device. $\overline{CS_0}$ is latched by $\overline{AS}$.
5	$\overline{DS}$*	I	Data Strobe. An active LOW $\overline{DS}$ provides t he timing for transfer of data to or from the CIO. The R/$\overline{W}$ input indicates the direction of data transfer.
6	R/$\overline{W}$	I	Read/Write. R/$\overline{W}$ is active HIGH when the CPU is reading from the CIO and active LOW when the CPU is writing to the CIO.
Z8536 Only			
34, 35	A_0-A_1	I	Address Lines. These two lines are used to select the register involved in the CPU transaction: Port A's Data register, Port B's Data register, Port C's Data register, or a control register.
36	$\overline{CE}$	I	Chip Enable (Active LOW). A LOW level on this input enables the CIO to be read from or written to.
37-40, 1-4	D_0-D_7	I/O	Data Bus (Bidirectional, 3-state). These eight data lines are used for transfers between the CPU and the CIO.
5	$\overline{RD}$**	I	Read (Active LOW). This signal indicates that a CPU is reading from the CIO. During an interrupt acknowledge cycle, this signal gates the interrupt vector onto the data bus if the CIO is the highest priority device requesting an interrupt.
6	$\overline{WR}$**	I	Write (Active LOW). This signal indicates a CPU write to the CIO.

*When $\overline{AS}$ and $\overline{DS}$ are detected LOW at the same time (normally an illegal condition), the CIO is reset.
**When $\overline{RD}$ and $\overline{WR}$ are detected LOW at the same time (normally an illegal condition), the CIO is reset.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... 65 to +150°C
Voltage at any Pin Relative V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.75W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	4MHz	6MHz
Commercial Operating Range T_A = 0 to +70°C V_{CC} = 5V ±5% V_{SS} = 0V	Z8036DC Z8036PC Z8536DC Z8536PC	Z8036ADC Z8036APC Z8536ADC Z8536APC
Industrial Operating Range T_A = −40 to +85°C V_{CC} = 5V ±10% V_{SS} = 0V	Z8036DI Z8536DI	

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input LOW Voltage		−0.5		+0.8	Volts
V_{IH}	Input HIGH Voltage	Standard	2.0		V_{CC}	Volts
		Industrial	2.4			
V_{OL}	Output LOW Voltage	I_{OL} = 2.0mA			0.4	Volts
		I_{OL} = 3.2mA			0.5	
V_{OH}	Output HIGH Voltage	I_{OH} = −250μA	2.4			Volts
I_{OZL}	Output Leakage Current	V_{OUT} = 0.4V			10	μA
I_{OZH}	Output Leakage Current	V_{OUT} = V_{CC}			10	μA
I_I	Input Leakage Current				±10	μA
I_{CC}	Power Supply Current	V_{CC} = MAX | T_A = 0°C			200	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground f = 1MHz			10	pF
C_{OUT}	Output Capacitance				15	pF
$C_{I/O}$	Bidirectional Capacitance				20	pF

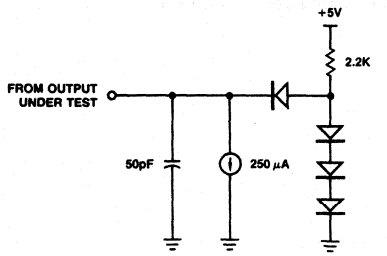
Note: See table for operating range. Typical conditions apply at T_A = 25°C, V_{CC} = 5.0V.

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

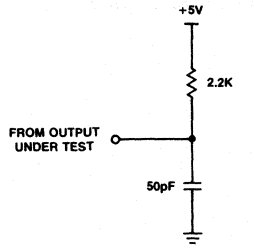
+4.75V ≤ V_{CC} ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ +70°C

Standard Test Load



TC001600

Open-Drain Test Load



TC001610

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z8036/Z8536 HANDSHAKE TIMING (Figures 2, 3, 4)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TsDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ Setup Time	0		0		ns
2	ThDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ Hold Time – Strobed HS	500		500		ns
3	TdACKf(RFD)	$\overline{ACKIN}$ ↓ to RFD ↓ Delay	0		0		ns
4	TwACKl	$\overline{ACKIN}$ Low Width – Strobed HS	250		250		ns
5	TwACKh	$\overline{ACKIN}$ High Width – Strobed HS	250		250		ns
6	TdRFDr(ACK)	RFD ↑ to $\overline{ACKIN}$ ↓ Delay	0		0		ns
7	TsDO(DAV)	Data Out to $\overline{DAV}$ ↓ Setup Time (Note 1)	25		20		ns
8	TdDAVf(ACK)	$\overline{DAV}$ ↓ to $\overline{ACKIN}$ ↓ Delay	0		0		ns
9	ThDO(ACK)	Data Out to $\overline{ACKIN}$ ↓ Hold Time	1		1		$\overline{AS}$ cycle
10	TdACK(DAV)	$\overline{ACKIN}$ ↓ to $\overline{DAV}$ ↑ Delay	1		1		$\overline{AS}$ cycle
11	ThDI(RFD)	Data Input to RFD ↓ Hold Time – Interlocked HS	0		0		ns
12	TdRFDf(ACK)	RFD ↓ to $\overline{ACKIN}$ ↑ Delay – Interlocked HS	0		0		ns
13	TdACKr(RFD)	$\overline{ACKIN}$ ↑ ($\overline{DAV}$ ↑) to RFD ↑ Delay – Interlocked and 3-Wire HS	0		0		ns
14	TdDAVr(ACK)	$\overline{DAV}$ ↓ to $\overline{ACKIN}$ ↑ (RFD ↑) – Interlocked and 3-Wire HS	0		0		ns
15	TdACK(DAV)	$\overline{ACKIN}$ ↑ (RFD ↑) to $\overline{DAV}$ ↓ Delay – Interlocked and 3-Wire HS	0		0		ns
16	TdDAVlf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay – Input 3-Wire HS	0		0		ns
17	ThDI(DAC)	Data Input to DAC ↑ Hold Time – 3-Wire HS	0		0		ns
18	TdDACOr(DAV)	DAC ↑ $\overline{DAV}$ ↑ Delay – Input 3-Wire HS	0		0		ns
19	TdDAVlr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay – Input 3-Wire HS	0		0		ns
20	TdDAVOf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay – Output 3-Wire HS	0		0		ns
21	ThDO(DAC)	Data Output to DAC ↑ Hold Time – 3-Wire HS	1		1		$\overline{AS}$ cycle
22	TdDAClr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay – Output 3-Wire HS	1		1		$\overline{AS}$ cycle
23	TdDAVOr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay – Output 3-Wire HS	0		0		ns

Note: 1. This time can be extended through the use of the Deskew Timers.

*Timings are all preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0."

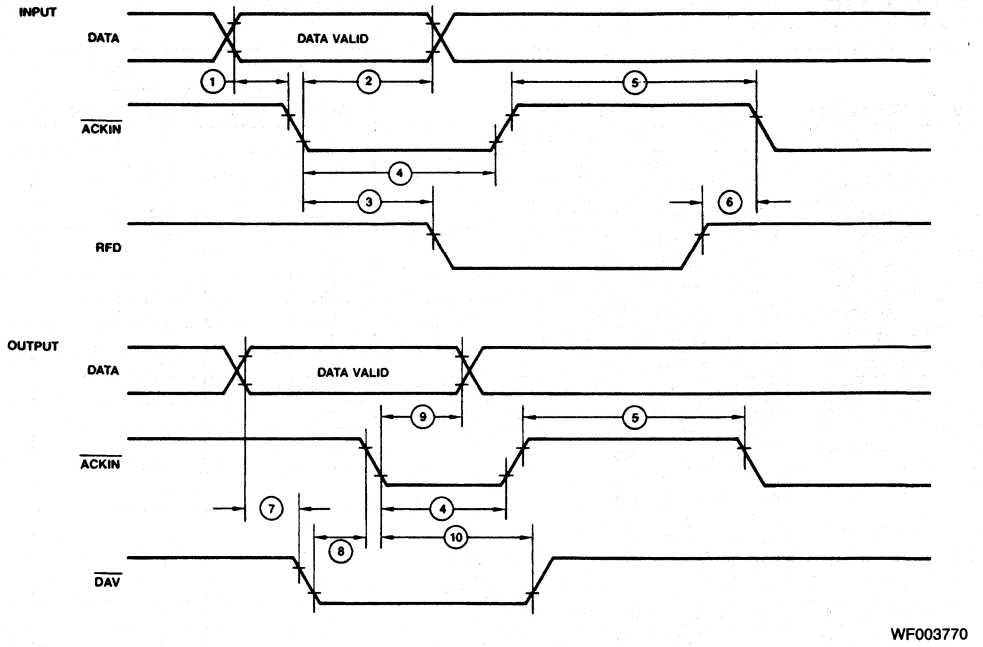


Figure 2. Strobed Handshake

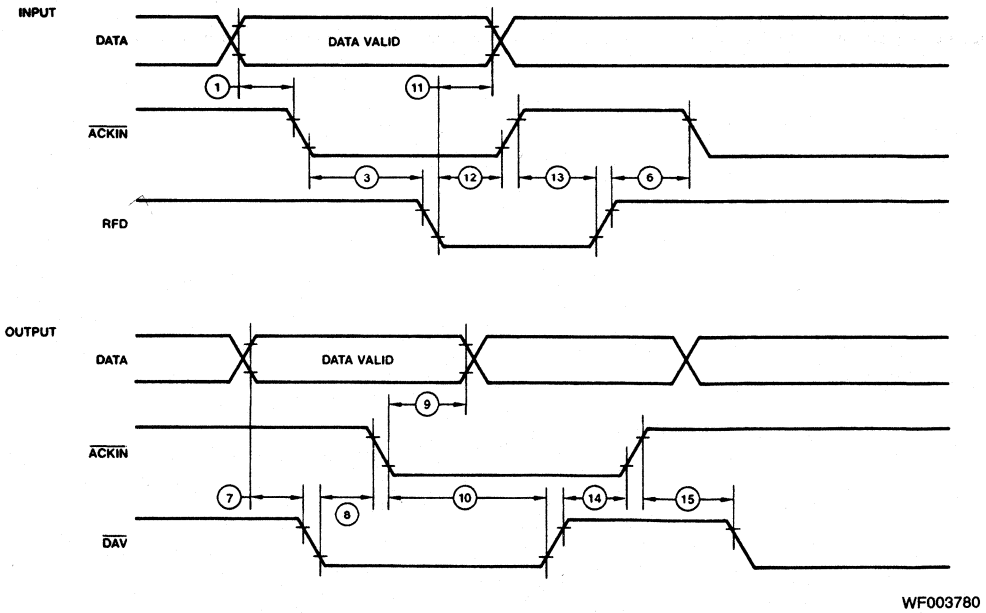


Figure 3. Interlocked Handshake

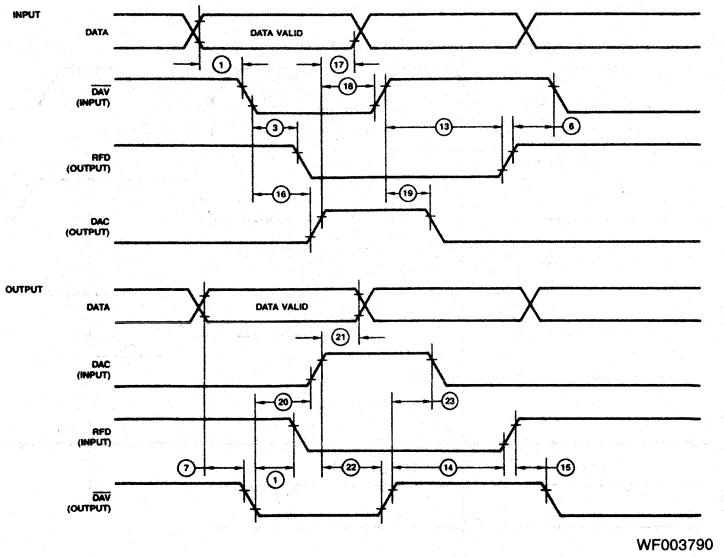


Figure 4. Three-Wire Handshake

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8036 COUNTER/TIMER TIMING (Figure 5)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TcPC	PCLK Cycle Time (Note 1)	250	4000	165	4000	ns
2	TwPCh	PCLK High Width	105	2000	70	2000	ns
3	TwPCl	PCLK Low Width	105	2000	70	2000	ns
4	TfPC	PCLK Fall Time		20		10	ns
5	TrPC	PCLK Rise Time		20		15	ns
6	TcCl	Counter Layout Cycle Time	500		330		ns
7	TCIh	Counter Input High Width	230		150		ns
8	TwCIl	Counter Input Low Width	230		150		ns
9	TfCl	Counter Input Fall Time		20		15	ns
10	TrCl	Counter Input Rise Time		20		15	ns
11	TsTI(PC)	Trigger Input to PCLK Setup Time (Timer Mode) (Note 2)	150		150		ns
12	TsTI(Cl)	Trigger Input to Counter Input Setup Time (Counter Mode) (Note 2)	150		150		ns
13	TwTI	Trigger Input Pulse Width (High or Low)	200		200		ns
14	TsGI(PC)	Gate Input to PCLK ↓ Setup Time (Timer Mode) (Note 2)	100		100		ns
15	TsGI(Cl)	Gate Input to Counter Input ↓ Setup Time (Counter Mode) (Note 2)	100		100		ns
16	ThGI(PC)	Gate Input to PCLK ↓ Hold Time (Timer Mode) (Note 2)	100		100		ns
17	ThGI(Cl)	Gate Input to Counter Input ↓ Hold Time (Counter Mode) (Note 2)	100		100		ns
18	TdPC(CO)	PCLK to Counter Output Delay (Timer Mode)		475		475	ns
19	TdCl(CO)	Counter Input to Counter Output Delay (Counter Mode)		475		475	ns

Notes: 1. PCLK is only used with the counter/timers (in Timer mode), the deskew timers, and the REQUEST/$\overline{\text{WAIT}}$ logic. If these functions are not used, the PCLK input can be held low.
2. These parameters must be met to guarantee that either the trigger or gate is valid for the next counter/timer cycle.

*Timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0."

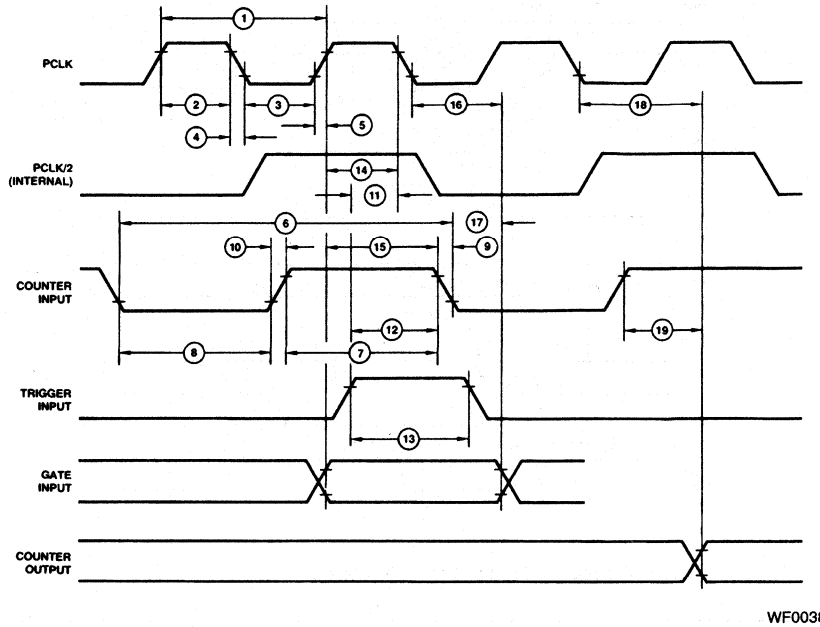


Figure 5. Counter/Timer Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8036 RESET TIMING (Figure 6)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TdDSQ(AS)	Delay from $\overline{DS}$ ↑ to $\overline{AS}$ ↓ for No Reset	40		15		ns
2	TdASQ(DS)	Delay from $\overline{AS}$ ↑ to $\overline{DS}$ ↓ for No Reset	50		30		ns
3	TwRES	Minimum Width of $\overline{AS}$ and $\overline{DS}$ both Low for Reset (Note 1)	250		170		ns

Note: 1. Internal circuitry allows for the reset provided by the Z8 ($\overline{DS}$ held Low while $\overline{AS}$ pulses) to be sufficient.

*Timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0".

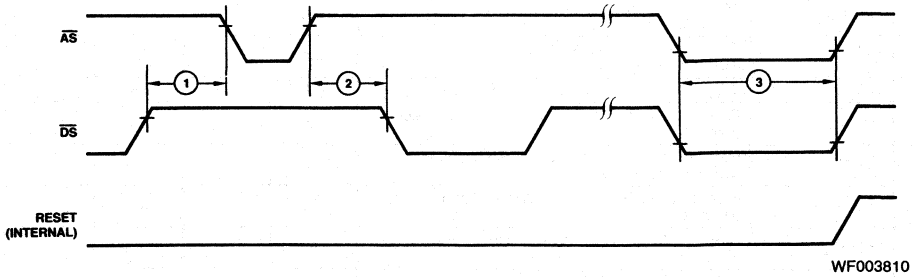


Figure 6. Reset Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8036/Z8536 MISCELLANEOUS PORT TIMING (Figure 7)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	Trl	Any Input Rise Rate Time		100		100	ns
2	Tfl	Any Input Fall Time		100		100	ns
3	Tw1's	1's Catcher High Width (Note 1)	250		170		ns
4	TwPM	Pattern Match Input Valid (Bit Port)	750		500		ns
5	TsPMD	Data Latched on Pattern Match Setup Time (Bit Port)	0		0		ns
6	ThPMD	Data Latched on Pattern Match Hold Time (Bit Port)	1000		650		ns

Note: 1. If the input is programmed inverting, a Low-going pulse of the same width will be detected.
*Timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0."

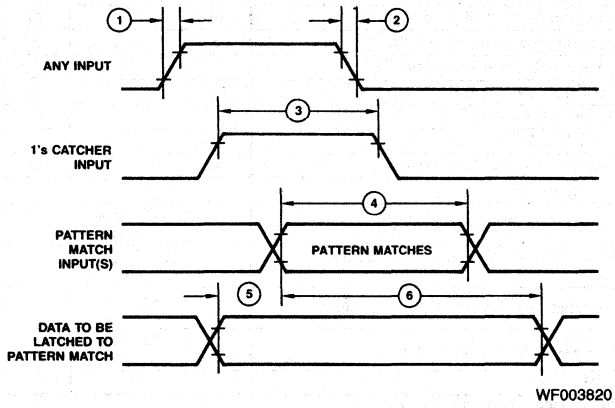


Figure 7. Miscellaneous Port Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z8036 CPU INTERFACE TIMING (Figure 8)

Number	Parameters	Description	4MHz Min	4MHz Max	6MHz Min	6MHz Max	Units
1	TwAs	$\overline{AS}$ Low Width	70	2000	50	2000	ns
2	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time (Note 1)	30		10		ns
3	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time (Note 1)	50		30		ns
4	TsA(DS)	Address to $\overline{DS}$ ↓ Setup Time (Note 1)	130		100		ns
5	TsCSO(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Setup Time (Note 1)	0		0		ns
6	ThCSO(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Hold Time (Note 1)	60		40		ns
7	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay (Note 1)	60		40		ns
8	TsCSI(DS)	CS_1 to $\overline{DS}$ ↓ Setup Time	100		80		ns
9	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Setup Time	100		80		ns
10	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Setup Time	0		0		ns
11	TwDS	$\overline{DS}$ Low Width	390		250		ns
12	TsDW(DSf)	Write Data to $\overline{DS}$ ↓ Setup Time	30		20		ns
13	TdDS(DRV)	$\overline{DS}$ (Read) ↓ to Address Data Bus Driven	0		0		ns
14	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	ns
15	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		20		ns
16	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		ns
17	TdDS(DRz)	$\overline{DS}$ ↑ Read Data Float Delay (Note 2)		70		45	ns
18	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		ns
19	ThCSI(DS)	CS_1 to $\overline{DS}$ ↑ Hold Time	55		40		ns
20	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		ns
21	Trc	Valid Access Recover Time (Note 3)	1000		650		ns
Z8036 INTERRUPT TIMING (Figure 9)							
22	TdPM(INT)	Pattern Match to INT Delay (Bit Port)		$\overline{AS}$ cycle + 800ns		1	AS cycle + ns
23	TdACK(INT)	$\overline{ACKIN}$ to $\overline{INT}$ Delay (Port with Handshake) (Note 4)		4 $\overline{AS}$ cycles + 600ns		4	AS cycle + ns
24	TdCI(INT)	Counter Input to $\overline{INT}$ Delay (Counter Mode)		$\overline{AS}$ cycle + 700ns		1	AS cycle + ns
25	TdPC(INT)	PCLK to $\overline{INT}$ Delay (Timer Mode)		$\overline{AS}$ cycle + 700ns		1	AS cycle + ns
26	TdAS(INT)	$\overline{AS}$ to $\overline{INT}$ Delay		300			ns
Z8036 INTERRUPT ACKNOWLEDGE TIMING (Figure 10)							
27	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	0		0		ns
28	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		250		ns
29	TsAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Setup Time (Note 5)	350		250		ns
30	TdDSA(DR)	$\overline{DS}$ (Acknowledge) ↓ to Read Data Valid Delay		250		180	ns
31	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		250		ns
32	TdAS(IEO)	$\overline{AS}$ ↓ to IEO ↓ Delay ($\overline{INTACK}$ Cycle) (Note 5)		350		250	ns
33	TdIEI(IEO)	IEI to IEO Delay (Note 5)		150		100	ns
34	TsIEI(DSA)	IEI to $\overline{DS}$ (Acknowledge) ↓ Setup Time (Note 5)	100		70		ns
35	ThIEI(DSA)	IEI to $\overline{DS}$ (Acknowledge) ↑ Hold Time	100		70		ns
36	TdDSA(INT)	$\overline{DS}$ (Acknowledge) ↓ to $\overline{INT}$ ↑ Delay		600		600	ns

Notes:
1. Parameters do not apply to Interrupt Acknowledge transactions.
2. Float Delay is measured to the time when the output has changed 0.5V from steady state with minimum AC load and maximum DC load.
3. This is the delay from $\overline{DS}$↑ of one CIO access to $\overline{DS}$↓ of another CIO access.
4. The delay is from $\overline{DAV}$↓ for 3-Wire Input Handshake. The delay is from DAC↑ for 3-Wire Output Handshake. One additional $\overline{AS}$ cycle is required for ports in the Single Buffered mode.
5. The parameters for the devices in any particular daisy chain must meet the following constraints: the delay from $\overline{AS}$↑ to $\overline{DS}$↓ must be greater than the sum of TdAS(IEO) for the highest priority peripheral, TsIEI(DSA) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.

*Timings are all preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0".

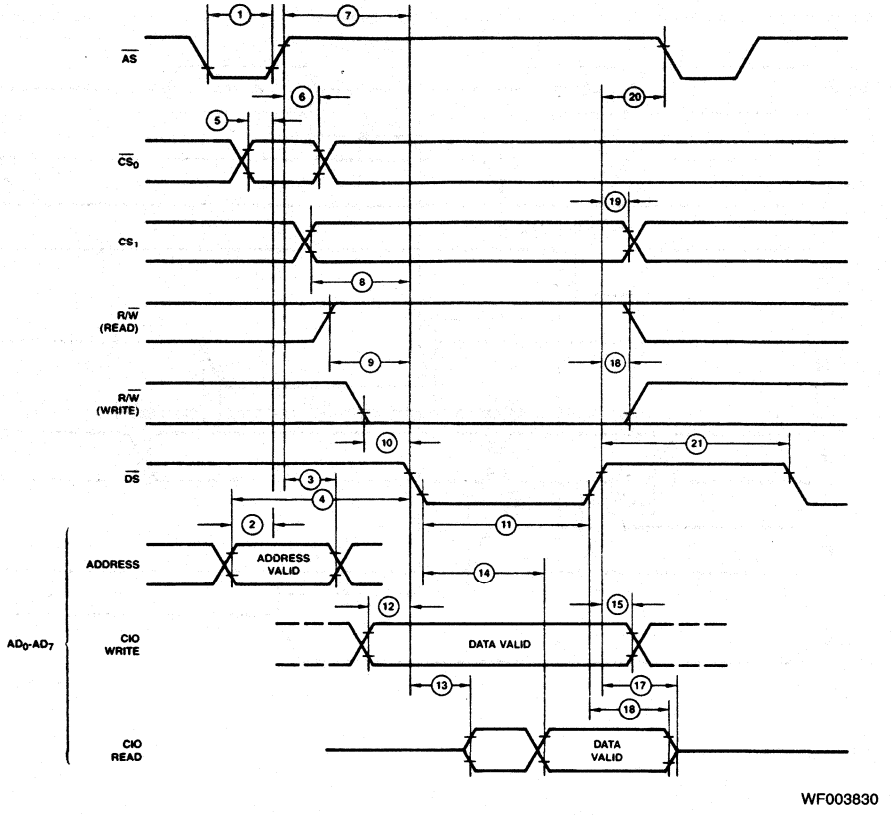


Figure 8. Z8036 CPU Interface Timing

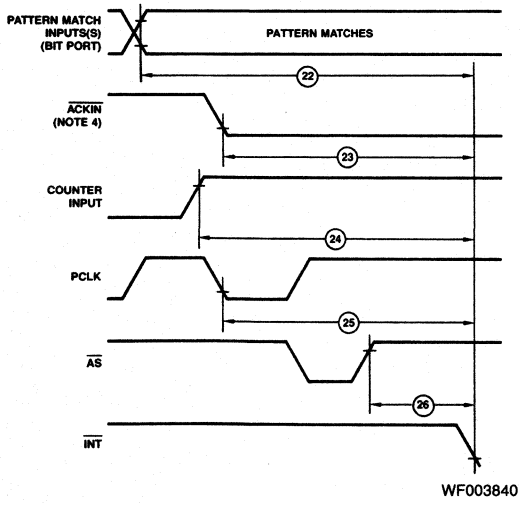


Figure 9. Z8036 Interrupt Timing

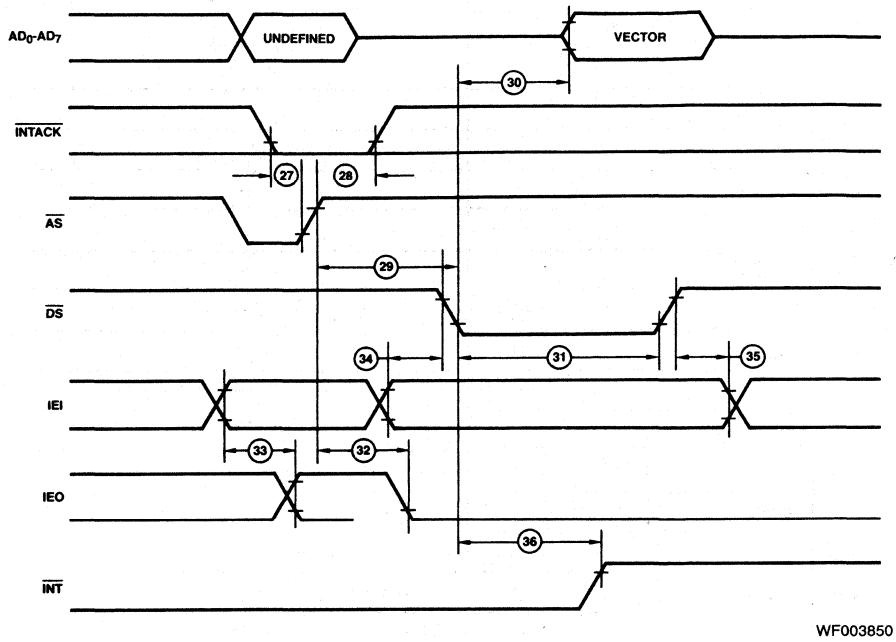


Figure 10. Z8036 Interrupt Acknowledge Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8536 CPU INTERFACE TIMING (Figure 11)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TcPC	PCLK Cycle Time	250	4000	165	4000	ns
2	TwPCh	PCLK Width (High)	105	2000	70	2000	ns
3	TwPCl	PCLK Width (Low)	105	2000	70	2000	ns
4	TrPC	PCLK Rise Time		20		10	ns
5	TfPC	PCLK Fall Time		20		15	ns
6	TslA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	100		100		ns
7	ThlA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	0		0		ns
8	TslA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1)	200		200		ns
9	ThlA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		ns
10	TslA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time	200		200		ns
11	ThlA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
12	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		80		ns
13	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		ns
14	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		80		ns
15	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		ns
16	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time (Note 1)	0		0		ns
17	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time (Note 1)	100		70		ns
18	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note 1)	0		0		ns
19	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		ns
20	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		70		ns
21	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		ns
22	TwRDl	$\overline{RD}$ Low Width (Note 1)	390		250		ns
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns
24	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180	ns
25	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay (Note 2)		70		45	ns
27	TwWRl	$\overline{WR}$ Low Width	390		250		ns
28	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	0		0		ns
29	ThDW(WR)	Write Data To $\overline{WR}$ ↑ Hold Time	0		0		ns
30	Trc	Valid Access Recovery Time (Note 3)	1000*		650*		ns
Z8536 INTERRUPT TIMING (Figure 12)							
31	TdPM(INT)	Pattern Match to $\overline{INT}$ Delay (Bit Port)		2		2	TcPC + ns
32	TdACK(INT)	$\overline{ACKIN}$ to $\overline{INT}$ Delay (Port with Handshake) (Note 4)		10		10	TcPC + ns
33	TdCl(INT)	Counter Input to $\overline{INT}$ Delay (Counter Mode)		2		2	TcPC + ns
34	TdPC(INT)	PCLK to $\overline{INT}$ Delay (Timer Mode)		3		3	TcPC + ns

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8536 INTERRUPT ACKNOWLEDGE TIMING (Figure 13)

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
35	TsIA(RDA)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Setup Time (Note 5)	350		250		ns
36	TwRDA	$\overline{RD}$ (Acknowledge) Width	350		250		ns
37	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		255		180	ns
38	TdIA(IEO)	$\overline{INTACK}$ ↓ to IEO↓ Delay (Note 5)		350		250	ns
39	TdIEI(IEO)	IEI to IEO Delay (Note 5)		150		100	ns
40	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time (Note 5)	100		70		ns
41	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	100		70		ns
42	TdRDA(INT)	$\overline{RD}$ ↓ (Acknowledge) to $\overline{INT}$ ↑ Delay		600		600	ns

Notes: 1. Parameters do not apply to Interrupt Acknowledge transactions.
2. Float Delay is measured to the time when the output has changed 0.5V with minimum AC load and maximum DC load.
3. Trc is 1μs or 3 TcPC, whichever is longer.
4. The delay is from $\overline{DAV}$↓ for 3-Wire Input Handshake. The delay is from DAC↑ for 3-Wire Output Handshake.
5. The parameters for the devices in any paritucular daisy chain must meet the following constraints: the delay from $\overline{INTACK}$↓ to $\overline{RD}$↓ must be greater than the sum of TdIA(IEO) for the highest priority peripheral, TsIEI(RDA) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.

*Timings are preliminary and subject to change.

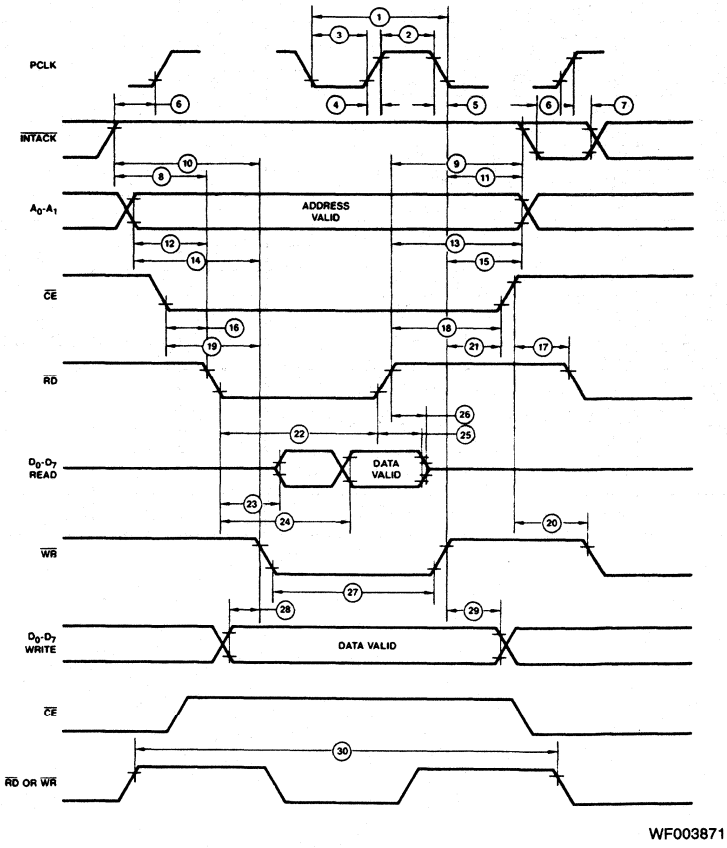


Figure 11. Z8536 CPU Interface Timing

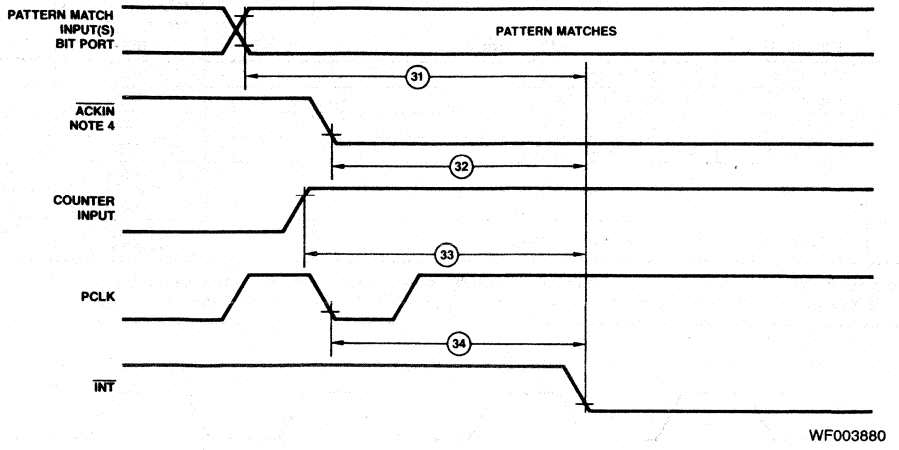


Figure 12. Interrupt Timing

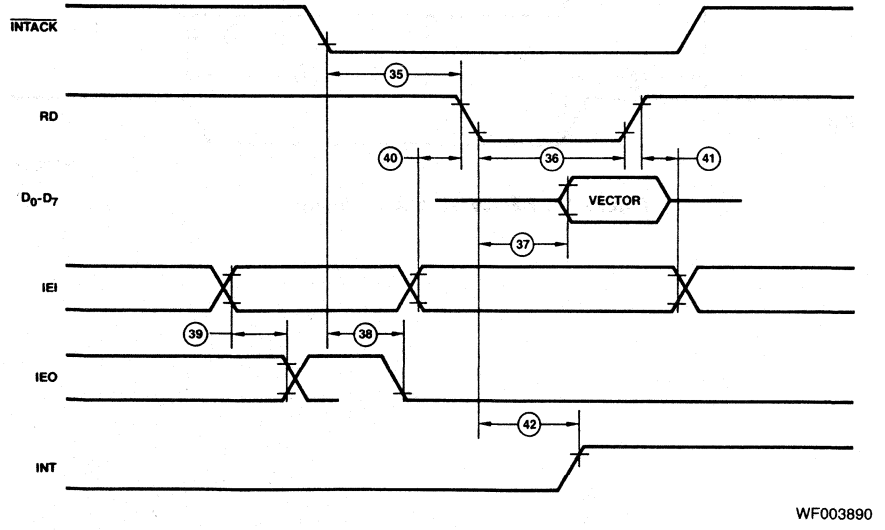


Figure 13. Interrupt Acknowledge Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z8036 REQUEST/$\overline{\text{WAIT}}$ TIMING (Figure 14)

Number	Parameters	Description	4MHz Min	4MHz Max	Units
1	TdDS(REQ)	$\overline{DS}$ ↓ to REQ ↓ Delay		500	ns
2	TdDS(WAIT)	$\overline{DS}$ ↓ to $\overline{WAIT}$ ↓ Delay		500	ns
3	TdPC(REQ)	PCLK ↓ to REQ ↑ Delay		300	ns
4	TdPC(WAIT)	PCLK ↓ to $\overline{WAIT}$ ↑ Delay		300	ns
5	TdACK(REQ)	$\overline{ACKIN}$ ↓ to REQ ↑ Delay (Note 1)		3 $\overline{AS}$ + 2PCLK + 1000ns	$\overline{AS}$ cycles + PCLK cycles + ns
6	TdACK(WAIT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ ↑ Delay		10 PCLK + 600ns	PCLK cycles + ns

Note: 1. The delay is from $\overline{DAV}$↓ for 3-Wire Input Handshake. The delay is from DAC↑ for 3-Wire Output Handshake.
*Timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0".

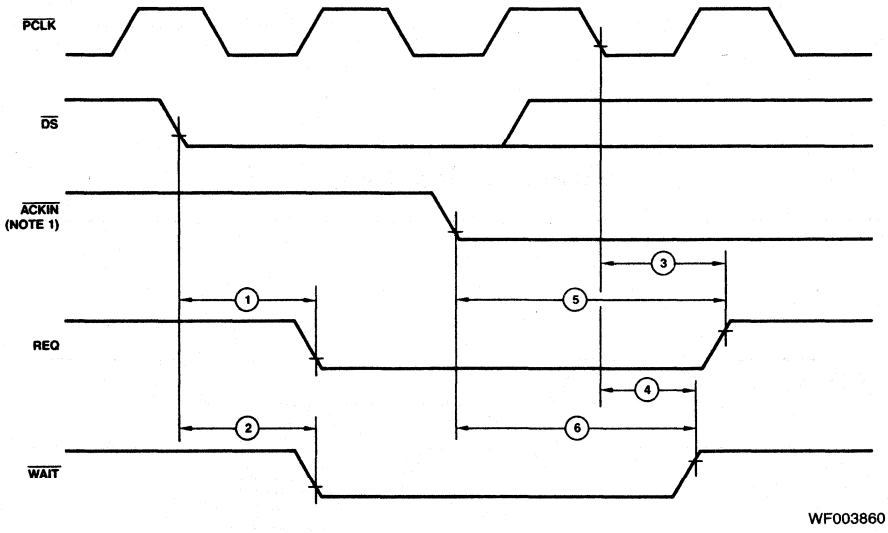


Figure 14. REQUEST/WAIT Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Z8036 REQUEST/$\overline{\text{WAIT}}$ TIMING (Figure 15)

Number	Parameters	Description	4MHz		Units
			Min	Max	
1	TdRD(REQ)	$\overline{RD}$ ↓ to REQ ↓ Delay		500	ns
2	TdRD(WAIT)	$\overline{RD}$ ↓ to $\overline{WAIT}$ ↓ Delay		500	ns
3	TdWR(REQ)	$\overline{WR}$ ↓ to REQ ↓ Delay		500	ns
4	TdWR(WAIT)	$\overline{WR}$ ↓ to $\overline{WAIT}$ ↓ Delay		500	ns
5	TdPC(REQ)	PCLK ↓ to REQ ↑ Delay		300	ns
6	TdPC(WAIT)	PCLK ↓ to $\overline{WAIT}$ ↑ Delay		300	ns
7	TdACK(REQ)	$\overline{ACKIN}$ ↓ to REQ ↑ Delay (Note 1)		8 + 100	TcPC + ns
8	TdACK(WAIT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ ↑ Delay (Note 1)		10 + 600	TcPC + ns

Note: 1. The delay is from $\overline{DAV}$↓ for 3-Wire Input Handshake. The delay is from DAC↑ for 3-Wire Output Handshake.
*Timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0".

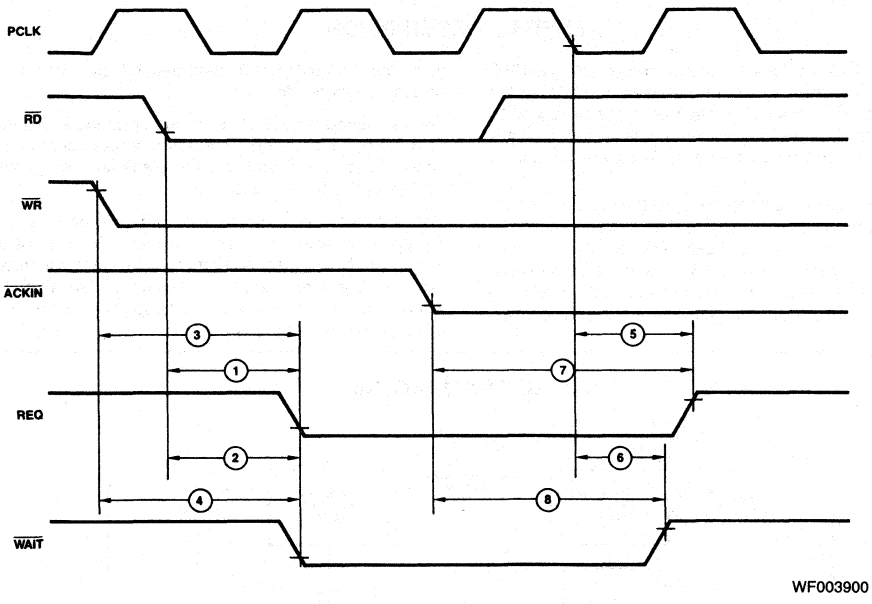


Figure 15. Request/$\overline{\text{WAIT}}$ Timing

Z8038(FIO)

128 Byte FIFO I/O Port

DISTINCTIVE CHARACTERISTICS

- **Asynchronous FIFO Interface** —
  128-byte FIFO provides bidirectional CPU to CPU or peripheral interface.
- **Expandable in Length and Width** —
  FIOs can be connected in parallel for wider words, can be cascaded for deeper stacks.
- **2-Wire and 3-Wire handshake logic** —
  Control logic on chip for interlocked two-wire handshake as well as three-wire scheme used in IEEE-488.
- **Pattern matching logic on chip** —
  FIO can detect a data pattern and interrupt CPU.
- **Byte count available to software** —
  An on-chip register which contains the actual number of bytes in the FIFO can be read by the software to determine stack status.

GENERAL DESCRIPTION

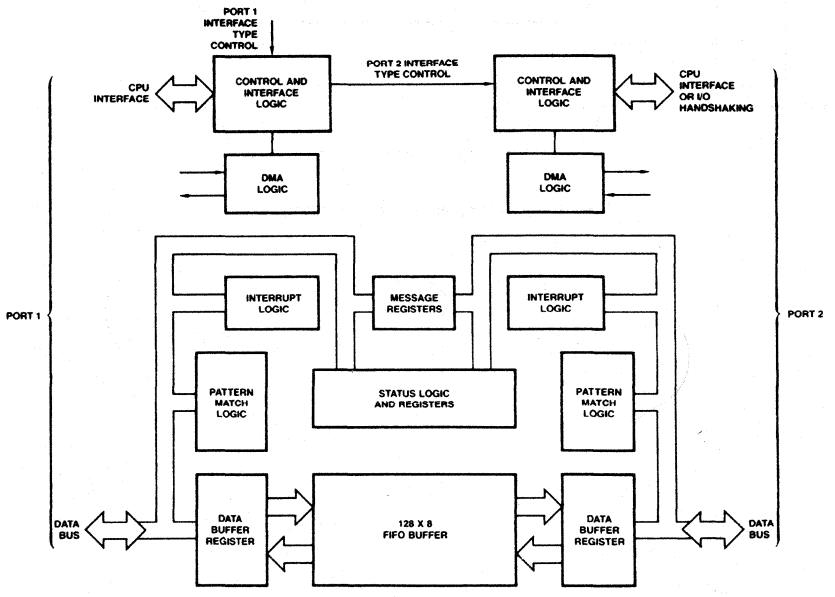
The Z8038* FIO provides an asynchronous 128-byte FIFO buffer between two CPUs or between a CPU and a peripheral device. This buffer interface expands to a 16-bit or wider data path and expands in depth to add as many Z8060 FIFOs (and an additional FIO) as are needed.

The FIO manages data transfers by assuming Z-BUS, non-Z-BUS microprocessor (a generalized microprocessor interface), Interlocked 2-Wire Handshake, and 3-Wire Handshake operating modes. These modes interface dissimilar CPUs or CPUs and peripherals running under differing speeds or protocols, allowing asynchronous data transactions and improving I/O overhead by as much as two orders of magnitude.

The FIO supports the Z-BUS interrupt protocols, generating seven sources of interrupts. Each interrupt source can be enabled or disabled, and can also place an interrupt vector on the port address/data lines.

The data transfer logic of the FIO has been specially designed to work with DMA (Direct Memory Access) devices for high-speed transfers. The FIO also supports the variably sized block length, improving system throughput when multiple variable length messages are transferred amongst several sources.

FIO BLOCK DIAGRAM



BD003350

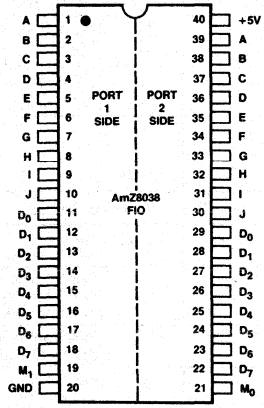
*Z8000 is a trademark of Zilog, Inc.

00867B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40, P-40



CD005180

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

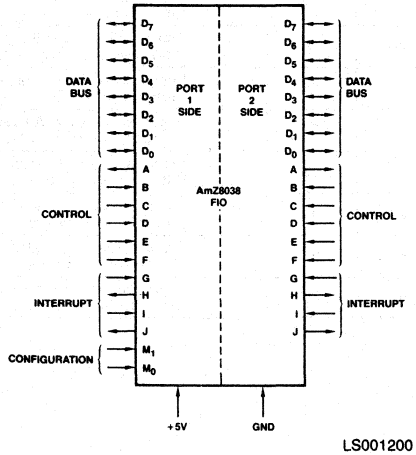
Z8038 A D C

- Z8038 — Basic Device Type
- A — Speed: Blank = 4MHz, A = 6MHz
- D — Package Type – 40-pin: P – Plastic Dip; D – Ceramic Dip; L – Ceramic Leadless Chip Carrier
- C — Temperature Range: C – Commercial (0°C to 70°C); I – Industrial (–40°C to 85°C)

Valid Combinations	
Z8038	DC, PC, DI, ADC, APC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.



LS001200

Pin Functions

PIN DESCRIPTION

Pin Signals	Pin No.		Name	Description
PINS COMMON TO BOTH SIDES				
M_0	21		M_0	M_1 and M_0 program Port 1 side CPU interface
M_1	19		M_1	
+5 Vdc	40		+5 Vdc	DC power source
GND	20		GND	DC power ground
	Pin No. Port			
Pin Signals	**1**	**2**	**Name**	**Description**
Z-BUS LOW BYTE MODE				
AD_0-AD_7 (Address/Data)	11-18	29-22	D_0-D_7	Multiplexed bidirectional address/data lines, Z-BUS compatible.
$\overline{REQ}/\overline{WAIT}$ (Request/Wait)	1	39	A	Output, active Low, REQUEST (ready) line for DMA transfer, WAIT line (open-drain) output for synchronized CPU and FIO data transfers.
$\overline{DMASTB}$ (Direct Memory Access Strobe)	2	38	B	Input, active Low. Strobes DMA data to and from the FIFO buffer.
$\overline{DS}$ (Data Strobe)	3	37	C	Input, active Low. Provides timing for data transfer to or from FIO.
$R/\overline{W}$ (Read/Write)	4	36	D	Input, active High signals CPU read from FIO; active Low signals CPU write to FIO.
$\overline{CS}$ (Chip Select)	5	35	E	Input, active Low. Enables FIO. Latched on the rising edge of $\overline{AS}$.
$\overline{AS}$ (Address Strobe)	6	34	F	Input, active Low. Addresses, $\overline{CS}$ and $\overline{INTACK}$, are sampled while $\overline{AS}$ is Low.
$\overline{INTACK}$ (Interrupt Acknowledge)	7	33	G	Input, active Low. Acknowledges an interrupt. Latched on the rising edge of $\overline{AS}$.
IEO (Interrupt Enable Out)	8	32	H	Output, active High. Sends interrupt enable to lower priority device IEI pin.
IEI (Interrupt Enable In)	9	31	I	Input, active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (Interrupt)	10	30	J	Output, open drain, active Low. Signals FIO interrupt request to CPU.
Z-BUS HIGH BYTE MODE				
AD_0-AD_7 (Address/Data)	11-18	29-22	D_0-D_7	Multiplexed bidirectional address/data lines, Z-BUS compatible.
$\overline{REQ}/\overline{WAIT}$ (Request/Wait)	1	39	A	Output, active Low, REQUEST (ready) line for DMA transfer, $\overline{WAIT}$ line (open-drain) output for synchronized CPU and FIO data transfers.
$\overline{DMASTB}$ (Direct Memory Access Strobe)	2	38	B	Input, active Low. Strobes DMA data to and from the FIFO buffer.
$\overline{DS}$ (Data Strobe)	3	37	C	Input, active Low. Provides timing for transfer of data to or from FIO.
$R/\overline{W}$ (Read/Write)	4	36	D	Input, active High. Signals CPU read from FIO; active Low signals CPU write to FIO.
$\overline{CS}$ (Chip Select)	5	35	E	Input, active Low. Enables FIO. Latched on the rising edge of $\overline{AS}$.
$\overline{AS}$ (Address Strobe)	6	34	F	Input, active Low. Addresses, $\overline{CS}$ and $\overline{INTACK}$, are sampled while $\overline{AS}$ is Low.
A_0 (Address Bit 0)	7	33	G	Input, active High. With A_1, A_2, and A_3, addresses FIO internal registers.
A_1 (Address Bit 1)	8	32	H	Input, active High. With A_0, A_2, and A_3, addresses FIO internal registers.
A_2 (Address Bit 2)	9	31	I	Input, active High. With A_0, A_1, and A_3, addresses FIO internal registers.
A_3 (Address Bit 3)	10	30	J	Input, active High. With A_0, A_1, and A_2, addresses FIO internal registers.

00867B

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin Signals	Pin No. Port 1	Pin No. Port 2	Name	Description
NON-Z-BUS MODE				
D_0-D_7 (Data)	11-18	29-22	D_0-D_7	Bidirectional data bus.
$\overline{REQ}/\overline{WT}$ (Request/Wait)	1	39	A	Output, active Low. REQUEST (ready) line for DMA transfer, $\overline{WAIT}$ line (open-drain) output for synchronized CPU and FIO data transfer.
$\overline{DACK}$ (DMA Acknowledge)	2	38	B	Input, active Low. DMA acknowledge.
$\overline{RD}$ (Read)	3	37	C	Input, active Low. Signals CPU read from FIO.
$\overline{WR}$ (Write)	4	36	D	Input, active Low. Signals CPU write to FIO.
$\overline{CE}$ (Chip Select)	5	35	E	Input, active Low. Used to select FIO.
C/$\overline{D}$ (Control/Data)	6	34	F	Input, active High. Identifies control byte on D_0-D_7. Active Low identifies data byte on D_0-D_7.
$\overline{INTACK}$ (Interrupt Acknowledge)	7	33	G	Input, active Low. Acknowledges an interrupt.
IEO (Interrupt Enable Out)	8	32	H	Output, active High. Sends interrupt enable to lower priority device IEI pin.
IEI (Interrupt Enable In)	9	31	I	Input, active High, Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (Interrupt)	10	30	J	Output, open drain, active Low. Signals FIO interrupt to CPU.
Pin Signals	**Pin No.**	**Mode**	**Name**	**Description**
PORT 2 - I/O PORT MODE				
D_0-D_7 (Data)	29-22	2-Wire HS* 3-Wire HS	D_0-D_7	Bidirectional data bus.
RFD/$\overline{DAV}$ (Ready for Data/Data Available)	39	2-Wire HS 3-Wire HS	A	Output, RFD active High. Signals peripherals that F IO is ready to receive data. $\overline{DAV}$ active Low signals that FIO is ready to send data to peripherals.
$\overline{ACKIN}$ (Acknowledge Input)	38	2-Wire HS	B	Input, active Low. Signals FIO that output data is received by peripherals or that input data is valid.
$\overline{DAV}$/DAC (Data Available/Data Accepted)	38	3-Wire HS	B	Input, $\overline{DAV}$ (active Low) signals that data is valid on bus. DAC (active High) signals that output data is accepted by peripherals.
FULL	37	2-Wire HS	C	Output, open drain, active High. Signals that FIO buffer is full.
DAC/RFD (Data Accepted/Ready for Data)	37	3-Wire HS	C	Direction controlled by internal programming. Both active High. DAC (an output) signals that FIO has received data from peripheral; RFD (an input) signals that the listeners are ready for data.
EMPTY	36	2-Wire HS 3-Wire HS	D	Output, open drain, active High. Signls that FIFO buffer is empty.
$\overline{CLEAR}$	35	2-Wire HS 3-Wire HS	E	Programmable input or output, active Low. Clears all data from FIFO buffer.
DATA DIR (Data Direction)	34	2-Wire HS 3-Wire HS	F	Programmable input or output. Active High signals data input to Port 2; Low signals data output from Port 2.
IN_0	33	2-Wire HS 3-Wire HS	G	Input line to D_0 of Control Register 3.
OUT_1	32	2-Wire HS 3-Wire HS	H	Output line from D_1 of Control Register 3.
$\overline{OE}$ (Output Enable)	31	2-Wire HS 3-Wire HS	I	Input, active Low. When Low, enables bus drivers. When High, floats bus drivers at high impedance.
OUT_3	30	2-Wire HS 3-Wire HS	J	Output line from D_3 of Control register 3.

*Handshake

ARCHITECTURE

The FIO is a universal interface between two independent systems operating asynchronously. Conceptually it consists of two programmable interfaces connected by a 128-byte FIFO buffer and a pair of message registers, as shown in Figure 1.

Each port contains 8 lines used for data and for programming the FIO, and 10 lines used for various control functions. The function and timing relationships of the 10 control lines change completely according to the type of interface the port is programmed to be. Either port can be programmed to interface directly to a Z8000 CPU Bus (Z-bus) or to an 8080 type bus (8080, 8085, Z80). The 10 control lines perform the functions needed for read and write strobes, DMA control, and interrupt handling. The port 2 side can also be programmed for interface to peripheral; in this case, the 10 lines are used for handshaking, direction control, and cascading.

The two sides are connected by the FIFO buffer and the message registers. The buffer is used to move data between the two ports. Since it is a FIFO, reads and writes can occur simultaneously and independently. Each port is tied to its own system interface, with the FIFO providing reliable data transfer between them. The message registers are used to transmit information between the two ports without going through the FIFO. They (one going each direction) are intended for control words being sent from one CPU to another.

The CPU interface includes a vectored interrupt capability which covers all the ways in which a transfer might be terminated. The CPU, therefore, can set up the FIO to communicate with a peripheral or another CPU, and then not deal with the FIO until an interrupt occurs. Interrupts can be programmed to occur if any of these conditions takes place:

Buffer Empty
Buffer Full
Overflow/Underflow
Byte Count Match
Data Pattern Match
Data Direction Change
Message Present

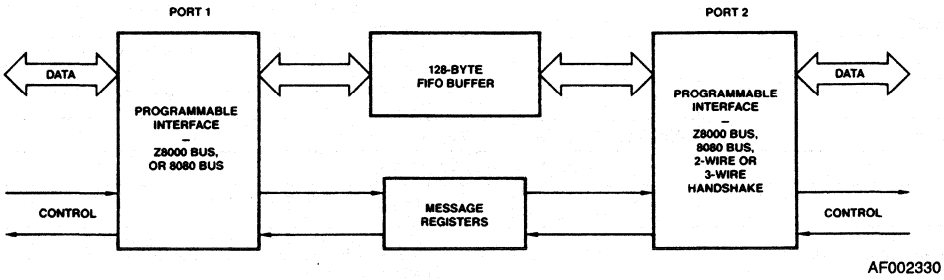


Figure 1. FIO Concept

DETAILED DESCRIPTION

OPERATING MODES

Ports 1 and 2 operate in any of twelve combinations of operating modes, listed in Table 2. Port 1 functions in either the Z-BUS or non-Z-BUS microprocessor modes, while Port 2 functions in Z-BUS, non-Z-BUS, Interlocked 2-Wire Handshake, and 3-Wire Handshake modes. Table 1 describes the signals and their corresponding pins in each of these modes.

The pin diagrams of the FIO are identical, except for two pins on the Port 1 side, which select that port's operating mode. Port 2's operating mode is programmed by two bits in Port 1's Control register 0. Table 2 describes the combinations of operating modes; PIN DESCRIPTION describes the control signals mapped to pins A–J in the five possible operating modes.

RESET

The FIO can be reset under either hardware or software control by one of the following methods:

- By forcing both $\overline{AS}$ and $\overline{DS}$ LOW simultaneously in Z-BUS mode (normally illegal).
- By forcing $\overline{RD}$ and $\overline{WR}$ LOW simultaneously in non-Z-BUS mode.
- By writing a 1 to the Reset bit in Control register 0 for software reset.

In the Reset state, all control bits are cleared to 0. Only after clearing the Reset bit (by writing a 0 to it) can the other command bits be programmed. This action is true for both sides of the FIO when programmed as a CPU interface.

TABLE 1. PIN ASSIGNMENTS

Control Signal Pins	Z-BUS Low Byte	Z-BUS High Byte	Non-Z-BUS	Port 2 Only	
				Interlocked HS Port	3-Wire HS Port
A	$\overline{REQ}/\overline{WT}$	$\overline{REQ}/\overline{WT}$	$\overline{REQ}/\overline{WT}$	RFD/$\overline{DAV}$	RFD/$\overline{DAV}$
B	$\overline{DMASTB}$	$\overline{DMASTB}$	$\overline{DACK}$	$\overline{ACKIN}$	$\overline{DAV}$/DAC
C	$\overline{DS}$	$\overline{DS}$	$\overline{RD}$	FULL	DAC/RFD
D	R/$\overline{W}$	R/$\overline{W}$	$\overline{WR}$	EMPTY	EMPTY
E	$\overline{CS}$	$\overline{CS}$	$\overline{CE}$	$\overline{CLEAR}$	$\overline{CLEAR}$
F	$\overline{AS}$	$\overline{AS}$	C/$\overline{D}$	DATA DIR	DATA DIR
G	$\overline{INTACK}$	A_0	$\overline{INTACK}$	IN_0	IN_0
H	IEO	A_1	IEO	OUT_1	OUT_1
I	IEI	A_2	IEI	OE	OE
J	$\overline{INT}$	A_3	$\overline{INT}$	OUT_3	OUT_3

For proper system control, when Port 1 is reset, Port 2 is also reset. In addition, all Port 2's outputs are floating and all inputs are ignored. To initiate the data transfer, Port 2 must be enabled by Port 1. The Port 2 CPU can determine when it is enabled by reading Control register 0, which reads "floating" data bus if not enabled and "01_H" if enabled.

TABLE 2. OPERATING MODES

MODE	M_1	M_0	B_1*	B_0*	Port 1	Port 2
0	0	0	0	0	Z-BUS Low Byte	Z-BUS Low Byte
1	0	0	0	1	Z-BUS Low Byte	Non-Z-BUS
2	0	0	1	0	Z-BUS Low Byte	3-Wire Handshake
3	0	0	1	1	Z-BUS Low Byte	2-Wire Handshake
4	0	1	0	0	Z-BUS High Byte	Z-BUS High Byte
5	0	1	0	1	Z-BUS High Byte	Non-Z-BUS
6	0	1	1	0	Z-BUS High Byte	3-Wire Handshake
7	0	1	1	1	Z-BUS High Byte	2-Wire Handshake
8	1	0	0	0	Non-Z-BUS	Z-BUS Low Byte
9	1	0	0	1	Non-Z-BUS	Non-Z-BUS
10	1	0	1	0	Non-Z-BUS	3-Wire Handshake
11	1	0	1	1	Non-Z-BUS	2-Wire Handshake

*Bits 3 and 2 of Control register 0. Read/Write from Port 1, Read-only from Port 2.

00867B

Refer to page 7-1 for Essential Information on Military Devices

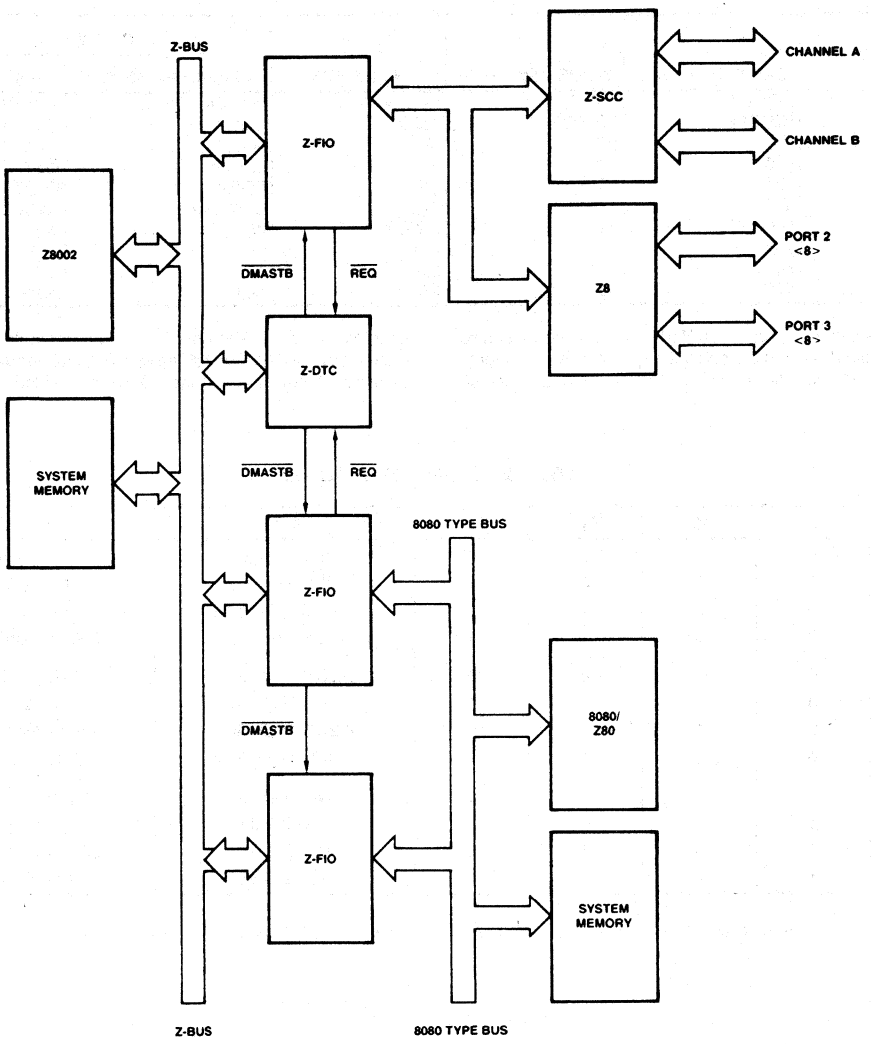


Figure 2. CPU to CPU Configuration

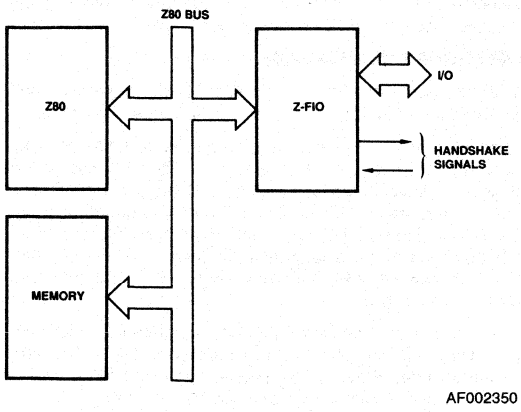


Figure 3. CPU to I/O Configuration

CPU INTERFACES

The FIO is designed to work with both Z-BUS and non-Z-BUS-type CPUs, on both Port 1 and Port 2. The Z-BUS configuration interfaces CPUs with time-multiplexed address and data information on the same pins. The Z8001, Z8002, and Z8 are examples of this type of CPU. The AS (Address Strobe) pin is used to latch the address and chip select information sent out by the CPU. The R/$\overline{W}$ (Read/Write) pin and the $\overline{DS}$ (Data Strobe) pin are used for timing reads and writes from the CPU to the FIO (Figures 4 and 5).

The non-Z-BUS configuration is used for CPUs where the address and data buses are separate. Examples of this type of CPU are the Z80 and 8080. The $\overline{RD}$ (Read) and $\overline{WR}$ (Write) pins are used to time reads and writes from the CPU to the FIO (Figures 7 and 8). The C/$\overline{D}$ (Control/Data) pin is used to directly access the FIFO buffer (C/$\overline{D}$ = 0) and to access the other registers (C/$\overline{D}$ = 1). Read and write to all registers except the FIFO buffer[1] are the two-step operations, described as follows (Figure 6). First, write the address of the register to be accessed with C/$\overline{D}$ = 1. The address goes into a pointer register, and the FIO switches to state 1. The next read or write with C/$\overline{D}$ = 1 will be to the register pointed to. Continuous status monitoring can be performed by continuous Control Read operations (C/$\overline{D}$ = 1).

[1]The FIFO buffer can also be accessed by this two-step operation.

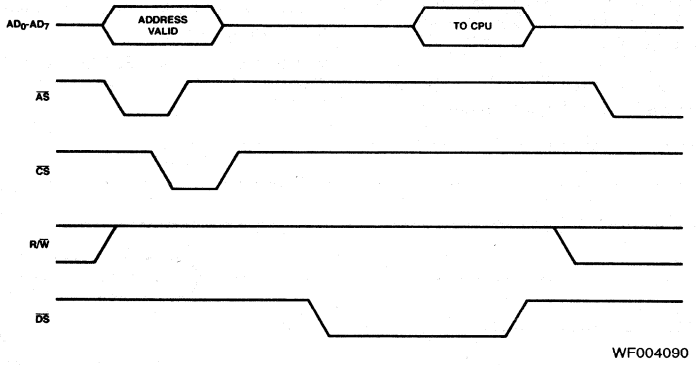


Figure 4. Z-BUS Read Cycle Timing

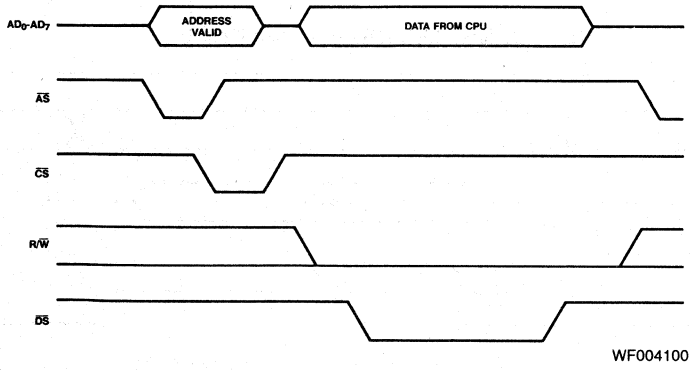


Figure 5. Z-BUS Write Cycle Timing

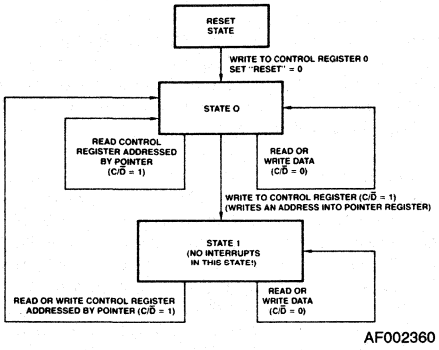


Figure 6. In Non-Z-BUS Mode, Control Registers are Accessed by First Writing the Address of the Register, then Reading or Writing the Contents of the Addressed Register.

WAIT OPERATION

When data is output from the CPU, the REQ/WT/ (WAIT) pin is active (LOW) only when the FIFO buffer is full, the chip is selected, and FIFO buffer is addressed. WAIT goes inactive when the FIFO buffer is not full.

When data is input to the CPU, the REQ/WT pin becomes active (LOW) only when the FIFO buffer is empty, the chip is selected, and the FIFO buffer is addressed. WAIT goes inactive when the FIFO buffer is not empty.

INTERRUPT OPERATION

The FIO supports Zilog's prioritized daisy chain interrupt protocol for both Z-BUS and non-Z-BUS operating modes.

Each side of the FIO has seven sources of interrupt. The priorities of these devices are fixed in the following order (highest to lowest): Mailbox message, Change in Data Direction, Pattern Match, Status Match, Overflow/Underflow Error, Buffer Full, and Buffer Empty. Each interrupt source has three bits that control how it generates the interrupt. These bits are Interrupt Pending (IP), Interrupt Enable (IE) and Interrupt Under Service (IUS).

In addition, each side of the FIO has an interrupt vector and four bits controlling the FIO interrupt logic. These bits are Vector Includes Status (VIS), Master Interrupt Enable (MIE), Disable Lower Chain (DLC) and No Vector (NV).

A typical Interrupt Acknowledge cycle for Z-BUS operation is shown in Figure 9 and for non-Z-BUS operation in Figure 10. The only difference is that in Z-BUS mode, INTACK is latched by AS, and in non-Z-BUS mode, INTACK is not latched.

When MIE = 1, reading the vector always includes status, independent of the state of the VIS bit. In this way, when VIS = 0, all information can be obtained with one additional read, thus conserving vector space. When MIE = 0, reading the vector register returns the unmodified base vector so that it can be verified.

In non-Z-BUS mode, IPs do not get set while in State 1. Therefore, to minimize interrupt latency, the FIO should be left in State 0.

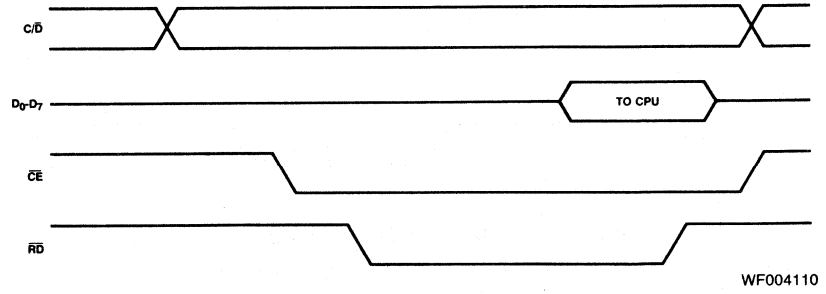


Figure 7. Non-Z-BUS Read Cycle Timing

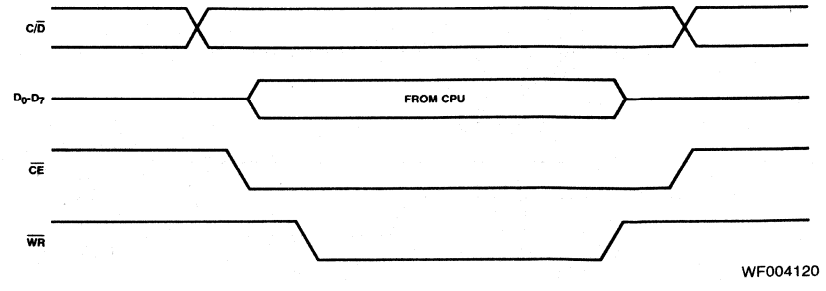


Figure 8. Non-Z-BUS Write Cycle Timing

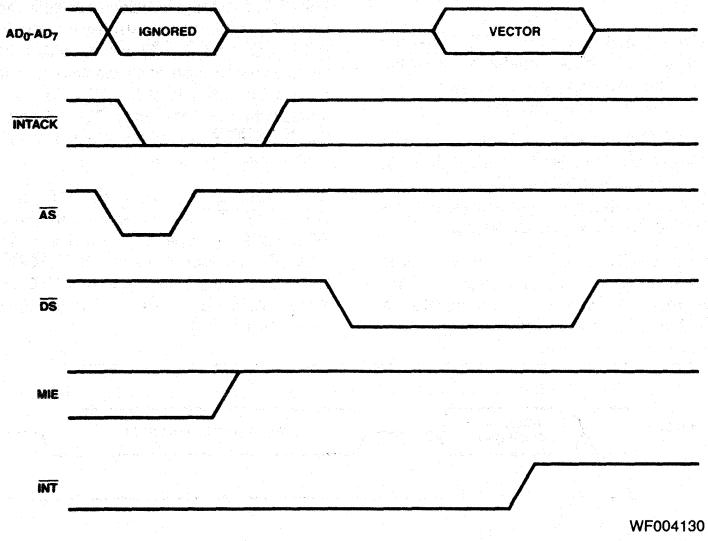


Figure 9. Z-BUS Interrupt Acknowledge Cycle

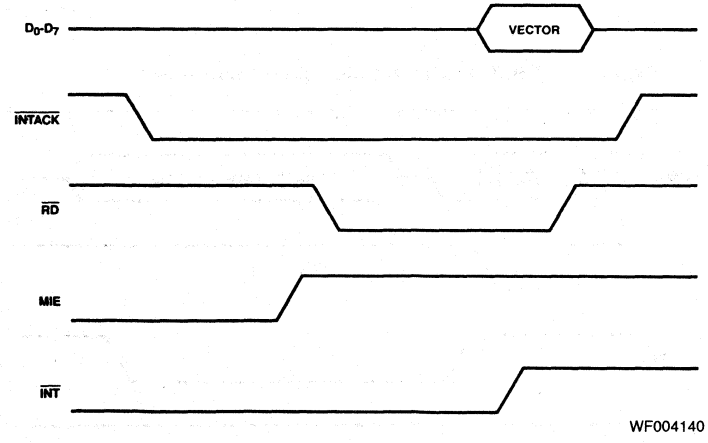


Figure 10. Non-Z-BUS Interrupt Acknowledge Cycle

CPU TO CPU OPERATIONS

DMA Operation

The FIO is particularly well suited to work with a DMA in both Z-BUS and non-Z-BUS modes. A data transfer between the FIO and system memory can take place during every machine cycle on both sides of the FIO simultaneously.

In Z-BUS mode, the $\overline{\text{DMASTB}}$ pin (DMA Strobe) is used to read or write into the FIFO buffer. The R/$\overline{\text{W}}$ (Read/Write) and $\overline{\text{DS}}$ (Data Strobe) signals are ignored by the FIO; however, the $\overline{\text{CS}}$ (Chip Select) signal is not ignored and therefore must be kept valid. Figures 11 and 12 show typical timing.

In Non-Z-BUS mode, the $\overline{\text{DACK}}$ pin (DMA Acknowledge) is used to tell the FIO that its DMA request is granted. After $\overline{\text{DACK}}$ goes Low, every read or write to the FIO goes into the FIFO buffer. Figures 13 and 14 show typical timing.

The FIO provides a special mode to enhance its DMA transfer capability. When data is written into the FIFO buffer, the $\overline{\text{REQ}}$/$\overline{\text{WT}}$ ($\overline{\text{REQUEST}}$/$\overline{\text{WAIT}}$) pin is active (LOW) until the FIFO buffer is full. It then goes inactive and stays inactive until the number of bytes in the FIFO buffer is equal to the value programmed into the Byte Count Comparison register. Then the $\overline{\text{REQUEST}}$ signal goes active and the sequence starts over again (Figure 15).

When data is read from the FIO, the $\overline{\text{REQ}}$/$\overline{\text{WT}}$ pin ($\overline{\text{REQUEST}}$/$\overline{\text{WAIT}}$) is inactive until the number of bytes in the FIFO buffer is equal to the value programmed in the Byte Count Comparison register. The $\overline{\text{REQUEST}}$ signal then goes active and stays active until the FIFO buffer is empty. When empty, $\overline{\text{REQUEST}}$ goes inactive and the sequence starts over again (Figure 16).

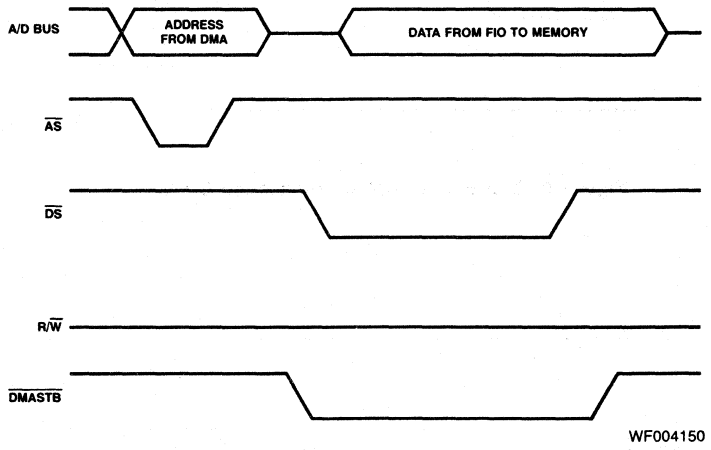


Figure 11. Z-BUS FIO to Memory Data Transaction

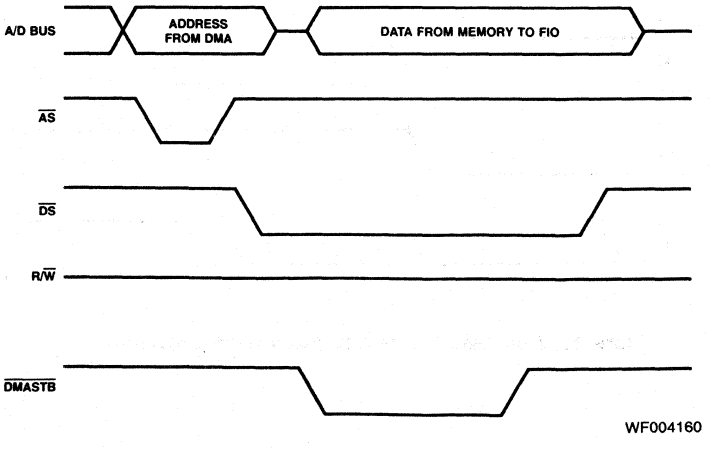


Figure 12. Z-BUS Memory to FIO Data Transaction

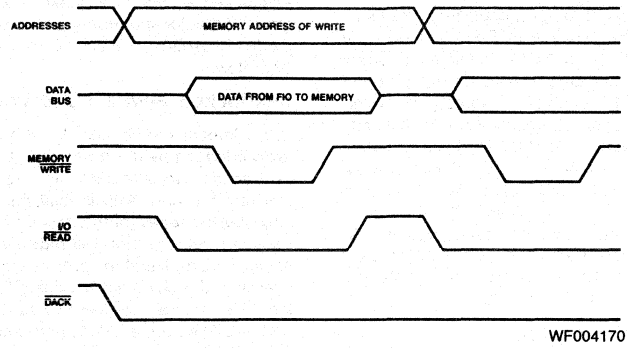


Figure 13. Non-Z-BUS FIO to Memory Transaction

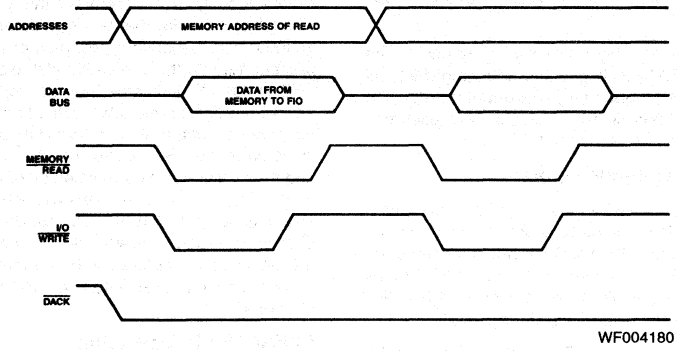
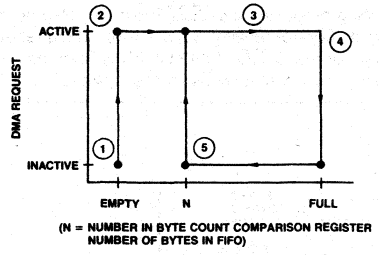
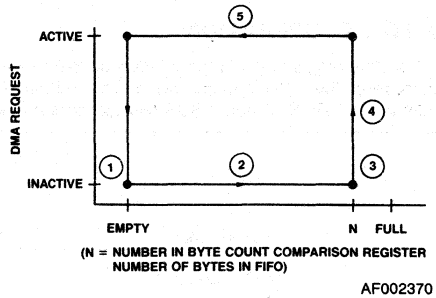


Figure 14. Non-Z-BUS Memory to FIO Data Transaction



Notes: 1. FIFO empty.
2. REQUEST enabled, FIO request DMA transfer.
3. DMA transfers data into the FIO.
4. FIFO full, REQUEST inactive.
5. The FIFO empties from the opposite port until the number of bytes in the FIFO buffer is the same as the number programmed in the Byte Count Comparison register.

Figure 15. Byte Count Control: Write to FIO



Notes: 1. FIFO empty.
2. CPU/DMA fills FIFO buffer from the opposite port.
3. Number of bytes in FIFO buffer is the same as the number of bytes programmed in the Byte Count Comparison register.
4. REQUEST goes active.
5. DMA transfers data out of FIFO until it is empty.

Figure 16. Byte Count Control: Read from FIO

Message Registers

Two CPUs can communicate through a dedicated "mailbox" register without involving the 128 x 8 bit FIFO buffer (Figure 19). This mailbox approach is useful for transferring control parameters between the interfacing devices on either side of the FIO without using the FIFO buffer. For example, when Port 1's CPU writes to the Message Out register, Port 2's message IP is set. If interrupts are enabled, Port 2's CPU is interrupted. Port 2's message status is readable from the Port 1 side via Control register 2. When Port 2's CPU reads the data from its Message In register, the Port 2 IP is cleared. Thus, Port 1's CPU can tell that the message has been read and can now send another message or follow whatever protocol that is set up between the two CPU's. The same transfer can also be made from Port 2's CPU to Port 1's CPU.

CLEAR (Empty) FIFO Operation

The $\overline{\text{CLEAR}}$ FIFO bit (active LOW) clears the FIFO buffer of data. Writing a 0 to this bit empties the FIFO buffer, inactivates the $\overline{\text{REQUEST}}$ line, and disables the handshake (if programmed). The $\overline{\text{CLEAR}}$ bit does not affect any control or data register. To remove the $\overline{\text{CLEAR}}$ state, write a 1 to the $\overline{\text{CLEAR}}$ bit.

In CPU/CPU mode, under program control, only one of the ports can empty the FIFO by writing to its Control register 3, bit 6. The Port 1 CPU must program bit 7 in Control register 3 to determine which port controls the CLEAR FIFO operation (0 = Port 1 control; 1 = Port 2 control).

Direction of Data Transfer Operation

The Data Direction bit controls the direction of data transfer in the FIFO buffer. The Data Direction bit is defined as 0 = output from CPU and 1 = input to CPU. This bit reads correctly when read by either port's CPU. For example, if Port 1's CPU reads a 0 (CPU output) in its Data Direction bit, then Port 2's CPU reads a 1 (input to CPU) in its Data Direction bit.

In CPU/CPU mode, under program control, only one of the ports can control the direction of data transfer. The Port 1 CPU must program bit 5 in Control register 3 to determine which port controls the data direction (0 = Port 1 control; 1 = Port 2 control). Figure 18 shows FIO data transfer options.

CPU TO I/O OPERATION

When Port 2 is programmed in the Interlocked 2-Wire Handshake mode or the 3-Wire Handshake mode, and Port A is programmed in Z-BUS or non-Z-BUS Microprocessor mode, the FIO interfaces a CPU and a peripheral device. In the interlocked 2-Wire Handshake mode, RFD/$\overline{\text{DAV}}$ and $\overline{\text{ACKIN}}$ strobe data to and from Port 2. In the 3-Wire Handshake mode, RFD/$\overline{\text{DAV}}$, $\overline{\text{DAV}}$/DAC, and DAC/RFD signal control data flow.

Interlocked 2-Wire Handshake

In the Interlocked Handshake, the action of the FIO must be acknowledged by the other half of the handshake before the next action can take place. In output mode, Port 2 does not indicate that new data is available until the external device indicates it is ready for the data. Similarly, in input mode, Port 2 does not indicate that it is ready for new data until the data source indicates that the previous byte of the data is no longer available, thereby acknowledging Port 2's acceptance of the last byte. This allows the FIO to directly interface to a Z8's port, a CIO's port, a UPC's port, another FIO port, or another FIFO Z8060, with no external logic (Figures 19 and 20).

3-Wire Handshake

The 3-Wire Handshake is designed for applications in which one output port is communicating with many input ports simultaneously. It is essentially the same as the Interlocked Handshake, except that two signals are used to indicate that an input port is ready for new data or that it has accepted the present data. In the 3-Wire Handshake, the rising edge of the RFD status line indicates that the port is ready for data, and the rising edge of the DAC status line indicates that the data has been accepted. With 3-Wire Handshake, the lines of many input ports can be bussed together with open-drain drivers, and the output port knows when all of the ports are ready and have accepted the data. This handshake is the same handshake used in the IEEE-488 Instruments. Since the port's direction can be changed under software control, bidirectional IEEE-488 type transfers can be performed. Figures 21 and 22 show the timings associated with 3-Wire Handshake communications.

CLEAR FIFO Operation

In CPU-to-I/O operation, the CLEAR FIFO operation can be performed by the CPU side (Port 1) under software control as previously explained. The CLEAR FIFO operation can also be performed under hardware control by defining the $\overline{\text{CLEAR}}$ pin of Port 2 as an input (Control register 3, bit 7 = 1).

For cascading purposes, the $\overline{\text{CLEAR}}$ pin can also be defined as an output (Control register 3, bit 7 = 0), which reflects the current state of the $\overline{\text{CLEAR}}$ FIFO bit. It can then empty other FIOs or initialize other devices in the system.

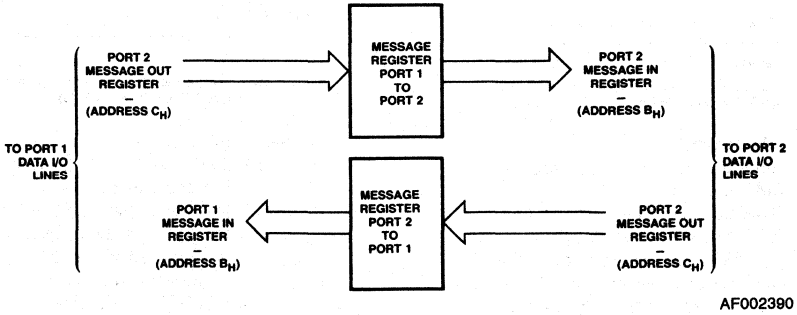


Figure 17. Message Register Operation

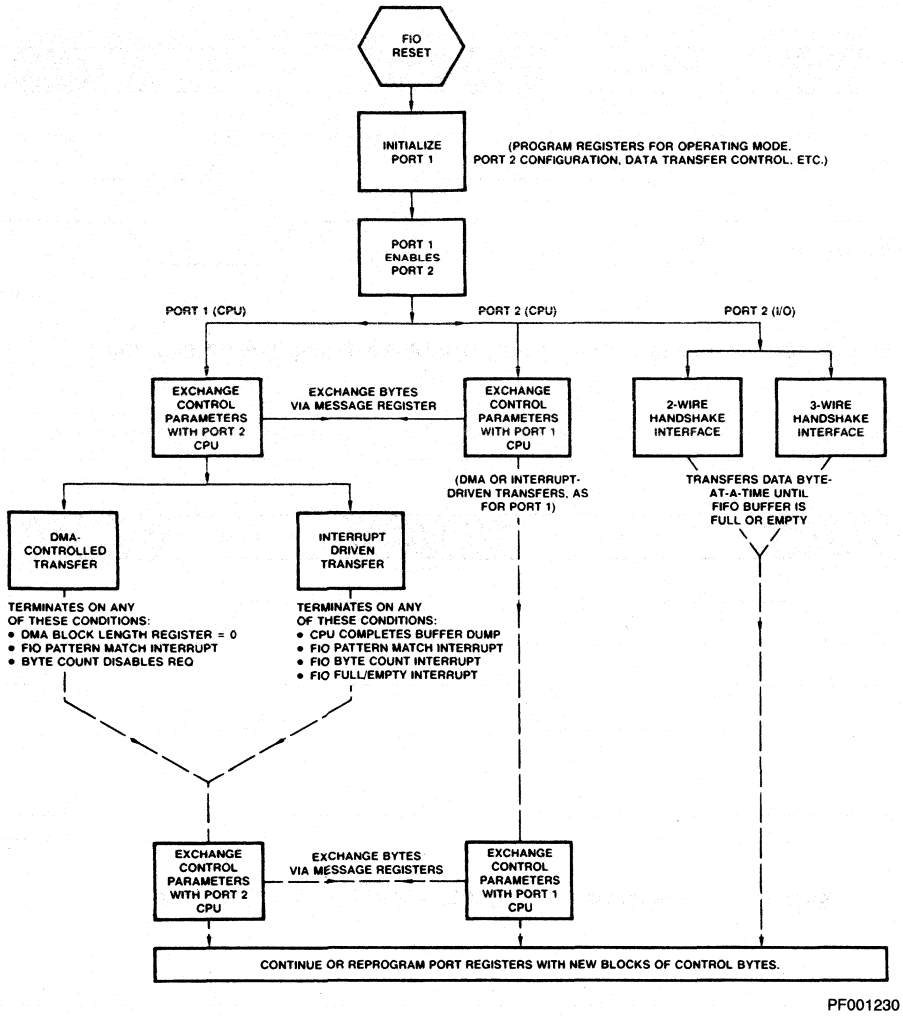


Figure 18. FIO Data Transfer Options

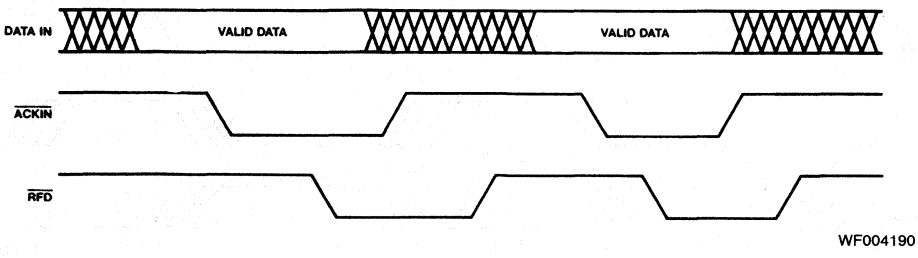


Figure 19. Input Timing for Interlocked Handshake Timing (Port 2 Side Only)

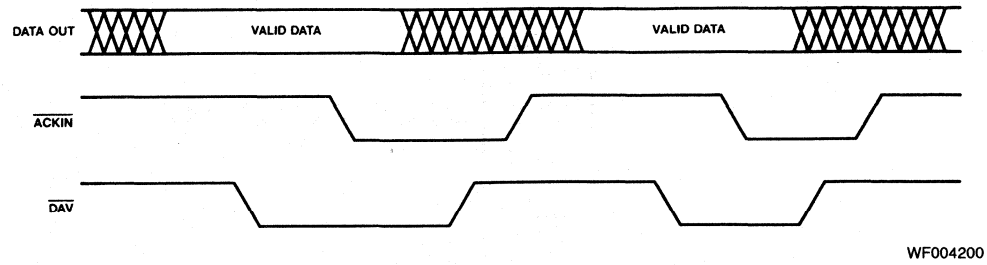


Figure 20. Output Timing for Interlocked Handshake Timing (Port 2 Side Only)

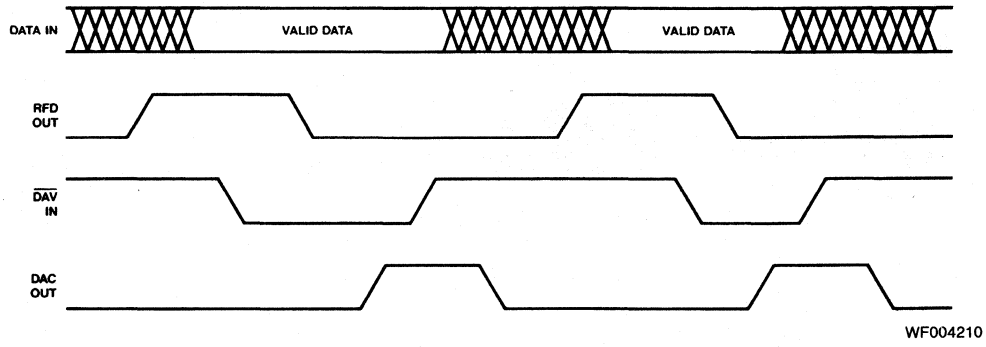


Figure 21. Input (Acceptor) Timing IEEE-488 HS (Port 2 Side Only)

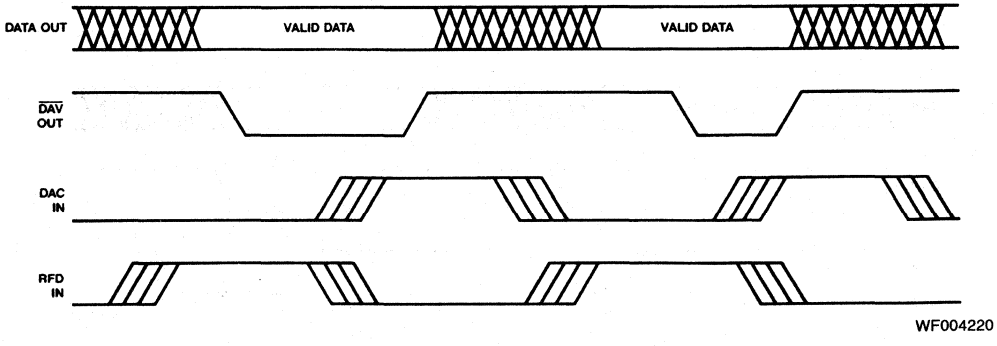


Figure 22. Output (Source) Timing IEEE-488 HS Port (Port 2 Side Only)

Data Direction Control

In CPU-to-I/O mode, the direction of data transfer can be controlled by the CPU side (Port 1) under software control as previously explained. The data direction can also be determined by hardware control by defining the Data Direction pin of Port 2 as an input (Control register 3, bit 5 = 1).

For cascading purposes, the Data Direction pin can also be defined as an output (Control register 3, bit 5 = 0) pin which reflects the current state of the Data Direction bit. It can then be used to control the direction of data transfer for other FIOs or for external logic.

On the Port 2 side, when data direction is 0, Port 2 is in Output Handshake mode. When data direction is 1, Port 2 is in Input Handshake mode.

PROGRAMMING INFORMATION

The Programming of the FIO is greatly simplified by the efficient grouping of the various operation modes in the control registers. Since all of the control registers are read/write, the need for maintaining their image in system memory is eliminated. Also, the read/write feature of the registers aids in system debugging.

Each side of the FIO has 16 registers. All 16 registers are used by the Port 1 side; Control register 2 is not used on the Port 2 side. All registers are addressable O_H through F_H.

In the Z-BUS Low Byte mode, the FIO allows two methods for register addressing under control of the Right Justify Address (RJA) bit in Control register 0. When RJA = 0, address bus bits 1 – 4 are used for register addressing and bits 0, 5, 6 and 7 are ignored (Table 3). When RJA = 1, bits 0 – 3 are used for the register addresses and bits 4 – 7 are ignored.

Control Registers

These four registers specify FIO operation. The Port 2 side control registers operate only if the Port 2 device is a CPU. The Port 2 CPU can control interface operations, including data direction, only when enabled by the setting of bit 0 in the Port 1 side of Control register 2. A 1 in bit 1 of the same register enables the handshake logic.

Interrupt Status Registers

These four registers control and monitor the priority interrupt functions for the FIO.

Interrupt Vector Register

This register stores the interrupt service routine address. This vector is placed on $D_0 - D_7$ when IUS is set by the Interrupt Acknowledge signal from the CPU. When bit 4 (Vector Includes Status) is set in Control register 0, the reason for the Interrupt is encoded within the vector address in bits 1, 2 and 3. If bit 5 is set in Control register 0, no vector is output by the FIO during an Interrupt Acknowledge cycle. However, IUS is set as usual.

Byte Count Compare Register

This register contains a value compared with the byte count in the Byte Count register. If the Byte Count Compare interrupt is enabled, an interrupt will occur upon compare.

Message Out Register

Either CPU can place a message in its Message Out register. If the opposite side Message register interrupt is enabled, the receiving side CPU will receive an interrupt request, advising that a message is present in its Message In register. Bit 5 in Control register 1 on the initiating side is set when a message is written. It is cleared when the Byte Count register read is completed.

TABLE 3. FIO REGISTER ADDRESS SUMMARY

Non-Z-Bus		D_7-D_4	D_3	D_2	D_1	D_0	
Z-BUS High Byte			A_3	A_2	A_1	A_0	
Z-BUS Low Byte	RJA = 0	AD_7-AD_5	AD_4	AD_3	AD_2	AD_1	AD_0
	RJA = 1	AD_7-AD_4	AD_3	AD_2	AD_1	AD_0	
Description							
Control Register 0		X	0	0	0	0	X
Control Register 1		X	0	0	0	1	X
Interrupt Status Register 0		X	0	0	1	0	X
Interrupt Status Register 1		X	0	0	1	1	X
Interrupt Status Register 2		X	0	1	0	0	X
Interrupt Status Register 3		X	0	1	0	1	X
Interrupt Vector Register		X	0	1	1	0	X
Byte Count Register		X	0	1	1	1	X
Byte Count Comparison Register		X	1	0	0	0	X
Control Register 2*		X	1	0	0	1	X
Control Register 3		X	1	0	1	0	X
Message Out Register		X	1	0	1	1	X
Message In Register		X	1	1	0	0	X
Pattern Match Register		X	1	1	0	1	X
Pattern Mask Register		X	1	1	1	0	X
Data Buffer Register		X	1	1	1	1	X

X = Don't Care
*Register is only on Port 1 side

Message in Register

This register receives a message placed in the Message Out register by the opposite side CPU.

Pattern Match Register

This register contains a bit pattern matched against the byte in the Data Buffer register. When these patterns match, a Pattern Match interrupt will be generated, if previously enabled.

Pattern Mask Register

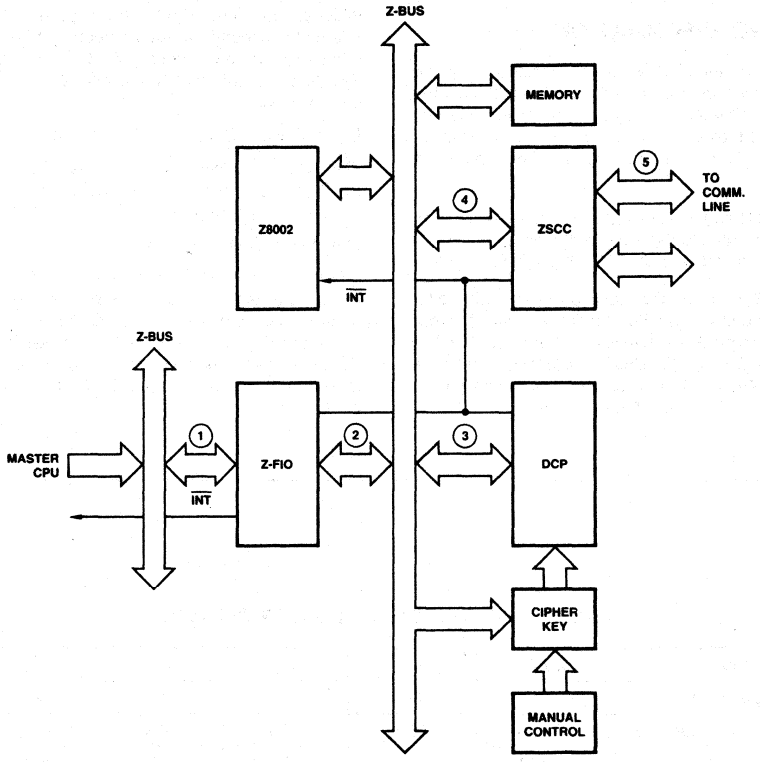
The Pattern Mask register may be programmed with a bit pattern mask that limits comparable bits in the Pattern Match register to non-masked bits (1 = mask).

Data Buffer Register

This register contains the data to be read from or written to the FIFO buffer.

Byte Count Register

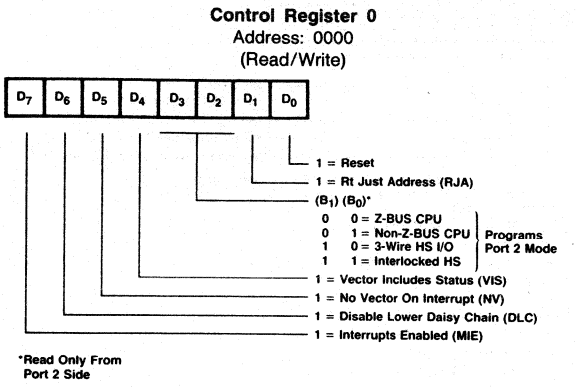
This is a read-only register, containing the byte count for the FIFO buffer. The byte count is derived by subtracting the number of bytes read from the buffer from the number of bytes written into the buffer. The count is copied into a holding register for an accurate reading by setting bit 6 (Freeze Status register) in Control register 1. This bit is cleared when the Byte Count register is completed.

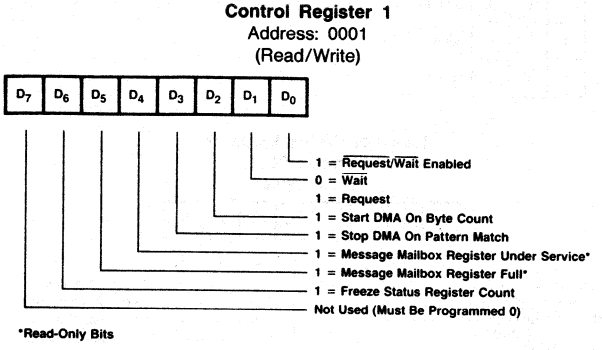


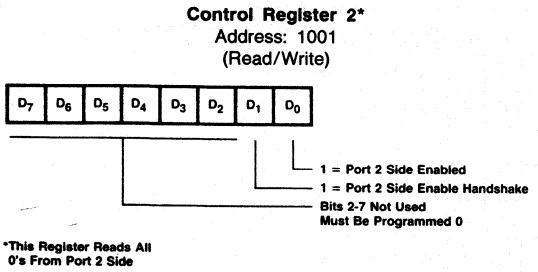
DATA FLOW: 1. Data from master CPU → Z-FIO Port 2.
2. Z-FIO Port 1 → DCP.
3. DCP → RAM.
4. RAM → Z-SCC.
5. Z-SCC → data comm. line loop.

Figure 23. Typical Application: Node Controller

REGISTERS







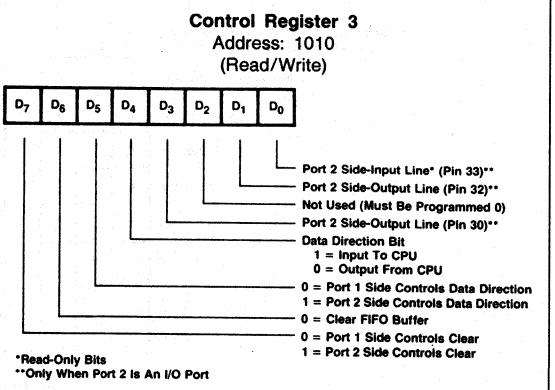
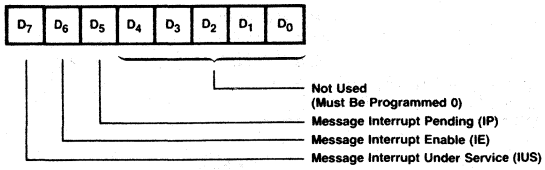


Figure 24. Control Registers

REGISTERS (Cont.)

Interrupt Status Register 0

Address: 0010
(Read/Write)



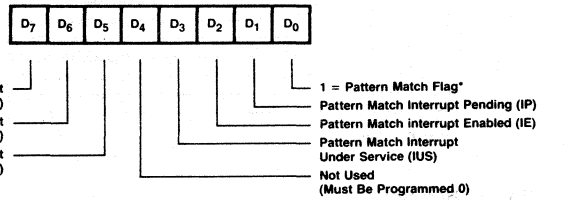
Messages IUS, IE, and IP are Written Using The Following Commands: ($D_4 - D_0 = 0$)

D_7	D_6	D_5	
0	0	0	Null Code (No Change)
0	0	1	Clear IP & IUS
0	1	0	Set IUS
0	1	1	Clear IUS
1	0	0	Set IP
1	0	1	Clear IP
1	1	0	Set IE
1	1	1	Clear IE

DF001411

Interrupt Status Register 1

Address: 0011
(Read/Write)



Directions IUS, IE, and IP are Written Using The Following Commands: (D_0 and $D_4 = 0$)

	D_7	D_6	D_5
Null Code (No Change)	0	0	0
Clear IP & IUS	0	0	1
Set IUS	0	1	0
Clear IUS	0	1	1
Set IP	1	0	0
Clear IP	1	0	1
Set IE	1	1	0
Clear IE	1	1	1

Pattern Matches IUS, IE, and IP are Written Using The Following Commands: (D_0 and $D_4 = 0$)

D_3	D_2	D_1	
0	0	0	Null Code (No Change)
0	0	1	Clear IP & IUS
0	1	0	Set IUS
0	1	1	Clear IUS
1	0	0	Set IP
1	0	1	Clear IP
1	1	0	Set IE
1	1	1	Clear IE

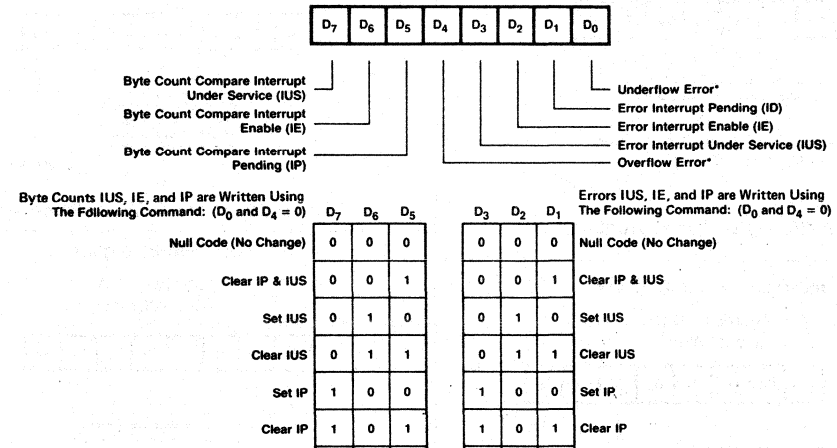
*Read only bits

DF001421

Figure 25. Interrupt Status Registers

REGISTERS (Cont.)

Interrupt Status Register 2
Address: 0100
(Read/Write)



Byte Counts IUS, IE, and IP are Written Using The Following Command: (D_0 and D_4 = 0)

	D_7	D_6	D_5
Null Code (No Change)	0	0	0
Clear IP & IUS	0	0	1
Set IUS	0	1	0
Clear IUS	0	1	1
Set IP	1	0	0
Clear IP	1	0	1
Set IE	1	1	0
Clear IE	1	1	1

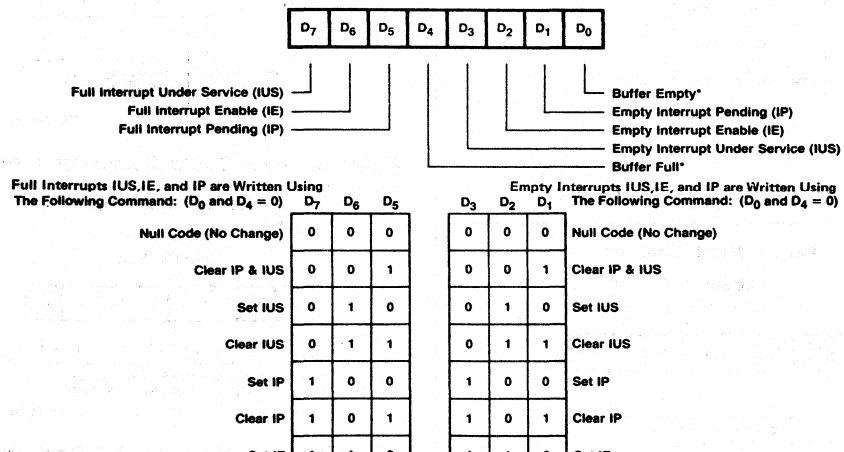
Errors IUS, IE, and IP are Written Using The Following Command: (D_0 and D_4 = 0)

D_3	D_2	D_1	
0	0	0	Null Code (No Change)
0	0	1	Clear IP & IUS
0	1	0	Set IUS
0	1	1	Clear IUS
1	0	0	Set IP
1	0	1	Clear IP
1	1	0	Set IE
1	1	1	Clear IE

*Read-Only Bits

DF001431

Interrupt Status Register 3
Address: 0101
(Read/Write)



Full Interrupts IUS,IE, and IP are Written Using The Following Command: (D_0 and D_4 = 0)

	D_7	D_6	D_5
Null Code (No Change)	0	0	0
Clear IP & IUS	0	0	1
Set IUS	0	1	0
Clear IUS	0	1	1
Set IP	1	0	0
Clear IP	1	0	1
Set IE	1	1	0
Clear IE	1	1	1

Empty Interrupts IUS,IE, and IP are Written Using The Following Command: (D_0 and D_4 = 0)

D_3	D_2	D_1	
0	0	0	Null Code (No Change)
0	0	1	Clear IP & IUS
0	1	0	Set IUS
0	1	1	Clear IUS
1	0	0	Set IP
1	0	1	Clear IP
1	1	0	Set IE
1	1	1	Clear IE

*Read-Only Bits

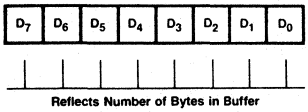
DF001441

Figure 25. Interrupt Status Registers (Cont.)

REGISTERS (Cont.)

Byte Count Register

Address: 0111



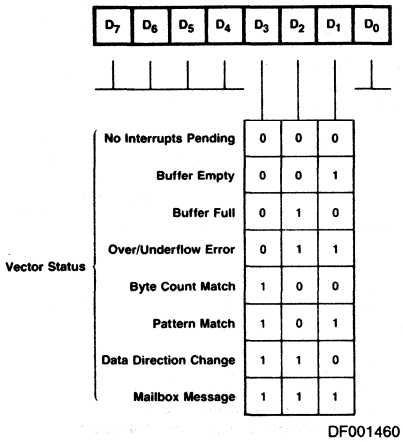
(Copied from actual Byte Counter by setting bit 6 of CR1.)

DF001450

Figure 26. Byte Count Register

Interrupt Vector Register

Address: 0110
(Read/Write)

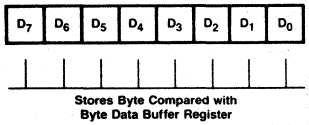


DF001460

Figure 27. Interrupt Vector Register

Pattern Match Register

Address: 1101
(Read/Write)

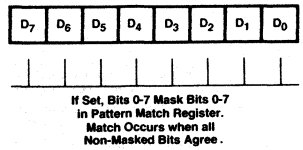


DF001470

Figure 28. Pattern Match Register

Pattern Mask Register

Address: 1110
(Read/Write)

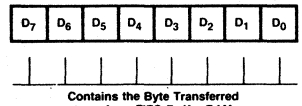


DF001481

Figure 29. Pattern Mask Register

Data Buffer Register

Address: 1111
(Read/Write)

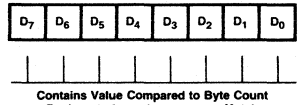


DF001490

Figure 30. Data Buffer Register

Byte Count Comparison Register

Address: 1000
(Read/Write)

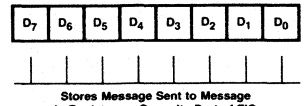


DF001500

Figure 31. Byte Count Comparison Register

Message Out Register

Address: 1011
(Read/Write)

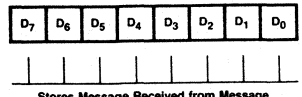


DF001510

Figure 32. Message Out Register

Message In Register

Address: 1100
(Read Only)



DF001520

Figure 33. Message In Register

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin Relative to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 1.75W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	4MHz	6MHz
	Z8038	Z8038A
Commercial Operating Range T_A = 0 to +70°C V_{CC} = 5V ±5% V_{SS} = 0V	Z8038DC Z8038PC	Z8038ADC Z8038APC
Industrial Operating Range T_A = −40 to +85°C V_{CC} = 5V ±10% V_{SS} = 0V	Z8038DI	

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (Note 1)

Parameters	Description	Test Conditions		Min	Typ	Max	Units
V_{IL}	Input LOW Voltage			−0.5		+.8	Volts
V_{IH}	Input HIGH Voltage	Standard Temp		2.0		V_{CC}	Volts
V_{OL}	Output LOW Voltage	I_{OL} = 3.2mA				0.5	Volts
		I_{OL} = 2.0mA				0.4	
V_{OH}	Output HIGH Voltage	I_{OH} = −250μA		2.4			Volts
I_{OZL}	Output Leakage Current	V_{OUT} = 0.4V				10	μA
I_{OZH}	Output Leakage Current	V_{OUT} = V_{CC}				10	μA
I_I	Input Leakage Current					±10	μA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.				10	pF
$C_{I/O}$	I/O Capacitance					20	pF
C_{OUT}	Output Capacitance					15	pF
I_{CC}	Power Supply Current	V_{CC} = MAX	T_A = 0°C			200	mA
			T_A = −55°C			250	

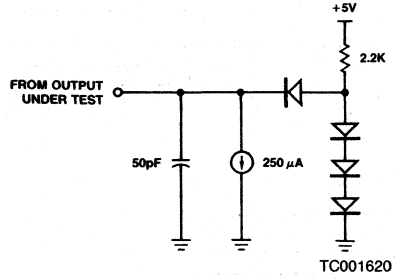
Note: 1. See table for operating range. Typical conditions apply at T_A = 25°C, V_{CC} = 5.0V.

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

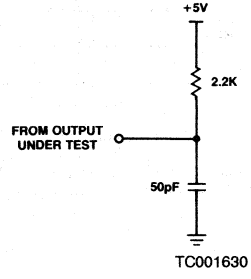
$+4.75V \leqslant V_{CC} \leqslant +5.25V$
GND = 0V
$0°C \leqslant T_A \leqslant +70°C$

Standard Test Load



TC001620

Open-Drain Test Load



TC001630

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z-BUS CPU INTERFACE TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes*
			Min	Max	Min	Max		
1	TwAS	$\overline{AS}$ Low Width	70		50		ns	
2	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time	30		10		ns	1
3	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time	50		30		ns	1
4	TsCSO(AS)	$\overline{CS}$ to $\overline{AS}$ ↑ Setup Time	0		0		ns	1
5	ThCSO(AS)	$\overline{CS}$ to $\overline{AS}$ ↑ Hold Time	60		40		ns	1
6	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↑ Delay	60		40		ns	1
7	TsA(DS)	Address to $\overline{DS}$ ↓	120		100		ns	
8	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Setup Time	100		80		ns	
9	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Setup Time	0		0		ns	
10	TwDS	$\overline{DS}$ Low Width	390		250		ns	
11	TsDW(DSf)	Write Data to $\overline{DS}$ ↓ Setup Time	30		20		ns	
12	TdDS(DRV)	$\overline{DS}$ (Read) ↓ to Address Data Bus Driven	0		0		ns	
13	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	ns	
14	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		20		ns	
15	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		ns	
16	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay		70		45	ns	2
17	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		ns	
18	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		ns	
19	Trc	Valid Access Recovery Time	1000		650		ns	3

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is measured up to the time when the output has changed 0.5V from steady state with minimum AC load and maximum DC load.
3. This is the delay from $\overline{DS}$ of one CIO access to $\overline{DS}$ of another FIO access (either read or write).
*All timings are preliminary and subject to change. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0."

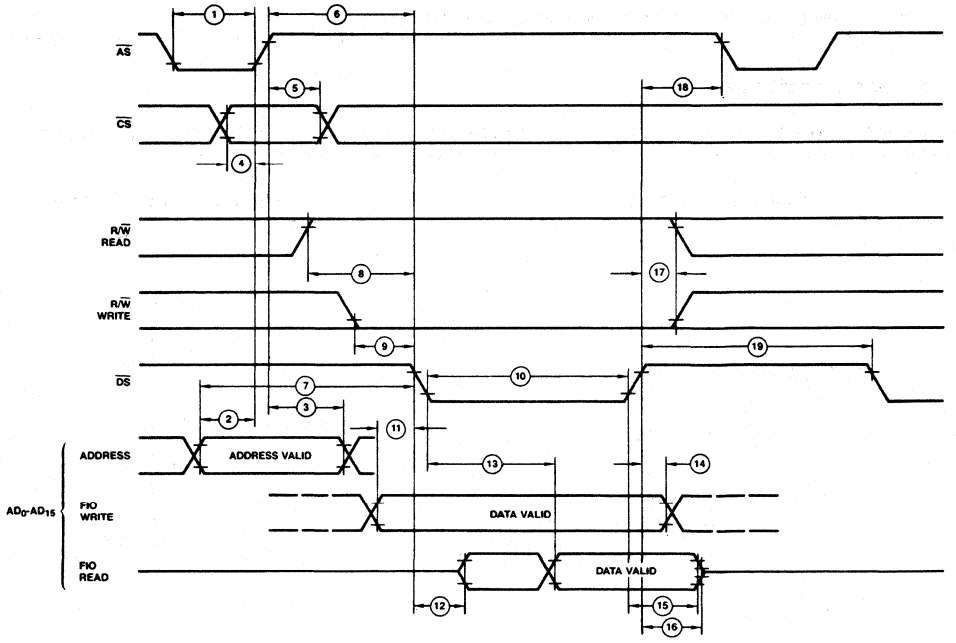


Figure 34. Z-BUS CPU Interface Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z-BUS CPU INTERRUPT ACKNOWLEDGE TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
20	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	0		0		ns	
21	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		250		ns	
22	TdDSA(DR)	$\overline{DS}$ (Acknowledge) ↓ to Read Data Valid Delay		250		180	ns	
23	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		250		ns	
24	TdAS(IEO)	$\overline{AS}$ ↑ to IEO ↓ Delay ($\overline{INTACK}$ Cycle)		350		250	ns	4
25	TdIEI(IEO)	IEI to IEO Delay		150		100	ns	4
26	TsIEI(DSA)	IEI to $\overline{DS}$ (Acknowledge) ↓ Setup Time	100		70		ns	
27	ThIEI(DSA)	IEI to $\overline{DS}$ (Acknowledge) ↑ Hold Time	50		30		ns	4
28	TdDS(INT)	$\overline{DS}$ ($\overline{INTACK}$ Cycle) to $\overline{INT}$ Delay		800		800	ns	
29	TdDCST	Interrupt Daisy Chain Settle Time					ns	4

Note: 4. The parameters for the devices in any particular daisy chain must meet the following constraints: the delay from $\overline{AS}$ to $\overline{DS}$ must be greater than the sum of TdAS(IEO) for the highest priority peripheral, TsIEI(DSA) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.

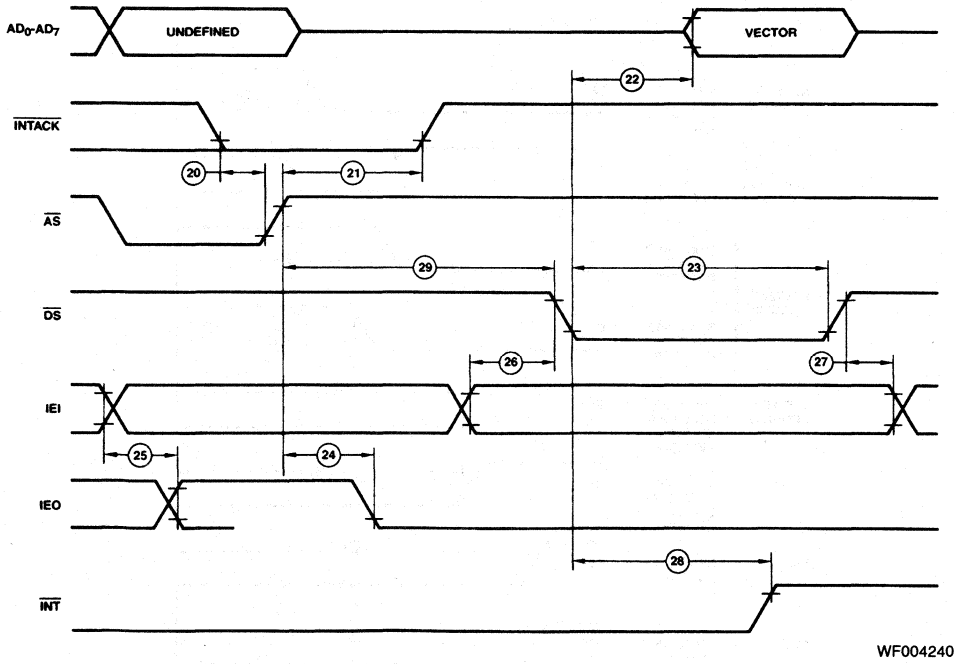


Figure 35. Z-Bus CPU Interrupt Acknowledge Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z-BUS INTERRUPT TIMING

Number	Parameters	Description	4MHz Min	4MHz Max	6MHz Min	6MHz Max	Units	Notes
30	TdMW(INT)	Message Write to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	5
31	TdDC(INT)	Data Direction Change to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	6
32	TdPMW(INT)	Pattern Match to $\overline{INT}$ Delay (Write Case)		1		1	$\overline{AS}$ Cycles + ns	
33	TdPMR(INT)	Pattern Match (Read Case) to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	
34	TdSC(INT)	Status Compare to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	6
35	TdER(INT)	Error to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	
36	TdEM(INT)	Empty to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	6
37	TdFL(INT)	Full to $\overline{INT}$ Delay		1		1	$\overline{AS}$ Cycles + ns	6
38	TdAS(INT)	$\overline{AS}$ to $\overline{INT}$ Delay					$\overline{AS}$ Cycles + ns	

Notes: 5. Write is from the other side of FIO.
6. Write can be from either side, depending on programming of FIO.

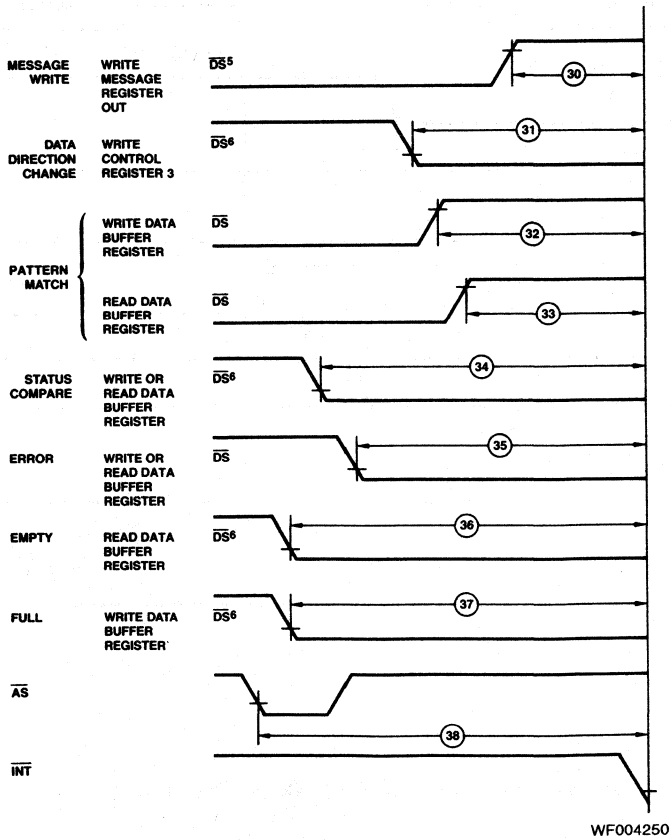


Figure 36. Z-Bus Interrupt Timing

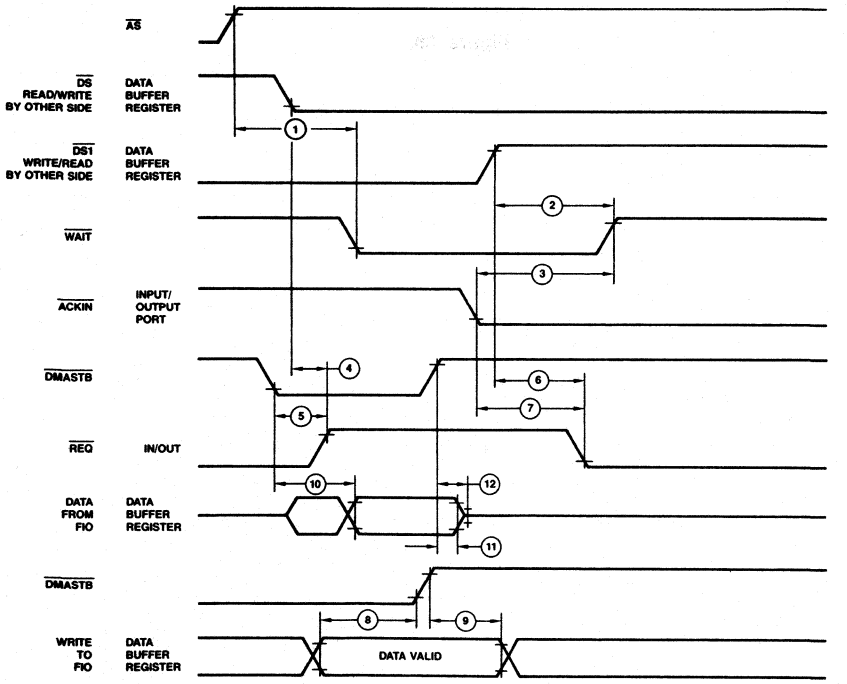
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z-BUS REQUEST/WAIT TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
1	TdDS(WAIT)	AS ↑ to WAIT ↓ Delay		190		160	ns	
2	TdDS1(WAIT)	DS1 ↑ to WAIT ↑ Delay		1000		1000	ns	
3	TdACK(WAIT)	ACKIN ↓ to WAIT ↑ Delay		1000		1000	ns	1
4	TdDS(REQ)	DS ↓ to REQ ↑ Delay		350		300	ns	
5	TdDMA(REQ)	DMASTB ↓ to REQ ↑ Delay		350		300	ns	
6	TdDS1(REQ)	DS1 ↑ to REQ ↓ Delay		1000		1000	ns	
7	TdACK(REQ)	ACKIN ↓ to REQ ↓ Delay		1000		1000	ns	
8	TdSU(DMA)	Data Setup Time to DMASTB	200		150		ns	
9	TdH(DMA)	Data Hold Time to DMASTB	30		20		ns	
10	TdDMA(DR)	DMASTB ↓ Data Valid		150		100	ns	
11	TdDMA(DRH)	DMASTB ↑ to Data Not Valid	0		0		ns	
12	TdDMA(DR2)	DMASTB ↑ to Data Bus Float		70		45	ns	

Note: 1. The Delay is from DAV ↓ for 3-Wire Input Handshake. The delay is from DAC ↑ for 3-Wire Output Handshake.

SWITCHING WAVEFORMS



WF004260

Figure 37.

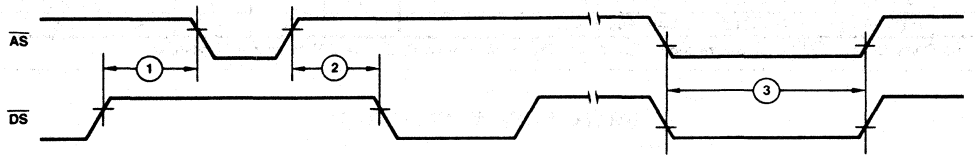
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Z-BUS RESET TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
1	TdDSQ(AS)	Delay from $\overline{DS}$ ↑ to $\overline{AS}$ ↓ for No. Reset	40		20		ns	
2	TdASQ(DS)	Delay for $\overline{AS}$ ↑ to $\overline{DS}$ ↓ for No. Reset	50		30		ns	
3	Tw(AS + DS)	Minimum Width of $\overline{AS}$ and $\overline{DS}$ both Low for Reset	500		350		ns	1

Note: 1. Internal circuitry allows for the reset provided by the Z8 ($\overline{DS}$ held Low while $\overline{AS}$ pulses) to be sufficient.

SWITCHING WAVEFORMS



WF007120

Figure 38.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

NON-Z-BUS CPU INTERFACE TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
1	TsA(RD)	Address Setup to $\overline{RD}$ ↓	80		80		ns	1
2	TsA(WR)	Address Setup to $\overline{WR}$ ↓	80		80		ns	
3	ThA(RD)	Address Hold Time to $\overline{RD}$ ↑	0		0		ns	1
4	ThA(WR)	Address Hold Time to $\overline{WR}$ ↑	0		0		ns	
5	TsCEl(RD)	$\overline{CE}$ Low Setup Time to $\overline{RD}$	0		0		ns	1
6	TsCEl(WR)	$\overline{CE}$ Low Setup Time to $\overline{WR}$	0		0		ns	
7	ThCEl(RD)	$\overline{CE}$ Low Hold Time to $\overline{RD}$	0		0		ns	1
8	ThCEl(WR)	$\overline{CE}$ Low Hold Time to $\overline{WR}$	0		0		ns	
9	TsCEh(RD)	$\overline{CE}$ High Setup Time to $\overline{RD}$	100		70		ns	1
10	TsCEh(WR)	$\overline{CE}$ High Setup Time to $\overline{WR}$	100		70		ns	
11	TwRD1	$\overline{RD}$ Low Width	390		250		ns	
12	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		ns	
13	TdRDf(DR)	$\overline{RD}$ ↓ to Valid Data Delay		250		180	ns	
14	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		ns	
15	TdRD(DRz)	$\overline{RD}$ ↑ to Data Bus Float		70		45	ns	2
16	TwWR1	$\overline{WR}$ Low Width	390		250		ns	
17	TsDW(WR)	Data Setup Time to $\overline{WR}$	0		0		ns	
18	ThDW(WR)	Data Hold Time to $\overline{WR}$	30		20		ns	
19	Trc	Valid Access Recovery Time	1000		650		ns	3

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is measured to the time the output has changed 0.5V from steady state with minimum AC load and maximum DC load.
3. This is the delay from $\overline{RD}$ ↑ or $\overline{WR}$ ↑ of one FIO access to $\overline{RD}$ ↓ or $\overline{WR}$ ↓ of another FIO access.

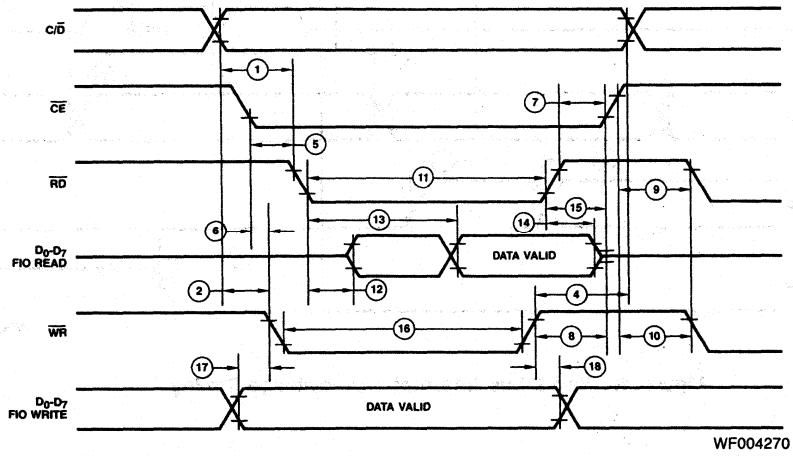


Figure 39a. Non-Z-Bus CPU Interface Timing

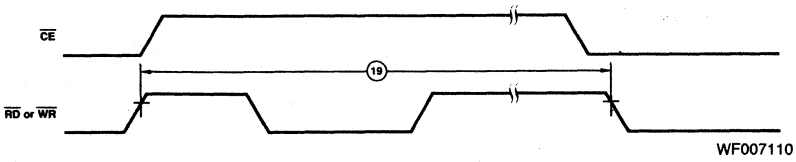


Figure 39b. Non-Z-Bus Interface Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

NON-Z-BUS INTERRUPT ACKNOWLEDGE TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
20	TdIEI(IEO)	IEI to IEO Delay		150		100	ns	4
21	TdI(IEO)	$\overline{INTACK}$ ↓ to IEO ↓ Delay		350		250	ns	4
22	TsIEI(RDA)	IEI Setup Time to $\overline{RD}$ (Acknowledge)	100		70		ns	4
23	TdRD(DR)	$\overline{RD}$ ↓ to Vector Valid Daly		250		180	ns	
24	TwRD1(IA)	Read Low Width (Interrupt Acknowledge)	390		250		ns	
25	ThIA(RD)	$\overline{INTACK}$ ↑ to $\overline{RD}$ ↑ Hold Time	30		20		ns	
26	ThIEI(RD)	IEI Hold Time to $\overline{RD}$ ↑	20		10		ns	
27	TdRD(INT)	$\overline{RD}$ ↑ to $\overline{INT}$ ↑ Delay		800		800	ns	
28	TdDCST	Interrupt Daisy Chain Settle Time		350		250	ns	4

Notes: 4. The parameter for the devices in any particular daisy chain must meet the following constraints: the delay from $\overline{INTACK}$↓ to $\overline{RD}$↓ must be greater than the sum of TdINA(IEO) for the highest priority peripheral, TsIEI(RD) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.

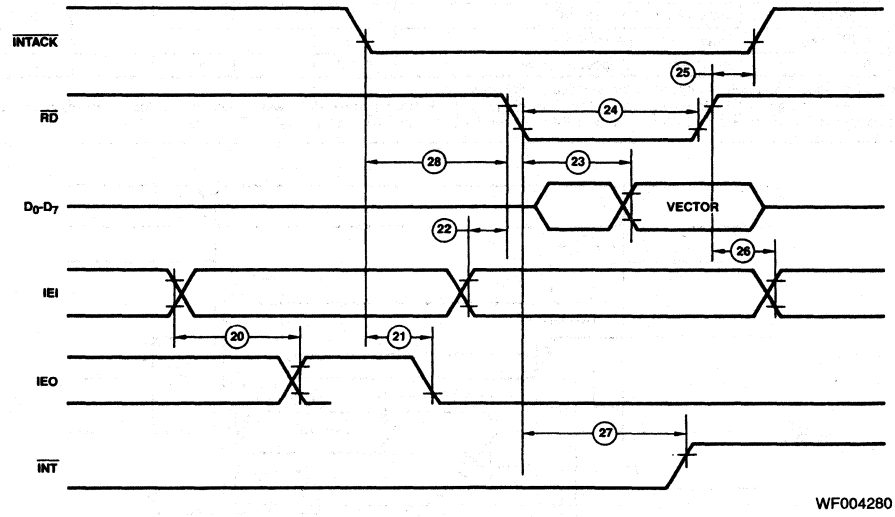


Figure 40. Non-Z-Bus Interrupt Acknowledge Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

NON-Z-BUS REQUEST/WAIT TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
1	TdRD(WT)	$\overline{CE}$ ↓ to $\overline{WAIT}$ Active		200		170	ns	
2	TdRD1(WT)	$\overline{RD1}$ ↓ to $\overline{WAIT}$ Inactive		1000		1000	ns	
3	TdACK(WT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ Inactive		1000		1000	ns	1
4	TdRD(REQ)	$\overline{RD}$ ↓ to $\overline{REQ}$ Inactive		350		300	ns	
5	TdRD1(REQ)	$\overline{RD1}$ ↑ to $\overline{REQ}$ Active		1000		1000	ns	
6	TdACK(REQ)	$\overline{ACKIN}$ ↓ to $\overline{REQ}$ Active		1000		1000	ns	
7	TdDAC(RD)	$\overline{DACK}$ ↓ to $\overline{RD}$ ↓ or $\overline{WR}$ ↓	100		80		ns	
8	TSU(WR)	Data Setup Time to $\overline{WR}$	200				ns	
9	Th(WR)	Data Hold Time to $\overline{WR}$	30		20		ns	
10	TdDMA	$\overline{RD}$ ↓ to Valid Data		150		100	ns	2
11	TdDMA(DRH)	$\overline{RD}$ ↑ to Data Not Valid	0		0		ns	2
12	TdDMA(DRZ)	$\overline{RD}$ ↑ to Data Bus Float		70		45	ns	2

Notes: 1. The delay is from $\overline{DAV}$↓ for 3-Wire Input Handshake. The delay is from DAC↑ for 3-Wire Input Handshake.
2. Only when $\overline{DACK}$ is active.

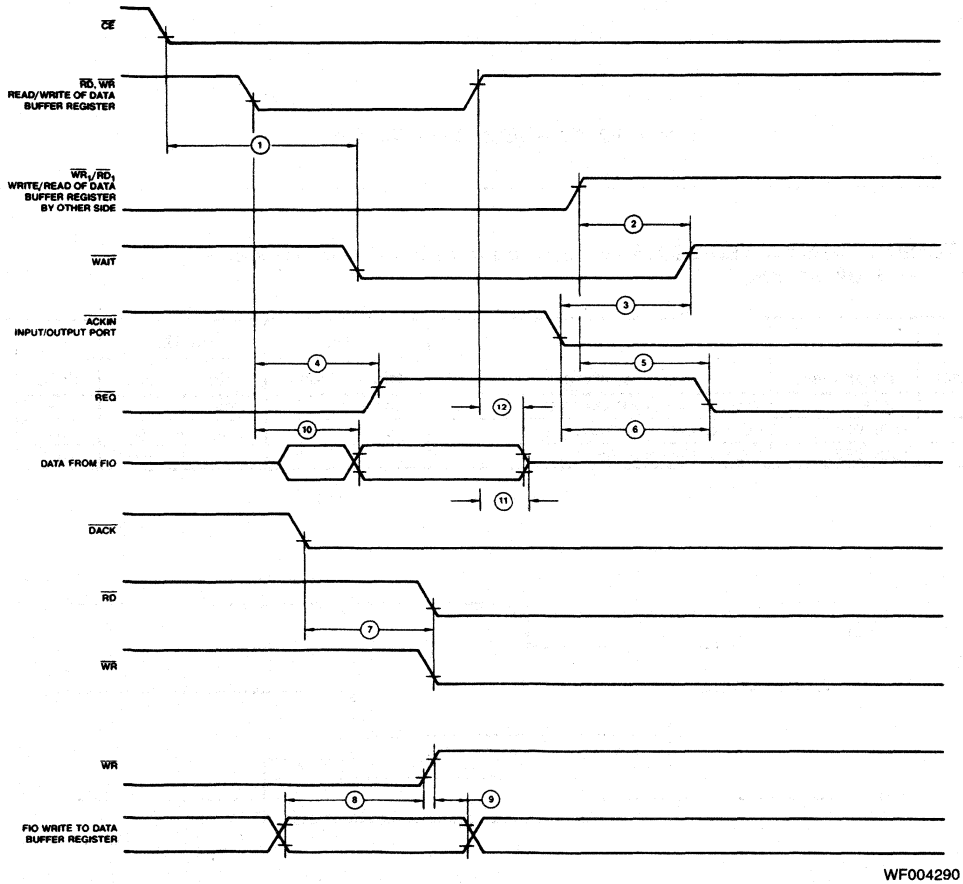


Figure 41. Non-Z-Bus Request/Wait Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
NON-Z-BUS RESET TIMING

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
1	TdWR(RD)	Delay from $\overline{WR}$ ↑ to $\overline{RD}$ ↓	100		70		ns	
2	TdRD(WR)	Delay from $\overline{RD}$ ↑ to $\overline{WR}$ ↓	100		70		ns	
3	TwRD + WR	Width of $\overline{RD}$ and $\overline{WR}$, both Low for Reset	500		350		ns	

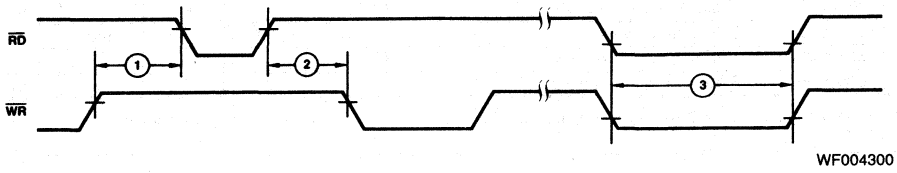


Figure 42. Non-Z-Bus Reset Timing

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
PORT 2 SIDE OPERATION

Number	Parameters	Description	4MHz		6MHz		Units	Notes
			Min	Max	Min	Max		
1	TwCLR	Width of Clear to Reset FIFO	700		700		ns	
2	TdOE(DO)	$\overline{OE}$ ↓ to Data Bus Driven	0	150	0	150	ns	
3	TdOE(DRZ)	$\overline{OE}$ ↑ to Data Bus Float		100		100	ns	

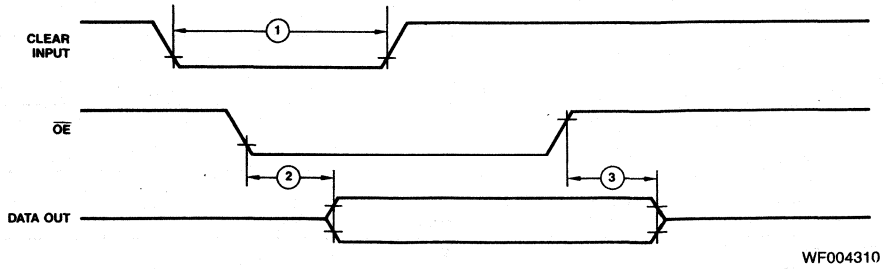


Figure 43. Port 2 Side Operation

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

FIO 2-WIRE HANDSHAKE TIMING

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TsDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ to Setup Time	50		50		ns
2	TdACKf(RFD)	$\overline{ACKIN}$ ↓ to RFD↓ Delay	0	500	0	500	ns
3	TdRFDr(ACK)	RFD ↑ to $\overline{ACKIN}$ ↓ Delay	0		0		ns
4	TsDO(DAV)	Data Out to $\overline{DAV}$ ↓ Setup Time	25		25		ns
5	TdDAVf(ACK)	$\overline{DAV}$ ↓ to $\overline{ACKIN}$ ↓ Delay	0		0		ns
6	ThDO(ACK)	Data Out to $\overline{ACKIN}$ Hold Time	50		50		ns
7	TdACK(DAV)	$\overline{ACKIN}$ ↓ to $\overline{DAV}$ ↑ Delay	0	500	0	500	ns
8	ThDI(RFD)	Data Input to RFD ↓ Hold Time	0		0		ns
9	TdRFDf(ACK)	RFD ↓ to $\overline{ACKIN}$ ↑ Delay	0		0		ns
10	TdACKr(RFD)	$\overline{ACKIN}$ ↑ ($\overline{DAV}$ ↑) to RFD ↑ Delay – Interlocked and 3-Wire Handshake	0	400	0	400	ns
11	TdDAVr(ACK)	$\overline{DAV}$ ↑ to $\overline{ACKIN}$ ↑ (RFD ↑)	0		0		ns
12	TdACKr(DAV)	$\overline{ACKIN}$ ↑ to $\overline{DAV}$ ↓	0	800	0	800	ns
13	TdACK(Empty)	$\overline{ACKIN}$ to Empty		600		600	ns
14	TdACK (Full)	$\overline{ACKIN}$ to Full		600		600	ns
15	$\overline{ACKIN}$ Clock Rate		1.0				µs

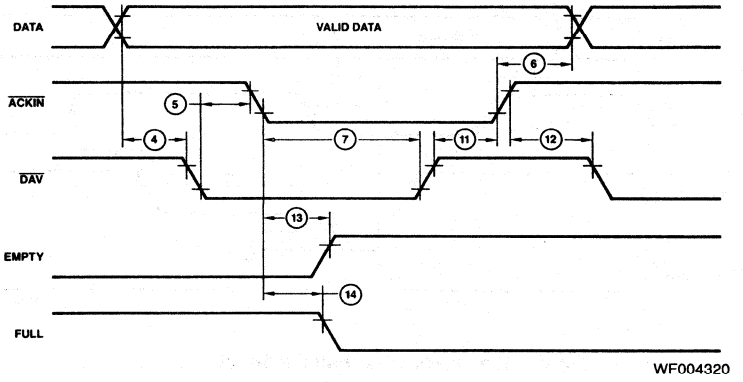


Figure 44a. 2-Wire Handshake (Port 2 Side Only) Output

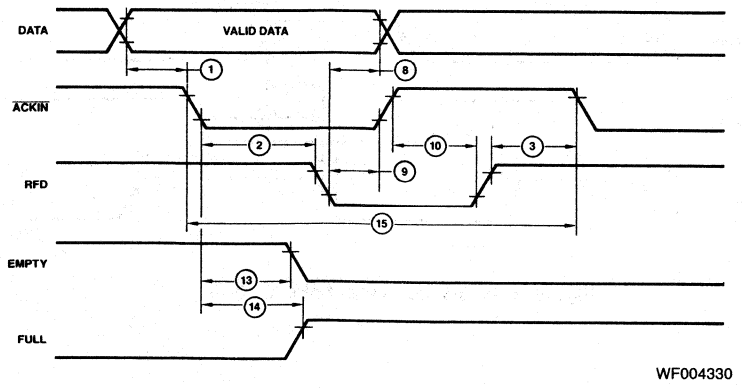


Figure 44b. 2-Wire Handshake (Port 2 Side Only) Input

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

3-WIRE HANDSHAKE

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TsDI(DAV)	Data Input to $\overline{DAV}$ ↓ Setup Time	50		50		ns
2	TdDAVIf(RFD)	$\overline{DAV}$ ↓ to RFD ↓ Delay	0	500	0	500	ns
3	TdDAVf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay	0	500	0	500	ns
4	ThDI(DAC)	Data In to DAC ↑ Hold Time	0		0		ns
5	TdDAClr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay	0		0		ns
6	TdDAVIr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay	0	500	0	500	ns
7	TdDAVIr(RFD)	$\overline{DAV}$ ↑ to RFD ↑ Delay	0	500	0	500	ns
8	TdRFDI(DAV)	RFD ↑ to $\overline{DAV}$ ↓ Delay	0		0		ns
9	TsDO(DAC)	Data Out to $\overline{DAV}$ ↓					ns
10	TdDAVOf(RFD)	$\overline{DAV}$ ↓ to RFD ↓ Delay	0		0		ns
11	TdDAVOf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay	0		0		ns
12	ThDO(DAC)	Data Out to DAC ↑ Hold Time					ns
13	TdDACOr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay		400		400	ns
14	TdDAVOr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay	0		0		ns
15	TdDAVOR(RFD)	$\overline{DAV}$ ↑ to RFD ↑ Delay	0		0		ns
16	TdRFDO(DAV)	RFD ↑ to $\overline{DAV}$ ↓ Delay	0	800	0	800	ns

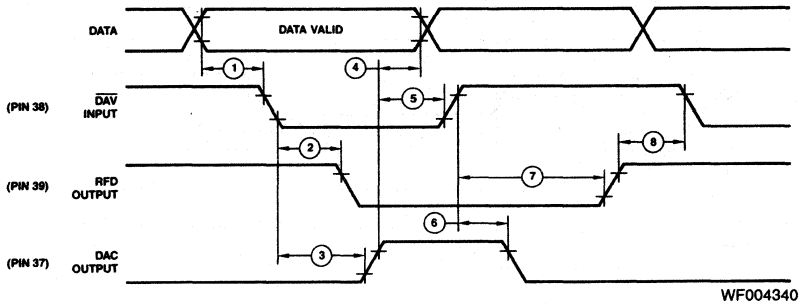


Figure 45a. 3-Wire Handshake Input

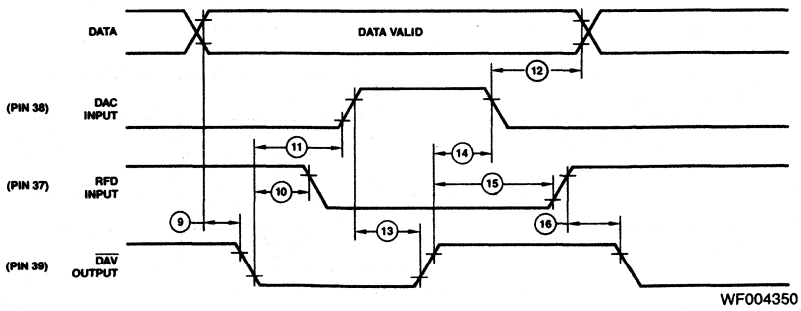


Figure 45b. 3-Wire Handshake Output

Am8052/8152A/8153A

Alphanumeric CRT Controller Chip-Set

DISTINCTIVE CHARACTERISTICS

- 100MHz video dot rate supports high resolution CRT monitors with 132/60 or 96/66 screen formats
- Background or window soft-scroll capability without external MSI or software overhead
- User-friendly CPU interface. Compatible with 8086, Z8000 and 68000 CPUs.
- On-chip line buffers support flicker free soft-scrolling
- Supports proportional character widths
- Automatic concatenation of up to three trailing blank pixels supports text justification
- Flexible attribute handling
- Color and bit-mapped graphics extension

GENERAL DESCRIPTION

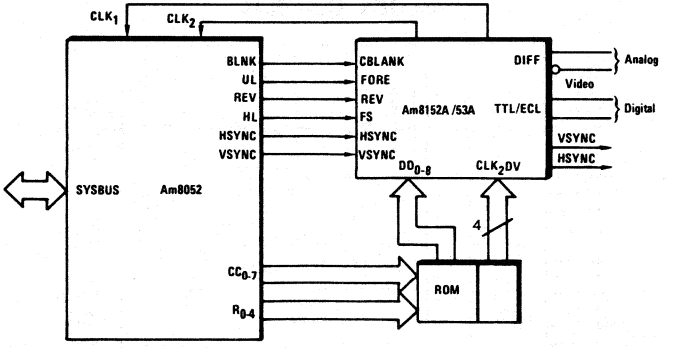
State-of-the-art CRT terminals incorporate advanced user programmable features, such as flexible attribute handling, proportional spacing of characters, split screens or multiple window display, soft-scrolling of windows, and variable character width and height in full page 132 x 60 screen formats. The video subsystem of a CRT terminal with these sophisticated video features can now be implemented with as few as three device packages with the Am8052/8152A chip-set, significantly reducing IC cost and board space without sacrificing performance. The Am8052/8152A chip-set consists of an NMOS LSI CRT Controller (CRTC), the Am8052, and the bipolar IMOX™ Video System Controller (VSC), the Am8152A.

The Am8052 has on-chip DMA which operates via linked list data structures to simplify text editing. In addition, it is the only known CRT controller with three line buffers on chip to support flicker free soft-scrolling of background or windows. The Am8052 has on-chip logic that can support a number of attributes, such as highlight, reverse video, underline, etc., which enhance display presentation (see Table 1 in the Am8052 CRTC datasheet). In addition, four user definable attributes are available providing user flexibility.

The Am8052 CRT controller performs all the data processing prior to video serialization. This latter task is performed by the Am8152A, for dot rates up to 60MHz, or by the Am8153A for dot rates up to 100MHz. Apart from the video output lines (TTL on the Am8152A and ECL on the Am8153A) these two devices are functionally identical. The high speed of the video controller, a result of AMD's patented IMOX™ technology, supports high resolution screens of 500-1000 scan lines per frame or rows with 100 or more characters. In addition, the Am8152A has on-chip logic that supports proportional spacing of characters and allows for text justification.

The CRT chip-set is designed for easy interface with all popular CPUs such as the 8086, 68000 and Z8000,* thus permitting design flexiblity with minimal logic.

TYPICAL APPLICATION WITH PROPORTIONAL SPACING



AF002321

IMOX is a registered trademark of Advanced Micro Devices, Inc.
*Z8000 is a trademark of Zilog, Inc.

03684A

Am8052 CRTC

Alphanumeric CRT Controller

DISTINCTIVE CHARACTERISTICS

- On-chip DMA capability, operating via linked-list data structures
- Three on-chip row buffers, each 132 characters by 20 bits support split-screen smooth-scrolling
- General purpose microprocessor interface. Compatible with 8086, Z8000* and 68000 CPUs.
- Smooth-scrolling capability, with minimal CPU overhead
- Multiple vertical and horizontal screen divisions, with optional smooth-scrolling within a window
- Character attributes (12 bits) can be invoked on a character by character basis
- Flexible vertical and horizontal sync control
- Flexible blanking for control of front and back porch positions
- Non-interlace, repeat field interlace, video interlace options
- High resolution five-bit character generator row addressing
- 16M byte system memory addressing capability
- Programmable blink options for cursors and characters

GENERAL DESCRIPTION

The Am8052 CRT Controller is a general purpose interface device for raster scan CRT displays. The CRTC provides efficient manipulation of complex character formats and screen structures to allow sophisticated text display without undue CPU overhead.

The CRTC is a register-oriented product that is fully user programmable. The timing definition and operating modes are initialized by the host CPU. Display formats are realtime programmable on a row-by-row basis. Character attributes are specified on a character or field basis, and are interpreted and acted upon during active display of a character row.

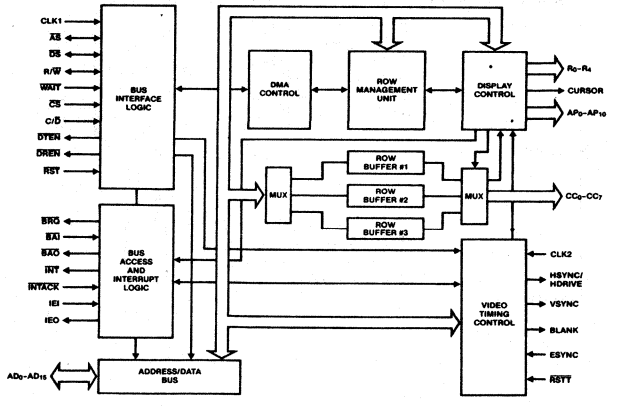
Internal DMA capability assures efficient transfer of display information to the three on-chip line buffers. These three line buffers prevent screen flashing in split-screen smooth-scrolling operations. The DMA loads the line buffers via linked list data blocks which facilitate easier editing and text composition.

The Am8052, in conjunction with the Am8152A bipolar video system controller, allows for the flexible assignment of visual attributes. The twelve attribute bits stored in the Am8052 include superscript, subscript, blink, highlight, reverse, underline, strike through and cursor. Both character and cursor can be made to blink at three different rates, and the blink duty cycle is programmable. Further flexibility is achieved by the Am8152A, which allows the video stream to be manipulated by selection of background and foreground as well as background/foreground reversal.

The Am8052 and Am8152A combination also supports proportional spacing, text justification and double width characters.

The Am8052 CRTC is assembled in a 68-pin package, while the bipolar Am8152A VSC is assembled in a 48-pin DIP. These interface circuits are available as a chip-set for high performance CRT applications.

BLOCK DIAGRAM



BD003360

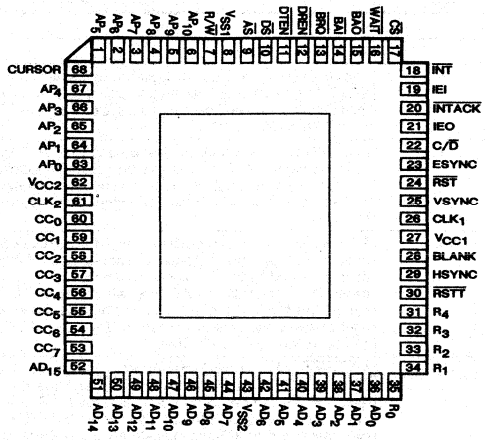
Figure 1.

Z8000 is a trademark of Zilog, Inc.

03684C

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

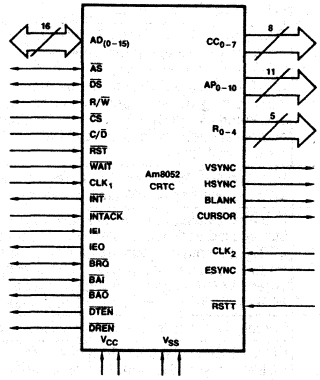


CD005191

Figure 2.

Also available in 68 PLCC. Package pin numbers correspond sequentially to device pin numbers. The Am8052 CRTC is molded upside down and rotated ninety degrees counterclockwise to match the LCC pinout.

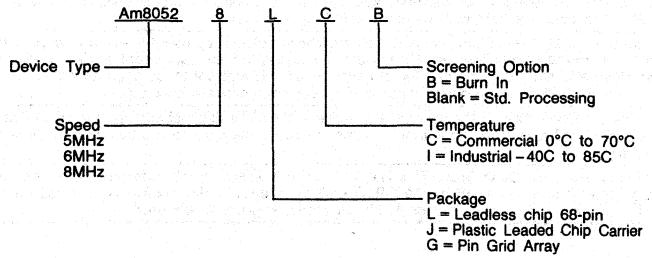
LOGIC SYMBOL



LS001211

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am8052	5LC 6LC 8LC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
8, 43	V_{SS1}, V_{SS2}		Ground
27, 62	V_{CC1}, V_{CC2}		+5V Power Supply
26	CLK_1	I	Timing Clock. The Clock 1 signal controls and times the DMA and peripheral portion of the CRTC. In proportional spacing applications, where CLK_2 is variable, CLK_1 must be used to time the horizontal and vertical sync rates. CLK_1 is non-TTL compatible, and is normally driven by the Am8152A/8153A VSC.
61	CLK_2	I	Display Clock. The Clock 2 signal is used to time character accesses from the CRTC line buffers. In applications which do not use proportional spacing, CLK_2 is fixed in frequency and can be used to time horizontal and vertical sync rates, allowing CLK_1, the system clock, to be unrelated and asynchronous to the display timing. CLK_2 is non-TTL compatible and should be driven by the VSC.
36-42, 44-52	AD_0-AD_{15}	I/O	Address/Data Bus, Three-State. The Address/Data Bus is a multiplexed, bidirectional, high-true, three-state bus. The presence of addresses is defined by the $\overline{AS}$ signal, and the presence of data is defined by the $\overline{DS}$ signal. When the CRTC is in control of the system via its internal DMA capability, it controls the AD Bus; when the CRTC is idle, the CPU or other external devices control the AD Bus and may use it to access the internal registers of the CRTC. The high-order 8-bit memory address is output on the $AD_0 - AD_7$ lines. Interrupt Vector information is also output in the $AD_0 - AD_7$ lines.
9	$\overline{AS}$	I/O	Address Strobe, Three-State. Address strobe is a bidirectional, active-LOW, three-state signal. When the CRTC is in the slave mode and the bus master is accessing the CRTC's internal registers, $\overline{AS}$ can be used to optionally latch $\overline{CS}$ and C/$\overline{D}$ information during the first part of the transaction. During a DMA operation when the CRTC is in control of the system, $\overline{AS}$ is an output generated by the CRTC to indicate a valid address on the bus. In the slave mode, the $\overline{AS}$ signal may be asynchronous to CLK_1.
10	$\overline{DS}$	I/O	Data Strobe. Three-State. Data Strobe is a bidirectional, active-LOW, three-state signal. When the CRTC is in the slave mode and the external system is transferring information to or from it, $\overline{DS}$ is a timing input used by the CRTC to move data to or from the AD bus. In the slave mode, the $\overline{DS}$ signal may be asynchronous to CLK_1. During a DMA operation when the CRTC is in control of the system, $\overline{DS}$ is an output generated by the CRTC and used by the system to move data onto the AD bus.
17	$\overline{CS}$	I	Chip Select. The $\overline{CS}$ input is an active LOW signal used by the host processor to select the CRTC for a slave transfer.
16	$\overline{WAIT}$	I	Wait. The $\overline{WAIT}$ input is an active LOW signal used to stretch the $\overline{DS}$ strobe whenever the CRTC has access to the host's bus for data transfer. The status of the $\overline{WAIT}$ signal is sampled on the falling edge CLK_1 during T_2 or T_W.
7	R/$\overline{W}$	I/O	Read/$\overline{Write}$, Three-State. Read/$\overline{Write}$ is a bidirectional, three-state signal indicating the data direction for the bus transaction under way, and remains stable for the length of the bus cycle. When $\overline{CS}$ input is active, Read (HIGH) indicates that the system is requesting data from the CRTC and Write (LOW) indicates that the system is presenting data to the CRTC. On the other hand, during a DMA operation when the CRTC is in control of the system, R/$\overline{W}$ is an output generated by the CRTC, with Read indicating that data is being requested by the CRTC from the addressed memory location and Write indicating that the CRTC is driving a high-order address to an external latch.
13	$\overline{BRQ}$	I/O	Bus Request. When the CRTC requires use of the bus for DMA activity, the $\overline{BRQ}$ line is driven LOW. It remains LOW until it has ceased using the bus.
14	$\overline{BAI}$	I	Bus Acknowledge In. Bus acknowledge In is an active LOW input. When the CRTC requires host bus access and has successfully pulled its $\overline{BRQ}$ pin LOW, a $\overline{BAI}$ LOW input signifies that the CRTC has obtained bus mastership after having internally synchronized its $\overline{BAI}$ active LOW input for two clock periods of CLK_1. The synchronization is required to alleviate metastable problems. When the CRTC does not require host bus access, the $\overline{BAI}$ input ripples to the $\overline{BAO}$. Forcing $\overline{BAI}$ HIGH will cause the Am8052 to relinquish the bus.
68	CURSOR	O	Cursor. This pin is the cursor output indicator.
23	ESYNC	I	External Sync. This pin is the external synchronization input line. If the ES bit in the mode register is set, the vertical frame scan will commence after the rising edge of ESYNC.
29	HSYNC	O	Horizontal Sync. HYSNC is an active HIGH output used to cause horizontal retrace of the CRT's electron beam. The output is held active LOW while the CRTC is reset to prevent unknown synchronization to the CRT which may cause damage to high bandwidth tubes. Note that this pin can also be initialized as Horizontal Drive.
25	VSYNC	O	Vertical Sync. VSYNC is an active HIGH output used to cause vertical retrace of the CRT's electron beam. VSYNC can be optionally synchronized by the ESYNC input. VSYNC is held LOW while the CRTC is reset to prevent damage to the CRT.
28	BLANK	O	Blank Video. BLANK is an active HIGH output. It serves to blank out inactive display areas of the CRT. The output is held active while the CRTC is reset.
35-31	$R_0 - R_4$	O	Row Control. $R_0 - R_4$ outputs are active HIGH. These outputs represent the binary count of the active scan line being displayed. These outputs address the least signficant address portion of an external character generator. The outputs are all held high for those scan lines that do not carry active video during normal character or superscript/subscript display.
60-53	$CC_0 - CC_7$	O	Character Code. $CC_0 - CC_7$ outputs are active HIGH. The 8-bit character port, $CC_0 - CC_7$, outputs eight bits of data stored in the character code section of the line buffer currently being displayed.
18	$\overline{INT}$	O	Interrupt Request, Open Drain. This line is used to indicate an interrupt request to the host processor. It is driven LOW by the CRTC until an interrupt acknowledge is received on the $\overline{INTACK}$ pin or the relevant IP or IE bits in Mode Register 2 are reset.

03684C

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
20	$\overline{INTACK}$	I	Interrupt Acknowledge. When $\overline{INTACK}$ is driven LOW, the CRTC examines its IEI line to determine whether it has been granted an acknowledge by the CPU. It also starts priority resolution of the Daisy Chain. When $\overline{DS}$ is active, the vector is placed on the bus if enabled.
19	IEI	I	Interrupt Enable-In. A HIGH on IEI during an Interrupt Acknowledge cycle is regarded as an interrupt acknowledge to the CRTC. A LOW on IEI during Interrupt Acknowledge signifies that a higher priority interrupt on the dasiy chain is being acknowledged.
21	IEO	O	Interrupt Enable-Out. IEO follows IEI during Interrupt Acknowledge if the CRTC has not made an interrupt request. IEO LOW disables lower priority devices from making interrrupt requests.
11, 12	$\overline{DTEN}$, $\overline{DREN}$	O	$\overline{Data\ Transmit\ Enable}$, $\overline{Data\ Receive\ Enable}$, Open Drain. Data Transmit Enable and Data Receive Enable are used to control bus transceiver external to the CRTC should they be required. When $\overline{DTEN}$ is LOW, the transceiver should transmit from the CRTC onto the bus. When $\overline{DREN}$ is LOW, the transceiver should receive data from the bus. $\overline{DTEN}$ and $\overline{DREN}$ are never LOW simultaneously.
22	C/$\overline{D}$	I	Command/$\overline{Data}$,. C/$\overline{D}$ is used by the CRTC when in the slave mode to determine if an I/O transaction with the host CPU is transferring a command or data. When the CRTC is not involved in an I/O transaction with the host, C/$\overline{D}$ is disregarded.
63-67, 1-6	$AP_0 - AP_{10}$	O	Attribute Port. These 11 lines are used to display character attribute information synchronous with each character and CLK_2. During horizontal SYNC, the row attribute information contained in the Row Redefinition Block is output on $AP_0 - AP_{10}$.
15	$\overline{BAO}$	O	$\overline{Bus\ Acknowledge\ Out}$. $\overline{BAO}$ output is forced active HIGH when the CRTC requests bus mastership; otherwise, the $\overline{BAI}$ input ripples out of the CRTC via the $\overline{BAO}$ output.
24	$\overline{RST}$	I	$\overline{Reset}$. A LOW on this input for at least 5 clock cycles is interpreted as a reset signal. The effect of reset is to drive all CRTC bus signals into the high-impedance state, to clear all mode bits except bits 9 through 15 in MR2, and to force the CRTC into the slave mode.
30	$\overline{RSTT}$	I	$\overline{Test\ Reset}$. For test use only. This pin is a No Connect.

TABLE 1.

Attribute	Effect
Reverse	– Causes the designated character to be displayed in reverse video.
Highlight	– Highlights the applicable character.
Blink	– Blinks the designated character at one of four programmed blink rates.
Underline	– Underlines the designated character at a programmable scan line.
Subscript	– Causes the character to be displayed as a subscript.
Superscript	– Causes the character to be displayed as a superscript.
Shifted Underline	– A second underline.
Cursor	– Causes the attribute or X-Y cursor to be displayed at the designated character position.
Latched	– Indicates that the attribute should be latched for all successive characters until changed.
Ignore	– Causes the CRTC to skip over the designated characters. Useful for embedded control characters and protected fields that do not get displayed.
User Definable	– Four attribute bits reserved for user definition.

PRODUCT OVERVIEW

The block diagram of the Am8052 CRTC is shown in Figure 1. Communication with the external host system takes place over the 16-bit Address/Data bus, $\overline{AD}_0$–$\overline{AD}_{15}$. Transfers over the AD bus are controlled by the $\overline{CS}$, C/$\overline{D}$, $\overline{AS}$, $\overline{DS}$, and R/$\overline{W}$ lines. When the CRTC is in the slave mode, these four bus control lines are inputs. When the CRTC is in the DMA mode, $\overline{AS}$, R/$\overline{W}$ and $\overline{DS}$ are outputs and control the external bus.

Following reset, the host system initializes the CRTC's timing and control registers, as well as one address pointer to the start of the display data location in the host memory. Following initialization and upon command from the host, the CRTC takes over bus control from the host and transfers display row control data, character code, and character attribute data. The CRTC requests the host bus by sampling the $\overline{BRQ}$ line for activity; if the $\overline{BRQ}$ line is HIGH, the CRTC drives it LOW, and also drives $\overline{BAO}$ HIGH, to obtain priority over lower priority bus requestors. The on-chip DMA Controller circuit controls the data transfer and performs character data loading into the on-board line buffers.

The CRTC is real-time programmable on a character row-by-row basis through a row control data block fetched either from the host memory or from a dedicated display memory. The row control block contains address links to the next row's row control block, a character and attribute data address for the current row and other pertinent control functions for the row. Data from the row control block is transferred into the appropriate set of registers for active control of display and data fetch operations during the subsequent display of character row data. A Top Of Page register contains the address of the Main Definition Block for the screen. The Main Definition Block, in turn, points to the first Row Control Block. The character row data, comprised of character code and attribute (if the latter is specified), is fetched starting at the address and for the character length obtained from the Row Control Block. The character code and its attribute consist of a 20-bit wide word which is stored, FIFO style, into one of the three on-board 132-character by 20-bit line buffers. Character attributes are on a character-by-character basis and are interpreted and acted upon by the CRTC during the active display period of the contents of a line buffer. Output lines CC_0–CC_7 form the transfer path for character code data to an external matrix type character generator, while the character attribute, after selective masking, is interpreted and combined with the resulting video.

Output lines R_0–R_4 exhibit the scan line number for the specific character being displayed, while the character Row Control logic allows alteration of the scan line number output at the R_0–R_4 lines to enable the display of normal superscript or subscript characters.

The HSYNC, VSYNC and BLANK output lines provide the CRT synchronization signals. The Horizontal and Vertical Control logic blocks contain counters and host programmable registers for deriving the timing signals from either the CLK_1 or the CLK_2 input as well as an ESYNC input line for frame synchronization to an external source, such as the power line frequency. CLK_2 runs at the display character rate. It is a submultiple of the dot clock, whose frequency is determined by the Am8152A oscillator. CLK_2 controls the CRT synchronization lines HSYNC and VSYNC, as well as BLANK, and the rate of character output from the CRTC. CLK_1, which may be asynchronous to CLK_2, controls all DMA and related bus activity, associated with the CRTC. In proportional spacing applications, CLK_1 may be also used to time the synchronization signals.

CHARACTER ATTRIBUTES

Character attributes affect various CRTC output signals and other operations on a character-by-character basis. Each attribute word occupies a 16-bit word in memory. Each character, however, need not invoke a new attribute.

Character attributes are stored in parallel with the corresponding character code in each line buffer.

The character attribute information which makes up the character attribute word is shown below:

AW_{15}	Latched/Unlatched	AW_7	User definable
AW_{14}	Cursor	AW_6	Highlight
AW_{13}	Ignore	AW_5	Reverse
AW_{12}	Reserved	AW_4	Superscript
AW_{11}	Reserved	AW_3	Subscript
AW_{10}	User definable	AW_2	Shifted underline/ Strike through
AW_9	User definable	AW_1	Underline
AW_8	User definable	AW_0	Blink

DESCRIPTION OF CHARACTER ATTRIBUTES

LATCHED/$\overline{\text{UNLATCHED}}$

When this bit is set to 1 ("latched"), the attribute information applies to all characters following the character that invoked the attribute word. Only the presence of a further latched attribute word cancels the effect of a previous latched attribute word. If the Latched/$\overline{\text{Unlatched}}$ bit is set to 0 ("unlatched"), then the attribute information only applies to the character that invoked the attribute word. All successive characters are modified by the latched attribute information that was valid prior to the unlatched attribute word. The Latched/$\overline{\text{Unlatched}}$ bit is not output to the attribute port. The initial state of the latched attribute value is undefined. At the start of any horizontal line, the latched attribute information is the same as at the end of the previous line, unless changed by a further latched attribute.

CURSOR

If this bit is set, then a cursor is displayed at the affected character position(s), dependent upon the mode of the cursor display logic. See the section on cursor display for further details.

IGNORE

When the ignore is set, it inhibits the loading of the associated character into the CRTC line buffer. Such character(s) may be used as control character or software tags, and are not displayed. Whenever the ignore encoding is detected, both the attribute word and its associated character code are not written into the line buffer, unless the DH (Display Hidden) bit in Mode Register 1 is set. Note that the ignore bit is not brought out to the attribute port.

USER DEFINABLE

The AW_7–AW_{10} attribute bits provide 4 bits of user definable attribute information. These bits are directly output on pins AP_7–AP_{10} of the attribute port. (In addition to these four user definable attribute bits, the cursor bit can also be user definable under certain conditions.)

HIGHLIGHT

When this bit is set and AP_6 is connected to the FS input of the Am8152A, the character is displayed highlighted. The AP_6 pin of the attribute port goes active for each scan line of the relevant character(s).

REVERSE

When this bit is set and AP_5 is connected to the REV input of the Am8152A, the character is displayed reversed. The AP_5 pin of the attribute port goes active for each scan line of the relevant character(s).

SUPERSCRIPT

When this bit is set to 1, the affected character is displayed as a superscript. Its position on the character row ($R_0 - R_4$) is determined by the superscript control field in the row redefinition block for that particular row.

SUBSCRIPT

When this bit is set, the affected character is displayed as a subscript. Its position on the character row ($R_0 - R_4$) is determined by the subscript control fields in the row redefinition block.

UNDERLINE/SHIFTED UNDERLINE

Attribute bits AW_1 and AW_2 provide underline and shifted underline display. The underline/shifted underline display information is output on the AP_1 and AP_2 attribute port pins, during applicable scan lines of the character. (The applicable scan lines have been programmed within the row redefinition blocks.)

BLINK

When this attribute is invoked, the attribute port pin AP_0 is gated with the character blink rate generator, during the time that the relevant character is output on $CC_0 - CC_7$.

The character blink rate and character blink duty cycle are derived from the blink field of the Main Definition block.

ATTRIBUTE FETCHES

Attributes can be fetched in three different ways to suit most design philosophies (see Figure 4). In Option 1, one attribute is fetched per character. This option, although straightforward, imposes heavy bus overhead since the DMA has to access the attribute list from memory for every character displayed on the screen. Bus overhead can be reduced considerably by fetching attributes on a demand basis. Options 2 and 3 accomplish this in two different ways. In Option 2 one character bit is set to 1 when an attribute is required. When this bit is set to 0, the attribute will not be fetched. This option allows 7 bits of character code or a 128 character set for display with no overhead for attribute incorporation.

Option 3 makes use of an 8-bit flag which precedes the character invoking the attribute. This option allows for a 255 character set with an 8-bit overhead (the flag) per attribute.

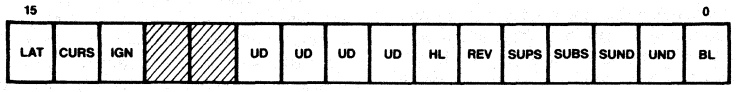


Figure 3. Am8052 Attribute Word

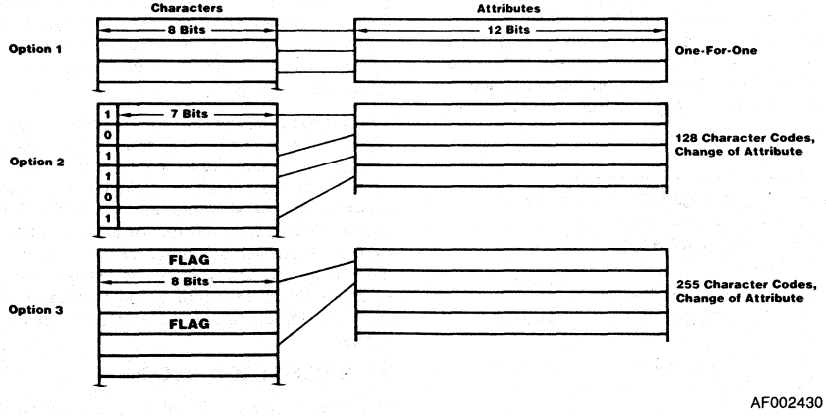


Figure 4. Attribute Fetch

CURSOR GENERATION

The CRTC can generate three different cursor formats: block, underline, and reverse, at variable blink rates and blink duty cycles.

Cursor information for the CRTC comes from two different sources, and each source can be independently steered to one of three different destinations. The two cursor sources are:

1. The XY cursor field which is held in the Main Definition Block for the screen.
2. Attribute Word bit 14 of the character attribute word. A cursor designated by an attribute will follow its row and character position whenever text is scrolled. The cursor controlled by positioning X and Y coordinates within the cursor X and Y register will be displayed on a fixed X, Y character position on the screen. The X, Y cursor should be disabled by resetting the CUE bit in Mode Register 2 during smooth scroll.

The steering of the cursor sources is under software control of the cursor mask field within mode register 2. The field is divided into two three-bit segments, one for the XY cursor and one for the attribute cursor. Three destinations are selectable for each cursor source:

(a) The cursor pin
(b) The underline pin
(c) The reverse video pin.

If (a) is selected, then either the whole character cell or partial character cell is selectable. If whole is selected, the cursor pin will be active for every scan line of the character cell. If part is selected, then the cursor pin will only be active for those scan lines within the limits of CURSOR START and CURSOR END, as specified in the row-redefinition block.

If (b) is selected, then either an underline will be active, if CURSOR START and END have the same values, or a block, if CURSOR START and END are not coincident.

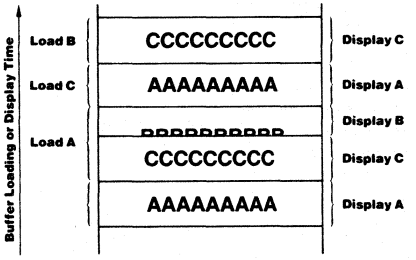
If (c) is selected, then either all or part of the character will be reversed, dependent upon the CURSOR START and CURSOR END setting as explained in (b).

In addition to these choices, either cursor can be made to blink (at the cursor blink rate) and duty cycle (as programmed into the main definition block blink field).

ROW BUFFERS

The on-chip DMA controller accesses the display memory and loads data from linked-list data blocks in memory into one of three row buffers. Each line buffer is 132 characters in length and 20 bits wide. Each 20-bit wide location accommodates an 8-bit character code and 12-bit attribute words. The row buffers operate in a rotating fill-display mode whereby one buffer is being loaded while another is being displayed.

The presence of three row buffers on-chip is of significant advantage in split screen smooth-scrolling operations where a character row may only be displayed for a single scan line. With two row buffers, this would not leave enough time for the reloading of the alternate line buffer. A partially filled buffer results in screen flashing. This can only be prevented by incorporating three line buffers. Figure 5 highlights this advantage.



AF002440

Figure 5. Triple Row Buffers

In the rotating fill-display mode, Row Buffer C is displayed when Row Buffer B is being loaded. Likewise the next Row Buffer C is loaded while Row Buffer A is being displayed. Because of the split-screen, Row Buffer B is displayed for one scan line only, while Row Buffer A is being loaded. By virtue of the third row buffer, the loading of Buffer A can spill over into the next buffer display, thus eliminating screen flashing.

SMOOTH-SCROLLING

A smooth-scroll is defined as the gradual displacement of a character row on a scan line-by-scan line basis. Smooth-scrolling is achieved by a gradual offsetting of the scan line counter, on a frame-by-frame basis. At the start of the scroll, the offset counter is set to zero or equal to the number of scan lines per character row, depending on whether the scroll is up or down. As the counter is incremented or decremented, the text travels up or down until the offset is equal to the number of scan lines or zero. The start of the screen pointer pointing to the character row is adjusted and the offset counter reset simultaneously to scroll the next successive character row. Smooth-scrolling of the entire screen is thus a simple task.

A number of applications require screen overlays, such as menu or status areas which must remain static while the major portion of the screen is scrolling or vice versa. The Am8052 can support multiple windows, each capable of being scrolled. (Only one window can be scrolled at a time.)

LINKED LIST DATA STRUCTURES

The DMA channel on the Am8052 operates via linked list structures that allow for the overlaying and independent smooth-scrolling of windows. The linked list data structures are particularly suited to the manipulation of data strings where insertions and deletions are common. A typical CRTC Linked List Structure is shown in Figure 6.

The linked list consists of Row Control Blocks (RCBs) for each character row on the screen. The RCB does not contain any displayable data, but contains the address which points to the character information. Each RCB is linked to the next block via an address link word (RCB ADR). The structure of the RCB linkage is shown in Figure 7. The Top of Page register on-chip points to the Main Definition Block, which in turn points to a linked list of RCBs.

The Am8052 allows for the separation of attribute and character lists. By extending the RCB, split screen segments can be constructed as in the case of RCB_2 in Figure 7. In parallel with the screen or background data structure, there exists a window structure which contains Window Control Blocks (WCBs) for each row of each window. Windows can exist in any position on the screen and are overlayed on top of the screen or background information. For example, the structure shown in Figure 8 could be used to implement a menu overlay at the top of the screen together with a status overlay.

MAIN DEFINITION BLOCK

The Main Definition Block is a set of control data and addresses, located in the system memory, which allow the user to specify screen oriented features. The TOP OF PAGE register points to the first word of the Main Definition Block. Cursor position, fill code and scroll rate are set by the appropriate fields within the block. The Main Definition Block also points to the first Row Control Block.

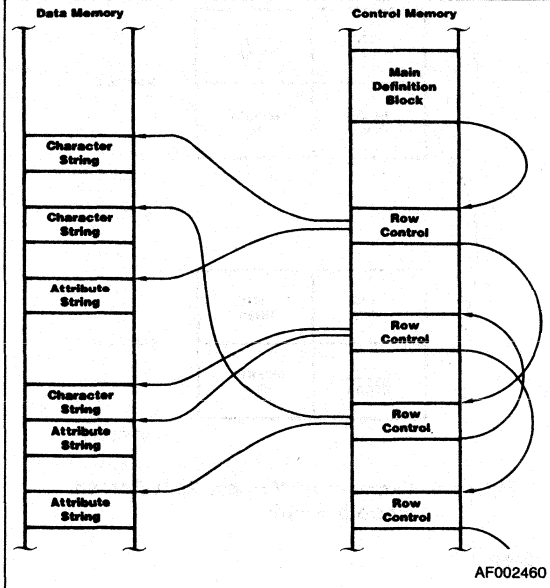


Figure 6. Am8052 Linked List Structure

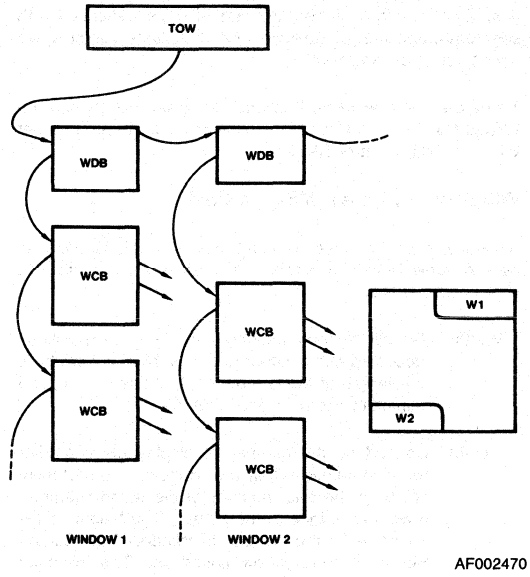


Figure 8. Window Data Structure

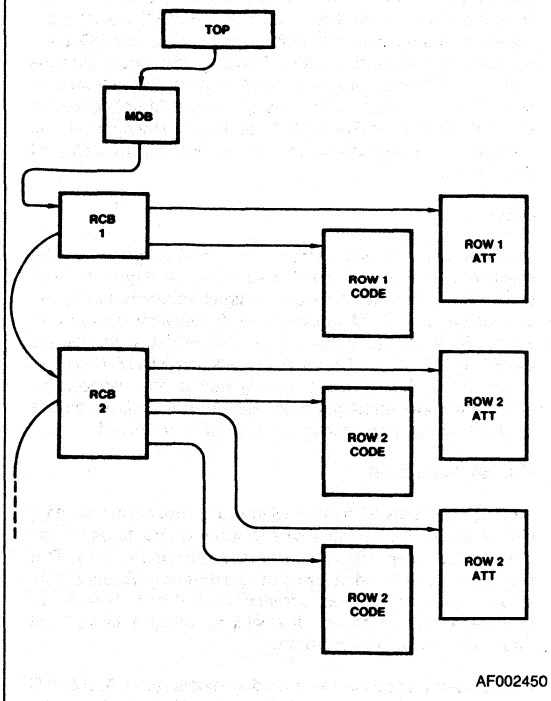


Figure 7. Background Data Structure

ROW CONTROL BLOCKS

The RCB Pointer in the Main Definition Block points to the first word of the first Row Control Block of the list. Each Row Control Block in the main chain is linked to the next via the RCB Pointer. Changing the RCB Pointer within the chain allows quick insertion or deletion of character rows.

Attributes associated with characters exist in their own separate lists. A character row may be composed of one or more segments of data. Each segment is a block of words with consecutive addresses. A Row Control Block has a character code pointer (2 words) and an attribute pointer (2 words) for each segment. A fifth word, HIDDEN # and VISIBLE #, defines the number of characters (byte count) contained in the segment as well as the number of displayed characters in the segment. Character attributes are in word format, and there can be as many character attributes as character codes.

WINDOW DEFINITION BLOCK

The Window Definition Block defines the size and location of the window. It is the header block for a list of Window Row Control Blocks and can also point to another Window Definition Block if more than one window is displayed on the screen. The TOP OF WINDOW register points to the first word of the first Window Definition Block. Within the first Window Definition Block, the WRCB Pointer points to the current window's first Window Row Control Block, while the next Window Definition Block Pointer points to the next window's Window Definition Block. Window size is specified by two words in the Window Definition Block. START WINDOW ROW # and END WINDOW ROW # are byte values which position the window vertically on the screen. The window display becomes active in the character row number specified by START WINDOW ROW # and will become inactive in the character row following END WINDOW ROW #.

WINDOW ROW CONTROL BLOCKS

The Window Row Control Blocks have the same format as the Row Control Blocks.

The WCB Pointer is the address link to the next row's Window Row Control Block. A window can also be described with segments, and the Window Row Control Block contains five words for each segment.

To hard-scroll a window, it is only necessary to change the WRCB Pointer in the Window Definition Block to an adjacent Window Row Control Block.

WINDOW DISPLAY MECHANISM

A window is any bounded area on the screen which is linked in by a Window Definition Block. The window has the following size characteristics:

Width: Defined by the number of character code positions occupied within a character row. Maximum width is the length of the line buffer (132 characters), and minimum width is one character.

Height: Defined by the number of displayable character rows contained within the window. The maximum height is the total number of displayed character rows on the face of the screen. The height limit is specified by the number of Window Row Control Blocks in the window linked list. The minimum height of a window is one row.

Window Positioning

The window is originally positioned to occupy any portion of the displayable character rows. It can be as large as the full screen or as small as one row high and one character wide. The window is always unscrolled when first displayed. (The counter holding the value of the first scan line of the uppermost character row of the window is reset.)

The window must be positioned horizontally such that its left- and right-hand sides begin and end at a background character row segment boundary. Any unfilled character positions within the window segment, and following the end of the window segment to the end of the line buffer (character position 131), are filled with the fill character code obtained from the Main Definition Block.

Multiple Windows

Multiple windows can be displayed simultaneously. Windows cannot be horizontally aligned to each other, and hence must be specified on non-overlapping character row boundaries (see section on virtual windows). Each window is defined by a Window Definition Block, and the scrolling windows are designated by a control bit within the Window Definition Block.

WINDOW POSITIONING

The window position is defined in the Window Definition Block. The coordinate units are background character rows and background character columns. When the background is scrolling, the window (or windows) remain stationary on the display.

EXAMPLE OF WINDOW OVERLAYS

The example (Figure 9) explains how windows are constructed using the Linked List feature that the Am8052 provides.

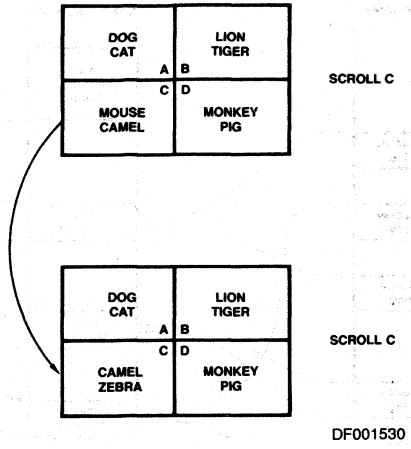


Figure 9. Example of Vertical Split Screen Smooth Scroll

Step 1

The first step toward constructing windows on a CRT screen is to split the screen horizontally and vertically using row control blocks with multiple data pointers. The data pointers in each RCB point to the first characters within each subscreen area defined by the horizontal/vertical splits. In this example, the RCB that controls the first character row (DOG/LION) contains two data pointers. The first points to subscreen DOG and the second to subscreen LION. The segment length information in the RCB indicates to the DMA when to switch from data field DOG to data field LION. The Linked List Structure for this example is shown in Figure 10. Note that in most applications, this split screen will have been set up prior to the invocation of the window.

Step 2

A window can now be overlayed on to the background by the creation of a window linked-list as shown in Figure 11. The scrollable window has a linked list structure pointed to by the Top of Window (TOW) pointer which functions similarly to TOP. The other information required for window definition is the START WINDOW CHAR # and END WINDOW CHAR # which define the start/end coordinates of the window. To effect a window scroll, just one change to the TOW value is required, which significantly relieves CPU overhead.

Virtual Windows

Although the rules of multiple windows do not permit overlapping windows, the background and window structures can be used to implement virtual horizontally aligned windows. This can be best described by using the illustration in Figure 9. The screen is divided into 4 subscreens: A, B, C and D; each can be independently defined as a window using a Linked List Structures similar to Figure 11.

If subscreen C is defined as a window, subscreens A, B and D are configured to be the background. Window C can be scrolled independently of the background by TOW pointer manipulation. Similarly, subscreen D can be defined as a window with A, B and C configured as background. Thus, two aligned subscreens can be independently defined as windows by intelligent use of linked list structures, giving the user the illusion of aligned windows.

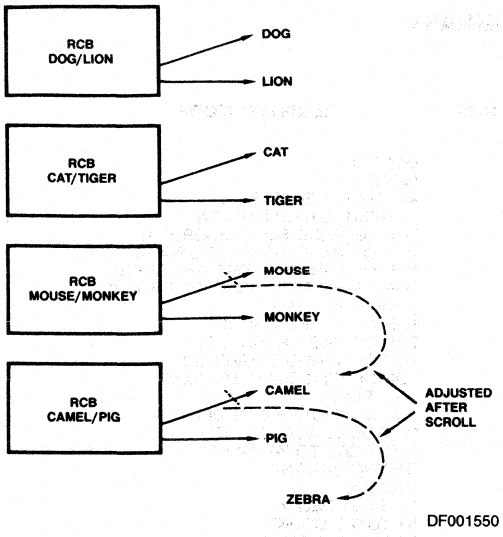


Figure 10. Split Screen Control Blocks

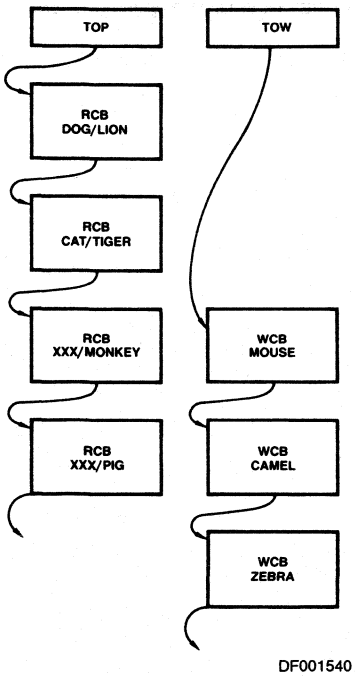


Figure 11. Window Overlay Structure

HORIZONTAL SCREEN FORMAT

The horizontal format defines the general timing of a single raster scan line. The scan line consists of two basic periods: visible raster line scan from left-to-right across the CRT screen and the right-to-left beam retrace period (or horizontal sync). The beam is always blanked during the retrace period. The front and back porch periods on either side of the horizontal sync are also blanked because no active video is desired during that time.

Horizontal scan frequencies range from a minimum of 15kHz for small screen, low bandwidth CRTs up to about 60kHz for 100MHz bandwidth large screen CRTs. The horizontal format versatility must accommodate this wide range of scan frequencies. The horizontal circuit generates two basic timing signals: horizontal sync and blanking. The horizontal blanking signal is "ored" with the vertical blanking signal prior to output at the BLANK pin.

HORIZONTAL TIMING CONTROL

Horizontal timing is controlled by the $\overline{RST}$ signal and the DE (Display Enable) bit in the mode register.

The HSYNC output is disabled (inactive) and the BLANK output active whenever the CRTC is reset by $\overline{RST}$ input (active low) or whenever the DE bit is reset (display disabled). $\overline{RST}$ active low is a hardware reset to the CRTC (this action also resets DE bit), and the DE bit is a software reset of the CRTC.

Am8052 VERTICAL SCREEN FORMAT

The vertical format defines the number of horizontal scan lines to be displayed in each frame. The front and rear porches, as well as the vertical retrace time, are also defined.

The CRTC operates in either an interlace or non-interlace mode. The I_1 bit, in Mode Register 1, determines if the CRTC will operate in the interlace or non-interlace mode. See below for each of the interlace options.

The Vertical Line Counter is clocked by either the horizontal sync rate in the non-interlaced or twice the horizontal sync rate in the interlaced mode. In non-interlaced mode all vertical frames (period between two vertical sync pulses) are *even*. In interlaced mode, the first vertical frame following a Display Enable (setting of DE bit in the Mode register) is always *even* and alternates between odd and even from there on.

EXTERNAL SYNC OPERATION

The ESYNC input allows synchronization of the CRT display vertical frame rate to the power line frequency to eliminate interference effects. The ES bit in Mode Register 1 specifies whether the ESYNC input is used to control the Vertical Sync rate.

The ESYNC input is recognized by the CRTC during every frame. It causes the VSYNC signal to become active at the occurrence of HSYNC. In non-interlaced mode, VSYNC becomes active at the rising edge of HSYNC active. In interlaced mode, VSYNC either becomes active at the next HSYNC, active when in the even frame, or active at the next half point between HSYNCs (2x HSYNC) in the odd frame.

INTERLACE

There are two types of interlace, Repeat Field Interlace (RFI) and Interlaced Video (IV). The effect of both schemes is to offset the vertical position of the scan lines of the odd numbered fields so that they will be physically interleaved with the scan lines of the even fields. For RFI, the same video information is displayed on both odd and even fields, the slight offset of the odd field tending to eliminate the horizontal stripes that sometimes occur between scan lines of non-interlaced displays.

Interlaced Video is used to increase the amount of information displayed on a monitor without increasing the horizontal or vertical scan rates. IV takes advantage of the odd field scan line offset by displaying half the video in the even field (alternating lines) and half in the odd field. The effect is to essentially double the vertical character density with respect to RFI or non-interlace.

REGISTER SUMMARY

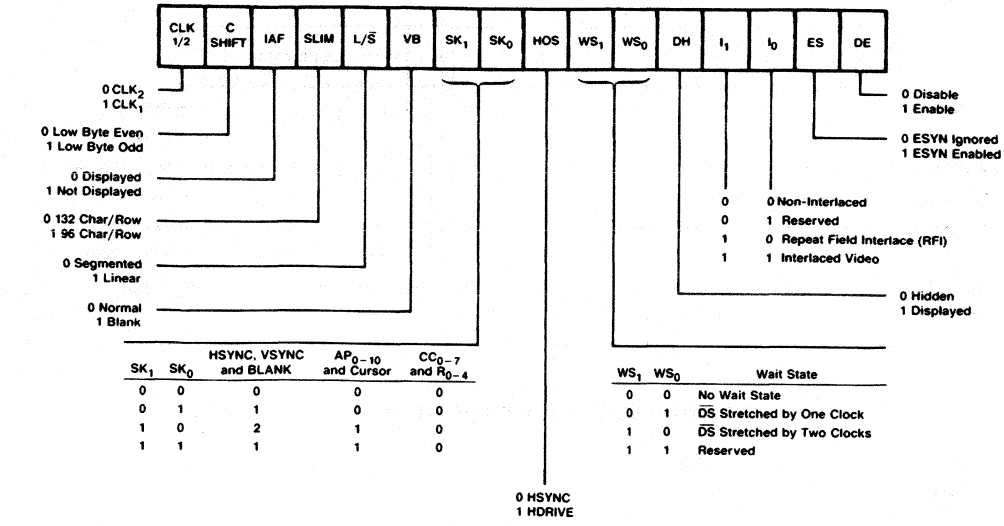
ADDRESS ($AD_4 - AD_0$)

BINARY	HEX	TYPE	ACTIVE BITS	REGISTER MODE
0 0 0 0 0	00	R/W	16	MODE 1
0 0 0 0 1	01	R/W	16	MODE 2
0 0 0 1 0	02	W	12	ATTRIBUTE ENABLE
0 0 0 1 1	03	W	5	ATTRIBUTE REDEFINITION
0 0 1 0 0	04	R/W	8	TOP OF PAGE SOFT (HI-ORDER)
0 0 1 0 1	05	R/W	16	TOP OF PAGE SOFT (LO-ORDER)
0 0 1 1 0	06	R/W	8	TOP OF WINDOW SOFT (HI-ORDER)
0 0 1 1 1	07	R/W	16	TOP OF WINDOW SOFT (LO-ORDER)
0 1 0 0 0	08	W	16	ATTRIBUTE FLAG
0 1 0 0 1	09	R/W	8	TOP OF PAGE HARD (HI)
0 1 0 1 0	0A	R/W	16	TOP OF PAGE HARD (LO)
0 1 0 1 1	0B	R/W	8	TOP OF WINDOW HARD (HI)
0 1 1 0 0	0C	R/W	16	TOP OF WINDOW HARD (LO)
1 0 0 0 0	10	W	16	DMA BURST
1 0 0 0 1	11	W	12	*VSYNC WIDTH/SCAN DELAY
1 0 0 1 0	12	W	12	*VERTICAL ACTIVE LINES
1 0 0 1 1	13	W	12	*VERTICAL TOTAL LINES
1 0 1 0 0	14	W	16	*HSYNC/VERTINT
1 0 1 0 1	15	W	9	*HDRIVE
1 0 1 1 0	16	W	9	*H SCAN DELAY
1 0 1 1 1	17	W	10	*H TOTAL COUNT
1 1 0 0 0	18	W	10	*H TOTAL DISPLAY

*These registers should only be accessed when Display Enable ("DE" bit in Mode Register 1) is reset, since they control the video timing signals.

MODE REGISTER 1

ADDRESS: 0 0 0 0 0
READ/WRITE



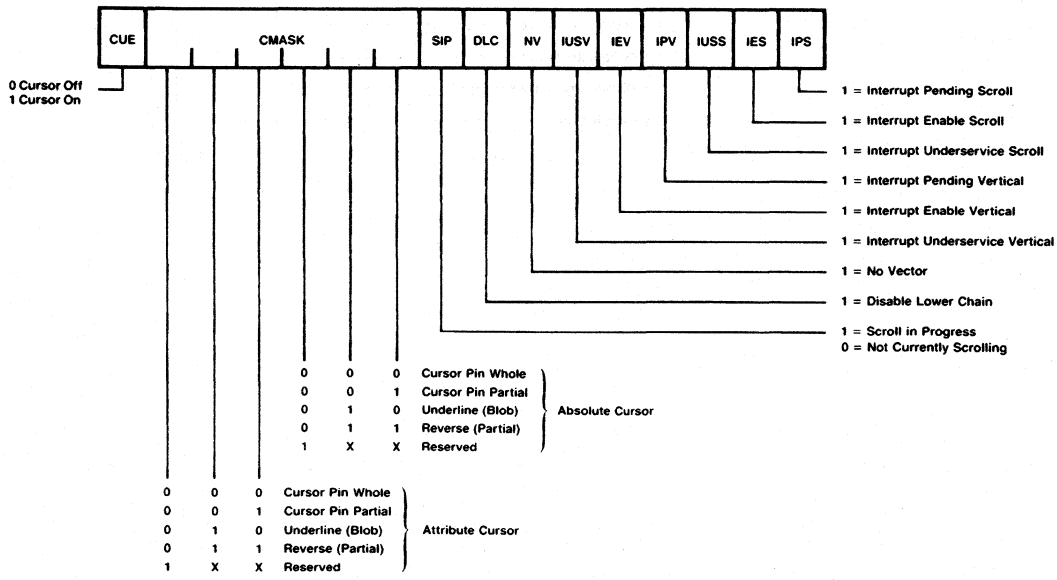
SK_1	SK_0	HSYNC, VSYNC and BLANK	AP_{0-10} and Cursor	CC_{0-7} and R_{0-4}
0	0	0	0	0
0	1	1	0	0
1	0	2	1	0
1	1	1	1	0

WS_1	WS_0	Wait State
0	0	No Wait State
0	1	$\overline{DS}$ Stretched by One Clock
1	0	$\overline{DS}$ Stretched by Two Clocks
1	1	Reserved

DF001561

MODE REGISTER 2

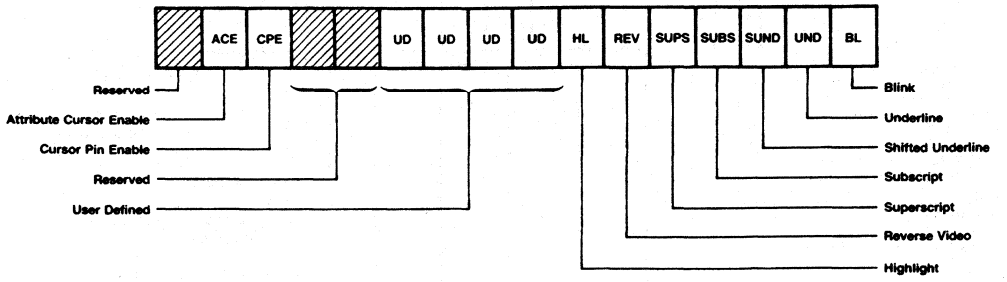
ADDRESS: 0 0 0 0 1
READ/WRITE



DF001570

ATTRIBUTE PORT ENABLE REGISTER

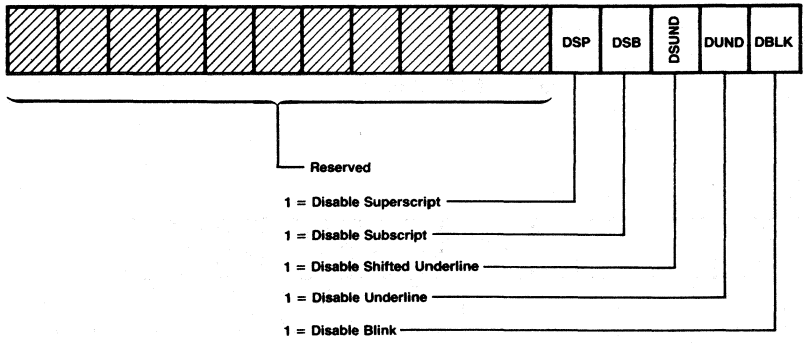
ADDRESS: 0 0 0 1 0
WRITE ONLY



DF001580

ATTRIBUTE REDEFINITION REGISTER

ADDRESS: 0 0 0 1 1
WRITE ONLY

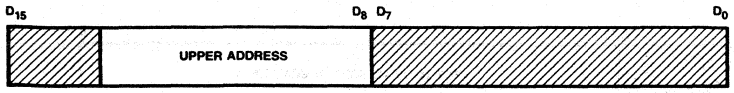


DF001590

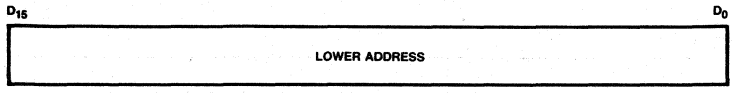
TOP OF PAGE/TOP OF WINDOW REGISTERS L/$\overline{S}$ = 0

READ/WRITE

ADDRESS	REGISTER	ACTIVE BITS
0 0 1 0 0	Top of Page Soft (HI)	14 . . . 8
0 0 1 0 1	Top of Page Soft (LO)	15 . . . 0
0 0 1 1 0	Top of Window Soft (HI)	14 . . . 8
0 0 1 1 1	Top of Window Soft (LO)	15 . . . 0
0 1 0 0 1	Top of Page Hard (HI)	14 . . . 8
0 1 0 1 0	Top of Page Hard (LO)	15 . . . 0
0 1 0 1 1	Top of Window Hard (HI)	14 . . . 8
0 1 1 0 0	Top Of Window Hard (LO)	15 . . . 0



DF001600

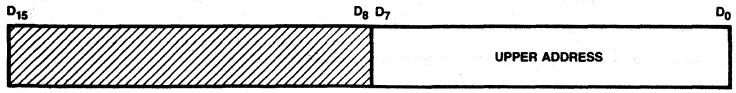


DF001610

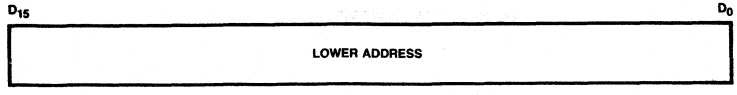
TOP OF PAGE/TOP OF WINDOW REGISTERS L/$\overline{S}$ = 1

READ/WRITE

ADDRESS	REGISTER	ACTIVE BITS
0 0 1 0 0	Top of Page Soft (HI)	7 . . . 0
0 0 1 0 1	Top of Page Soft (LO)	15 . . . 0
0 0 1 1 0	Top of Window Soft (HI)	7 . . . 0
0 0 1 1 1	Top of Window Soft (LO)	15 . . . 0
0 1 0 0 1	Top of Page Hard (HI)	7 . . . 0
0 1 0 1 0	Top of Page Hard (LO)	15 . . . 0
0 1 0 1 1	Top of Window Hard (HI)	7 . . . 0
0 1 1 0 0	Top of Window Hard (LO)	15 . . . 0



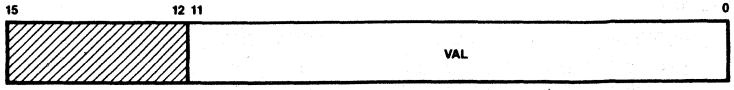
DF001620



DF001630

VERTICAL ACTIVE LINES REGISTER

ADDRESS: 1 0 0 1 0
WRITE ONLY



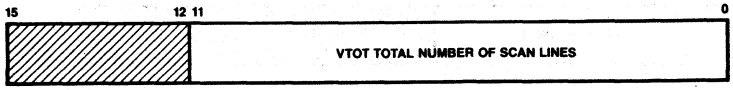
VSYNC ↓ TO VBLANK ↑ = VAL+1 NON-INTERLACED
(VAL*+1)/2 INTERLACED

DF001640

* Must be odd

VERTICAL TOTAL LINES REGISTER

ADDRESS: 1 0 0 1 1
WRITE ONLY



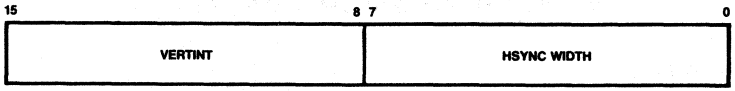
VSYNC TO VSYNC = VTOT+1 SCAN LINES NON-INTERLACED
= (VTOT**+1)/2 SCAN LINES INTERLACED

DF001651

** Must be even

HORIZONTAL SYNC AND VERTICAL INTERRUPT ROW REGISTER

ADDRESS: 1 0 1 0 0
WRITE ONLY



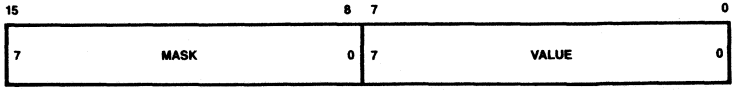
ROW NUMBER AT WHICH VERTICAL INTERRUPT OCCURS

IN NUMBER OF CLK_1 OR CLK_2 PERIODS DEPENDING ON $CLK_{1/2}$ IN MODE REGISTER 1

DF001661

ATTRIBUTE FLAG REGISTER

ADDRESS: 0 1 0 0 0
WRITE ONLY

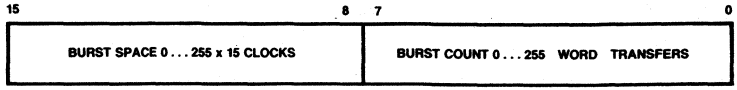


Note: When a mask-bit is set to 0, the corresponding value-bit must be 0.

DF001670

BURST REGISTER

ADDRESS: 1 0 0 0 0
WRITE ONLY



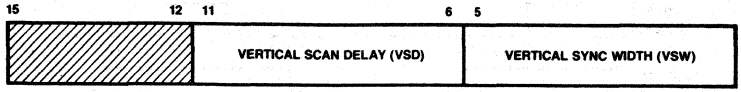
SPACE = 0 KEEPS BUS

COUNT = 0 NO DMA ACTIVITY

DF001681

VERTICAL SYNC WIDTH/VERTICAL SCAN DELAY REGISTER

ADDRESS: 1 0 0 0 1
WRITE ONLY



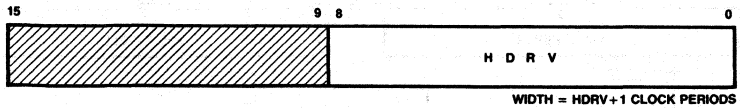
NON-INTERLACE DELAY = (VSD+1) SCAN LINES WIDTH = (VSW+1) SCAN LINES
INTERLACE DELAY = (VSD*+1)/2 SCAN LINES WIDTH = (VSW*+1)/2 SCAN LINES

*Must be odd

DF001690

HORIZONTAL DRIVE REGISTER

ADDRESS: 1 0 1 0 1
WRITE ONLY

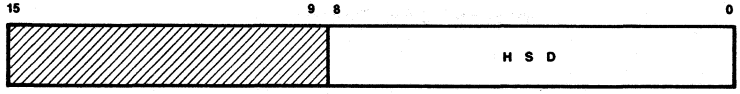


WIDTH = HDRV+1 CLOCK PERIODS

DF001700

HORIZONTAL SCAN DELAY REGISTER

ADDRESS: 1 0 1 1 0
WRITE ONLY

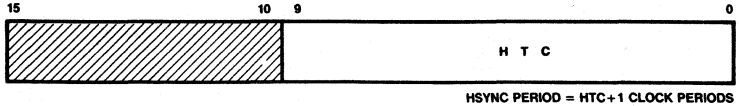


DELAY = HSD+1 CLOCK PERIODS

DF001710

HORIZONTAL TOTAL COUNT REGISTER

ADDRESS: 1 0 1 1 1
WRITE ONLY

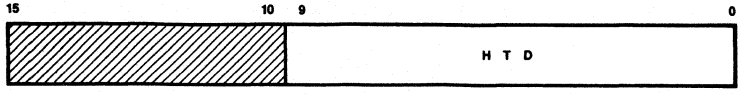


HSYNC PERIOD = HTC+1 CLOCK PERIODS

DF001721

HORIZONTAL TOTAL DISPLAY REGISTER

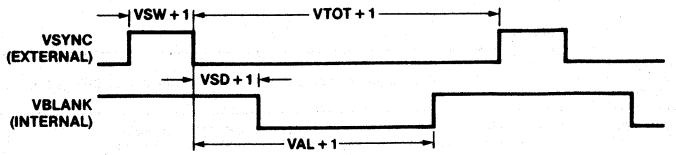
ADDRESS: 1 1 0 0 0
WRITE ONLY



HSYNC TO BLANK INTERVAL = HTD+1 CLOCK PERIODS
(MUST BE ODD IN INTERLACE MODE)

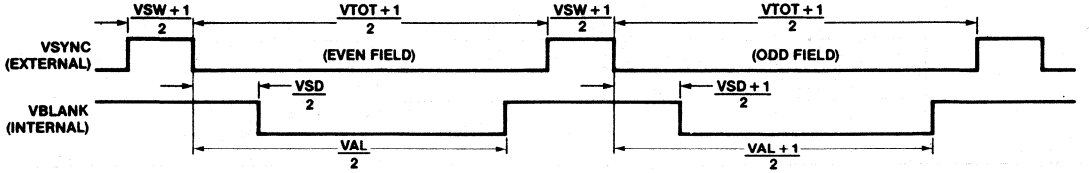
DF001730

Non-Interlaced Video Vertical Sync Timing



WF008910

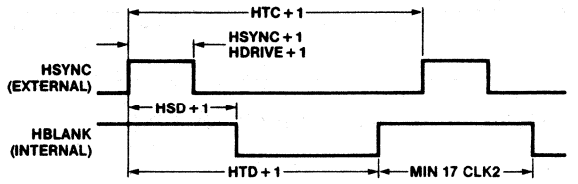
RFI and Video Interlace Sync Timing



NOTE: VSD, VSW, VAL MUST BE ODD
VTOT MUST BE EVEN

WF008920

Horizontal Sync Timing



WF008930

FRAME TIMING SIGNALS SUMMARY:

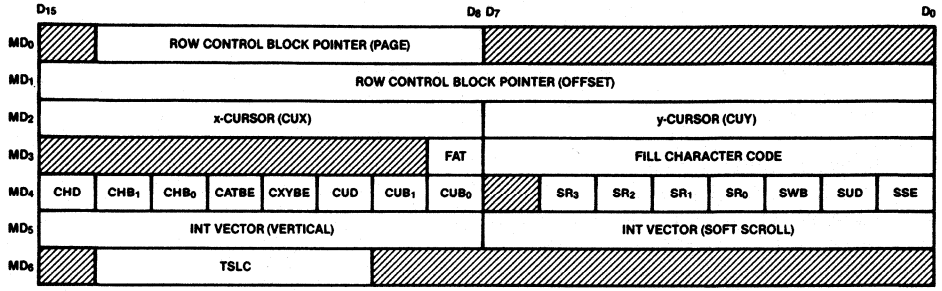
NON-INTERLACED MODE

VERTICAL SYNC WIDTH	VSW + 1
FRONT PORCH (VBLANK R.E. TO VSYNC R.E.)	VTOT – VAL
BACK PORCH (VSYNC F.E. to VBLANK F.E.)	VSD + 1
VSYNC F.E. TO NEXT VBLANK R.E.	VAL + 1
TOTAL SCAN LINES/FRAME-VSYNC WIDTH	VTOT + 1
HORIZONTAL SYNC WIDTH	HSYNC + 1
HORIZONTAL SYNC PERIOD	HTC + 1
HSYNC R.E. TO NEXT HBLANK R.E.	HTD + 1
HSYNC R.E. TO HBLANK F.E.	HSD + 1
HDRIVE R.E. TO HDRIVE F.E.	HDRV + 1

INTERLACED MODE

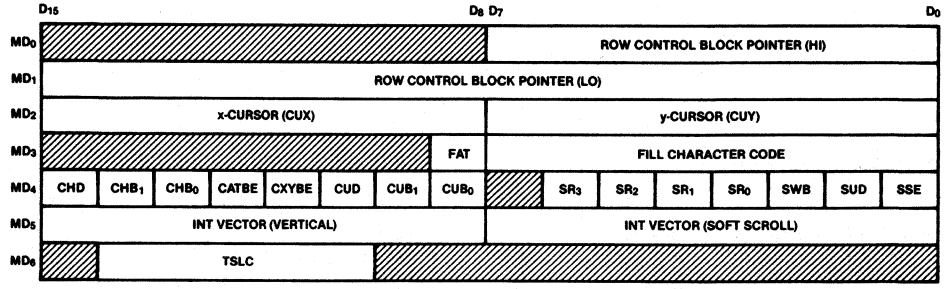
VERTICAL SYNC WIDTH	(VSW + 1)/2, VSW ODD	
BACK PORCH	VSD/2, EVEN FIELDV (VSD + 1)/2, ODD FIELD	} VAL ODD
VSYNC F.E. TO NEXT VBLANK R.E.	(VAL + 1)/2, ODD FIELD VAL/2, EVEN FIELD	} VSD ODD
TOTAL SCAN LINES/FRAME – VSYNC WIDTH	(VTOT + 1)/2, VTOT EVEN	
HORIZONTAL SYNC WIDTH	HSYNC + 1	
HORIZONTAL SYNC PERIOD	HTC + 1	
HSYNC R.E. TO NEXT HBLANK R.E.	HTD + 1, HTD ODD	
HSYNC E. TO HBLANK F.E.	HSD + 1	
HDRIVE R.E. TO HDRIVE F.E.	HDRV + 1	
FRONT PORCH (VBLANK R.E. to VSYNC R.E.)	(VTOT-VAL)/2, EVEN FIELD (VTOT + 1-VAL)/2, ODD FIELD VAL ODD, VTOT EVEN	

Main Definition Block (L/$\overline{S}$ = 0)



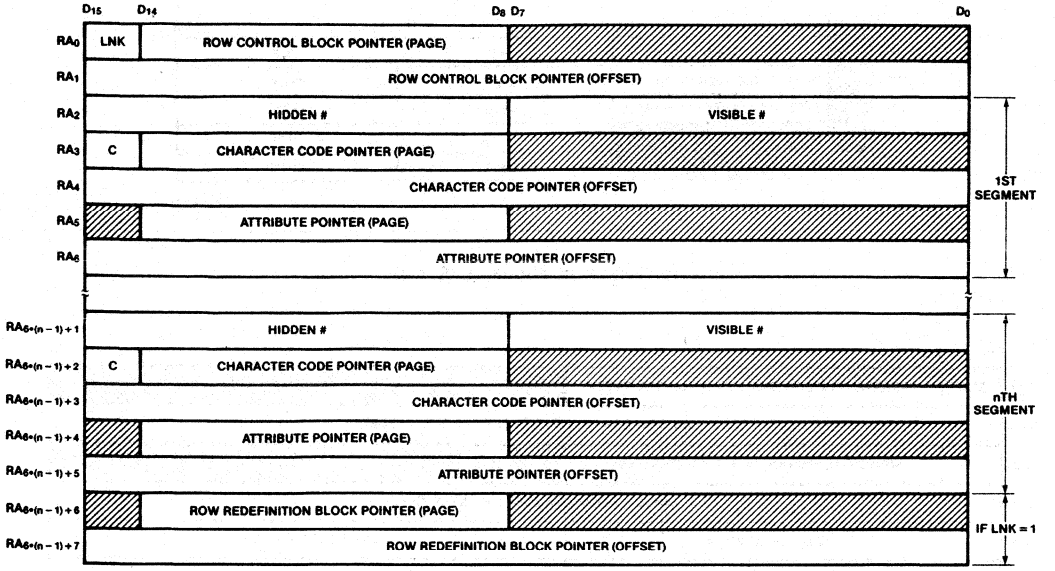
DF003990

Main Definition Block (L/$\overline{S}$ = 1)



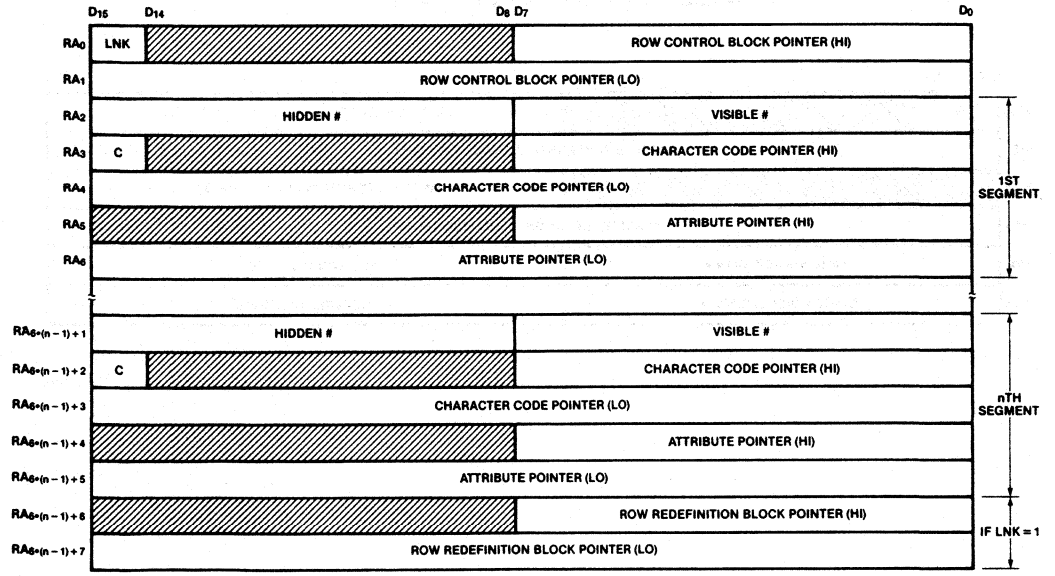
DF004000

Row Control Block (L/$\overline{S}$ = 0)



DF004010

Row Control Block (L/$\overline{S}$ = 1)



DF004020

Row Redefinition Block

	D_{15} D_{14}	D_{10} D_9	D_5 D_4	D_0
RR_0	TSLC	NCS		NCE
RR_1	ROW ATTRIBUTES (AP_{10}–AP_6)	SPCS		SPCE
RR_2	ROW ATTRIBUTES (AP_4–AP_0)	SBCS		SBCE
RR_3		CURS		CURE
RR_4	DR_1 DR_0	UND		SUND

DF004030

Window Definition Block (L/$\overline{S}$ = 0)

	D_{15}	D_{14}	D_8 D_7	D_0
WD_0	SCW	WINDOW ROW CONTROL BLOCK POINTER (PAGE)		
WD_1	WINDOW ROW CONTROL BLOCK POINTER (OFFSET)			
WD_2	0	WINDOW DEFINITION BLOCK POINTER (PAGE)		
WD_3	WINDOW DEFINITION BLOCK POINTER (OFFSET)			
WD_4	START WINDOW ROW #		END WINDOW ROW #	
WD_5	START WINDOW CHAR #		END WINDOW CHAR #	

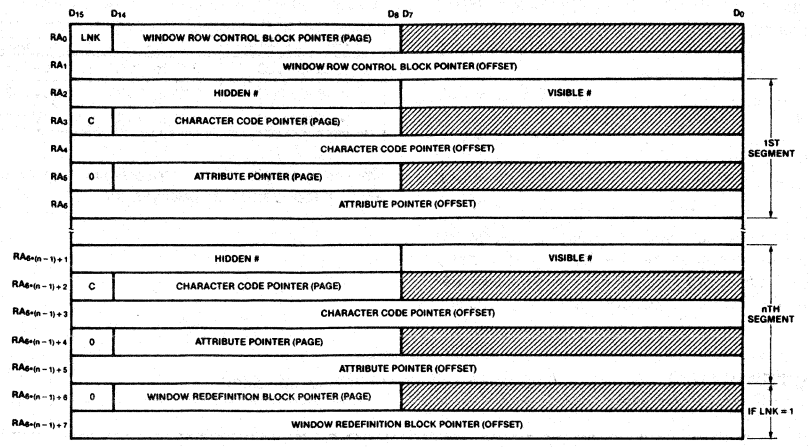
DF004040

Window Definition Block (L/$\overline{S}$ = 1)

	D_{15}	D_{14}	D_8 D_7	D_0
WD_0	SCW		WINDOW ROW CONTROL BLOCK POINTER (HI)	
WD_1	WINDOW ROW CONTROL BLOCK POINTER (LO)			
WD_2			WINDOW DEFINITION BLOCK POINTER (HI)	
WD_3	WINDOW DEFINITION BLOCK POINTER (LO)			
WD_4	START WINDOW ROW #		END WINDOW ROW #	
WD_5	START WINDOW CHAR #		END WINDOW CHAR #	

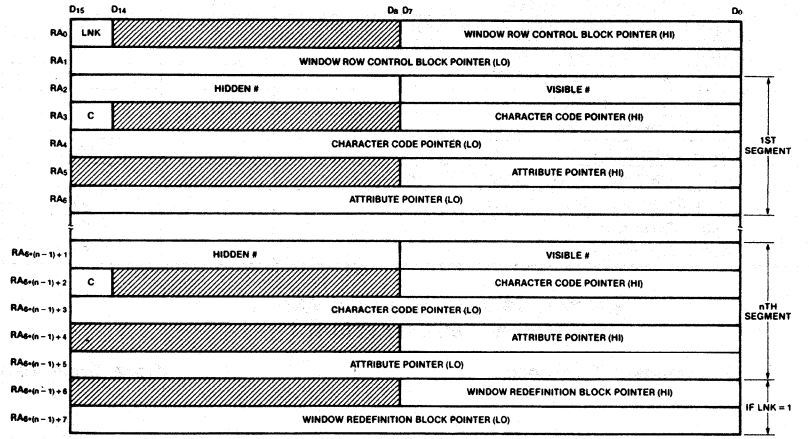
DF004050

Window Row Control Block (L/$\overline{S}$ = 0)



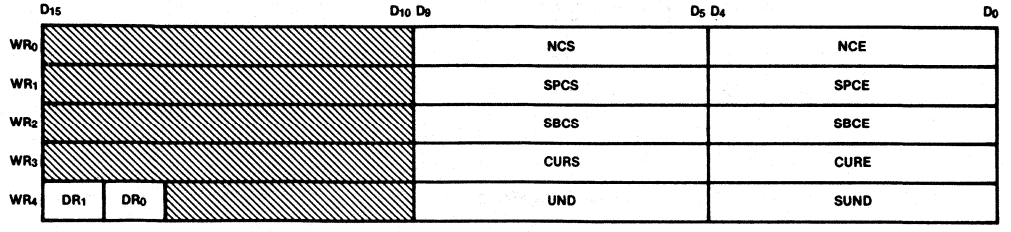
DF004070

Window Row Control Block (L/$\overline{S}$ = 1)



DF004080

Window Redefinition Block



DF004090

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Temperature Ambient
under Bias ..... −65 to +125°C
Supply Voltage to Ground Potential
Continuous ..... −0.5 to +7.0V
DC Voltage Applied to Outputs for High
Output State ..... −0.5V to +V_{CC}
DC Input Voltage ..... −0.5 to +7.0V
DC Output Current into Outputs 30mA
DC Input Current ..... −30 to +5.0mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	V_{SS}
Am8052-5LC	0°C to 70°C	5.0V ±5%	0V
Am8052-6LC	0°C to 70°C	5.0V ±5%	0V
Am8052-8LC	0°C to 70°C	5.0V ±5%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

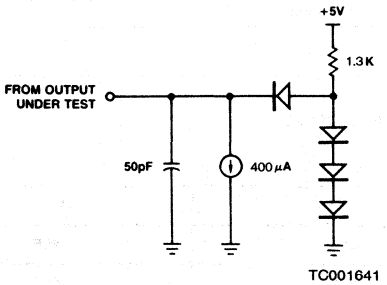
DC CHARACTERISTICS over operating range unless otherwise specified

T_A = 0°C to +70°C; V_{CC} = 5V ±5%

Parameters	Description	Min	Max	Units
V_{OH}	Output High Voltage (I_{OH} = 400μA)	2.4		V
V_{OL}	Output Low Voltage (I_{OL} = 3.2mA)	0	0.4	V
V_{IH}	Input High Voltage (except CLK_1 and CLK_2)	2.0	V_{CC} +0.5	V
V_{CIH}	CLK_1/CLK_2 Input High Voltage	4.0		V
V_{IL}	Input Low Voltage (except CLK_1 and CLK_2)	−0.5	0.8	V
V_{CIL}	CLK_1/CLK_2 Input Low Voltage		0.3	V
I_{IX}	Input Load Current (except $\overline{RSTT}$)		±10	μA
I_{IXR}	Input Load Current ($\overline{RSTT}$)		±100	μA
I_O	Output Leakage Current		±10	μA
I_{CC}	Supply Current		500	mA
C_{IN}	Input Capacitance (all pins except CLK_1 and CLK_2)		15	pF
C_{CIN}	Input Capacitance, CLK_1 and CLK_2		80	pF
C_{OUT}	Output Capacitance		50	pF
$C_{I/O}$	Bidirectional Capacitance		50	pF

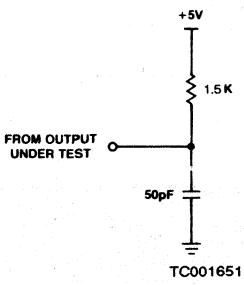
SWITCHING TEST CIRCUITS AND SWITCHING TEST WAVEFORM

Standard Test Load



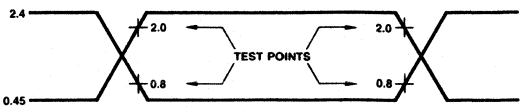
TC001641

Open Drain Test Load



TC001651

Input Waveform



WF007430

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 BUS MASTER READ/WRITE

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
1	t_{PHL}	CLK_1 ↑ TO $\overline{AS}$ ↓		55		55		45
2	t_{PLH}	CLK ↓ TO $\overline{AS}$ ↑		55		55		45
3	t_{PW}	$\overline{AS}$ PULSE WIDTH	60		60		45	
4	t_S	ADDRESS VALID TO $\overline{AS}$ ↑	40		40		30	
5	t_H	ADDRESS FROM $\overline{AS}$ ↑	20		20		20	
6	t_{PHL}	CLK_1 ↑ TO $\overline{DS}$ ↓		65		65		45
7	t_S	DATA IN TO CLK_1 ↓	15		15		10	
8	t_H	DATA IN FROM $\overline{DS}$ ↑	−10		−10		0	
9	t_{PLH}	CLK_1 ↓ TO $\overline{DS}$ ↑		65		65		45
10	t_{PLH}	CLK_1 ↑ TO R/$\overline{W}$	0	55	0	55	0	45
11	t_H	CLK_1 ↓ TO $\overline{DREN}$ ↑		65		65		65
12	t_S	$\overline{WAIT}$ VALID TO CLK_1↓	15		15		10	
13	t_H	$\overline{WAIT}$ FROM CLK_1 ↓	20		20		20	
14	t_{PHL}	CLK_1 ↓ TO $\overline{DREN}$ ↓		55		55		45
15	t_H	$\overline{DREN}$ FROM $\overline{DS}$ ↑	−10		−10		0	
16	t_R, t_F	RISE, FALL TIME, CLK_1		15		15		10
17	t_{PHL}	CLK_1 ↑ TO DTEN ↓		55		55		45
18	t_{PLH}	CLK_1 ↑ TO DTEN ↑		55		55		45
19	t_{PW}	CLK_1 HIGH PULSE WIDTH	85	500	70	500	45	500
20	t_{PW}	CLK_1 LOW PULSE WIDTH	75	500	70	500	45	500
40	t_{CYC}	CLK_1 PERIOD	200	1000	165	1000	125	1000
41	t_{AVDV}	ADD VALID TO DATA IN (Note)		395		310		225
42	t_{ASDV}	$\overline{AS}$ ↑ TO DATA VALID (Note)		315		245		185
43	t_{DSDV}	$\overline{DS}$ ↓ TO DATA VALID (Note)		205		155		115
46	t_{DRT}	$\overline{DREN}$ ↑ TO $\overline{DTEN}$ ↓	20		20		20	

Notes:

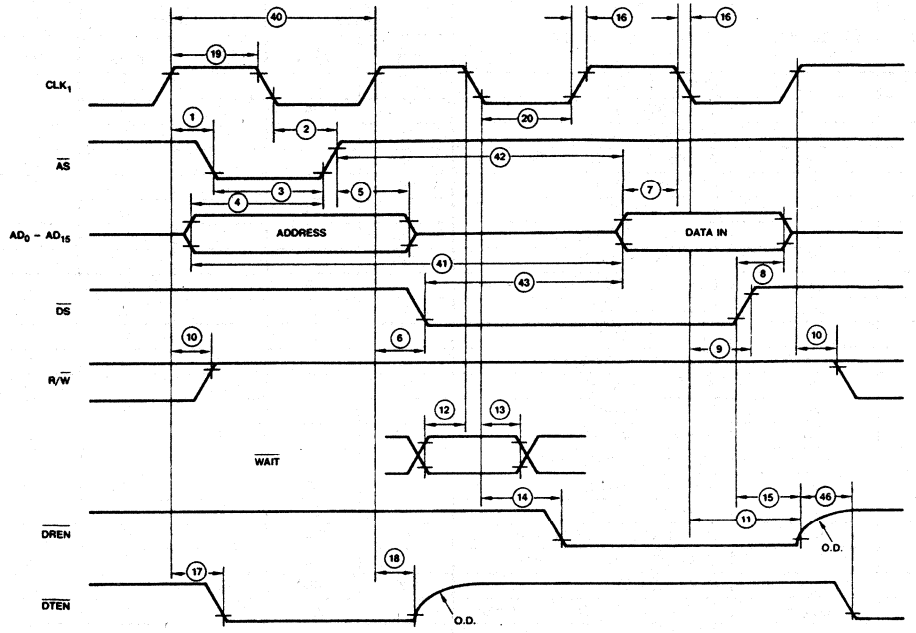
㊶ = 2·㊵ + ⑲ − ① − ③ + ④ − ⑦

㊷ = 2·㊵ − ② − ⑦ − ⑯

㊸ = ㊵ + ⑲ − ⑥ − ⑦

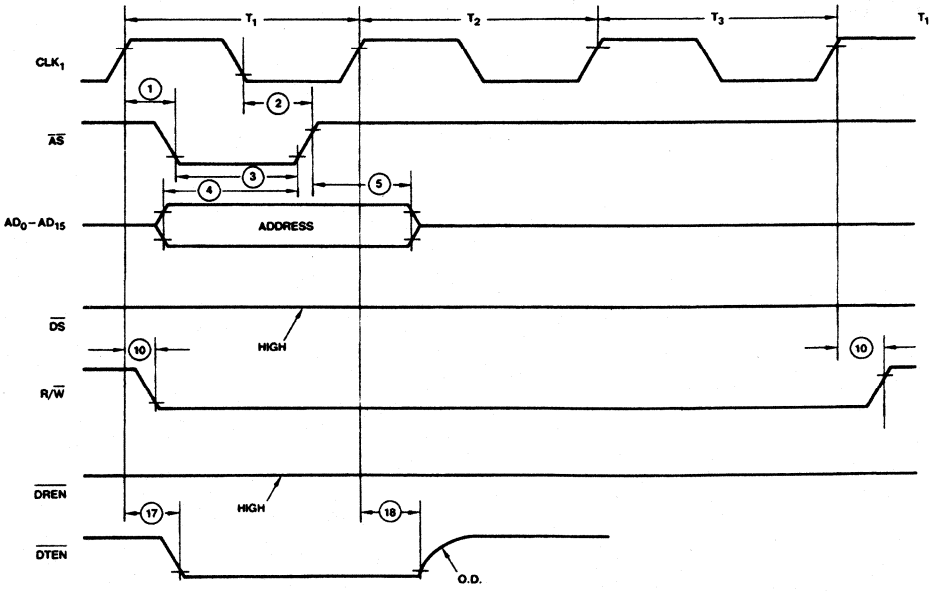
(At max CLK_1 freq.)

8052 BUS MASTER READ



WF004360

8052 BUS MASTER WRITE



WF004370

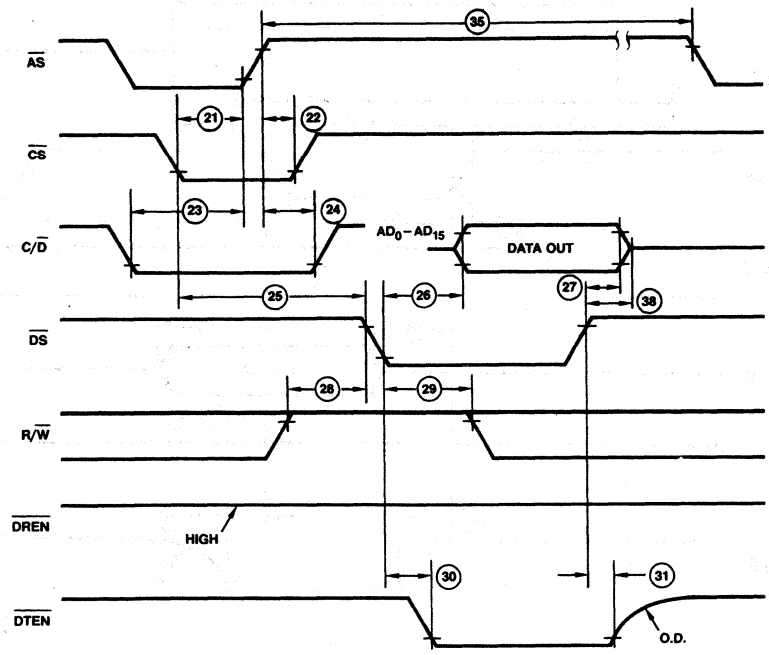
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 BUS SLAVE READ LATCHED

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
21	t_S	$\overline{CS}$ ↓ TO $\overline{AS}$ ↑	0		0		0	
22	t_H	$\overline{CS}$ LOW FROM $\overline{AS}$ ↑	25		25		20	
23	t_S	C/D TO AS ↑	0		0		0	
24	t_H	$\overline{C/D}$ FROM $\overline{AS}$ ↑	25		25		20	
25	t_{PD}	$\overline{CS}$ ↓ TO $\overline{DS}$ ↓	40		40		30	
26	t_{DSDV}	$\overline{DS}$ ↓ TO DATA VALID		180		180		150
27	t_H	DATA VALID FROM DS↑	15		15		10	
28	t_S	R/$\overline{W}$ TO $\overline{DS}$ ↓	10		10		0	
29	t_H	R/$\overline{W}$ VALID FROM $\overline{DS}$ ↓	40		40		40	
30	t_{PD}	DELAY FROM $\overline{DS}$ ↓		55		55		45
31	t_{PD}	DELAY FROM $\overline{DS}$ ↑ TO $\overline{DTEN}$ ↑		55		.55		45
35	t_{SRT}	SLAVE RECOVERY TIME	440		440		330	
38	t_Z	DS ↑ TO $AD_0 - AD_{15}$ HI-Z	10	60	10	60	10	50

Notes: 1. R/W latched internally by DS ↓.
2. CS latched internally by AS ↑.
3. C/D latched internally by AS ↑.
4. t_{SRT}: ㉟ = 3·㊵ − ③.

8052 BUS SLAVE READ LATCHED



WF004380

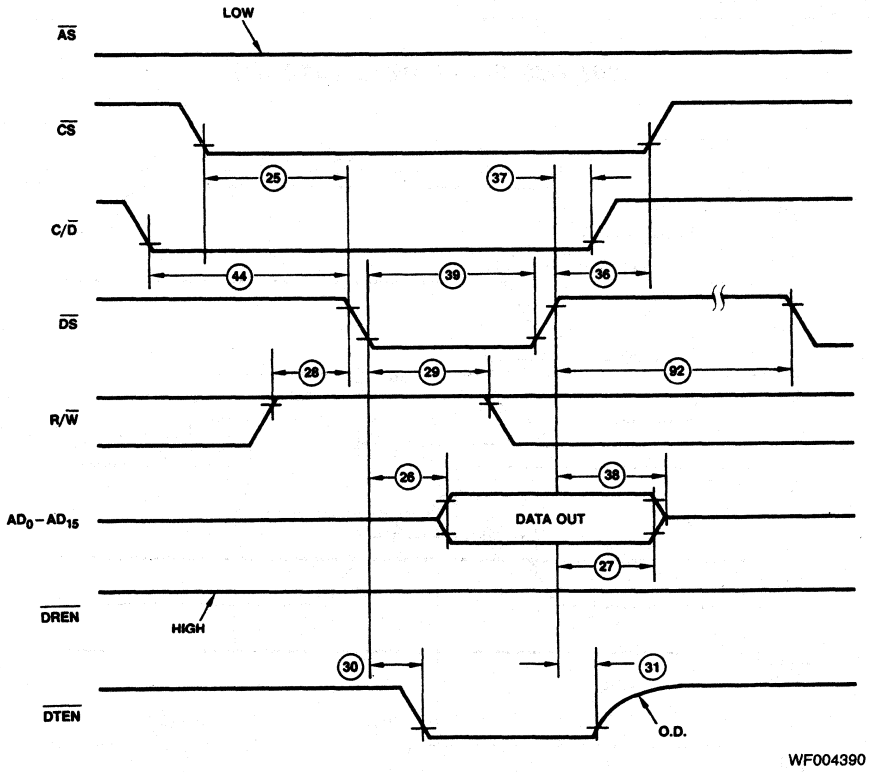
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 BUS SLAVE READ UNLATCHED

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
36	t_H	$\overline{CS}$ LOW FROM $\overline{DS}$ ↑	7		7		5	
37	t_H	C/$\overline{D}$ LOW FROM $\overline{DS}$ ↑	7		7		5	
39	t_{PW}	$\overline{DS}$ ↓ TO DS ↑ READ	200		200		150	
44	t_S	C/$\overline{D}$ TO $\overline{DS}$ ↓	40		40		30	
92	t_{SRT}	SLAVE RECOVERY TIME (Note 1)	300		300		225	

Note 1: t_{SRT}: ⑨② = 3·④⓪ − ③⑨.

8052 BUS SLAVE READ UNLATCHED



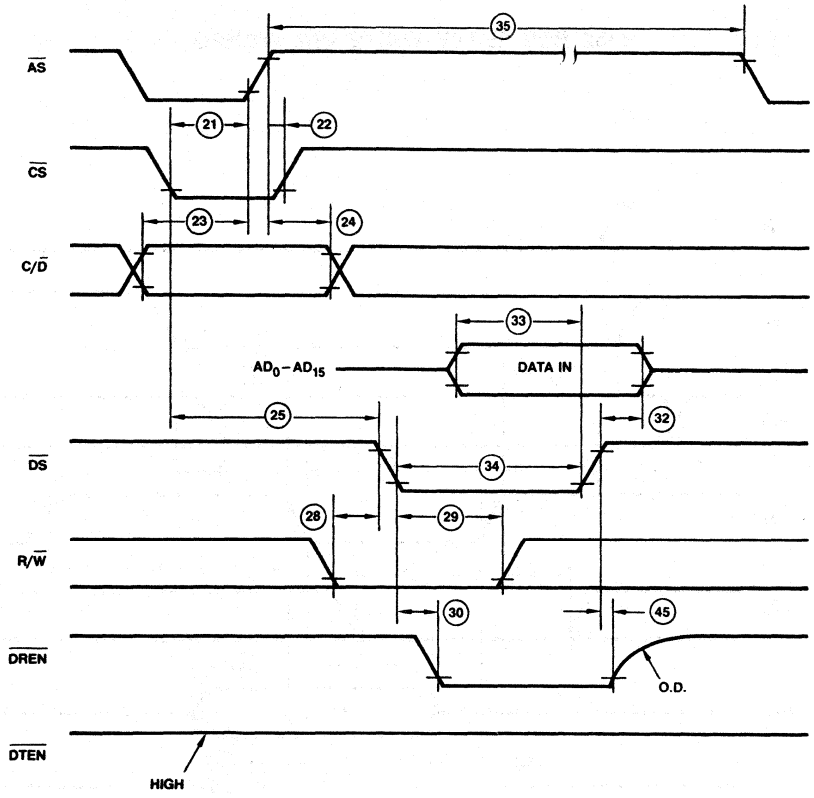
WF004390

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 BUS SLAVE WRITE LATCHED

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
32	t_H	DATA IN VALID FROM $\overline{DS}$ ↑	20		20		20	
33	t_S	DATA IN VALID TO $\overline{DS}$ ↑	90		90		80	
34	t_{PW}	DS PULSE WIDTH	135		135		100	
45	t_H	DELAY FROM $\overline{DS}$ ↑ TO $\overline{DREN}$ ↑	20	70	20	70	20	70

8052 BUS SLAVE WRITE LATCHED



WF004400

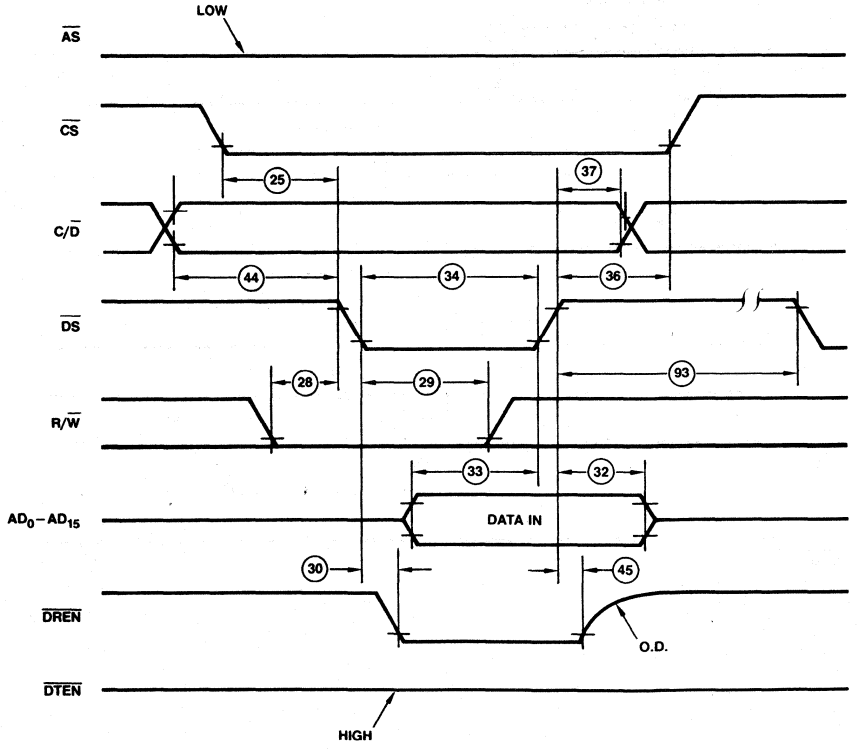
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 BUS SLAVE WRITE UNLATCHED

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
93	t_{SCT}	SLAVE RECOVERY TIME (Note 1)	365		365		275	

Note 1: t_{SCT}: ⑨③ = 3·④⓪ − ③④.

8052 BUS SLAVE WRITE UNLATCHED



WF004411

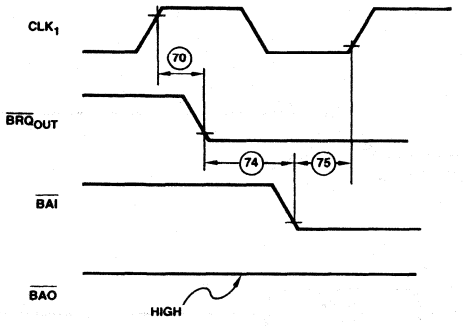
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 BUS EXCHANGE

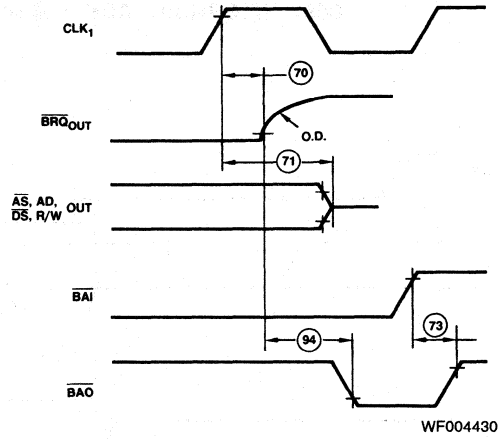
Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
70	t_{PD}	CLK_1 ↑ TO $\overline{BRQ}$ OUT		115		115		100
71	t_{PZ}	CLK_1 ↑ TO FLOAT		115		115		100
72								
73	t_{PD}	$\overline{BAI}$ TO $\overline{BAO}$		50		50		40
74	t_{PD}	$\overline{BRQ}$ ↓ TO $\overline{BAI}$ ↓ DELAY	0		0		0	
75	t_S	$\overline{BAI}$ ↓ TO CLK_1 ↑ (Note 1)	50		50		40	
94	t_{PD}	$\overline{BRQ}$ ↑ TO $\overline{BAO}$ ↓	0	60	0	60	0	50

Note 1: This parameter for testing only.

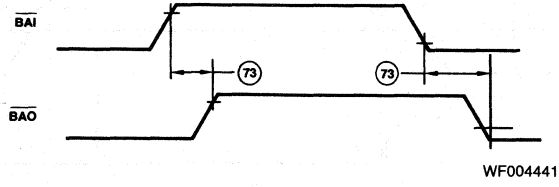
8052 BUS EXCHANGE



REQUESTING



RELEASING



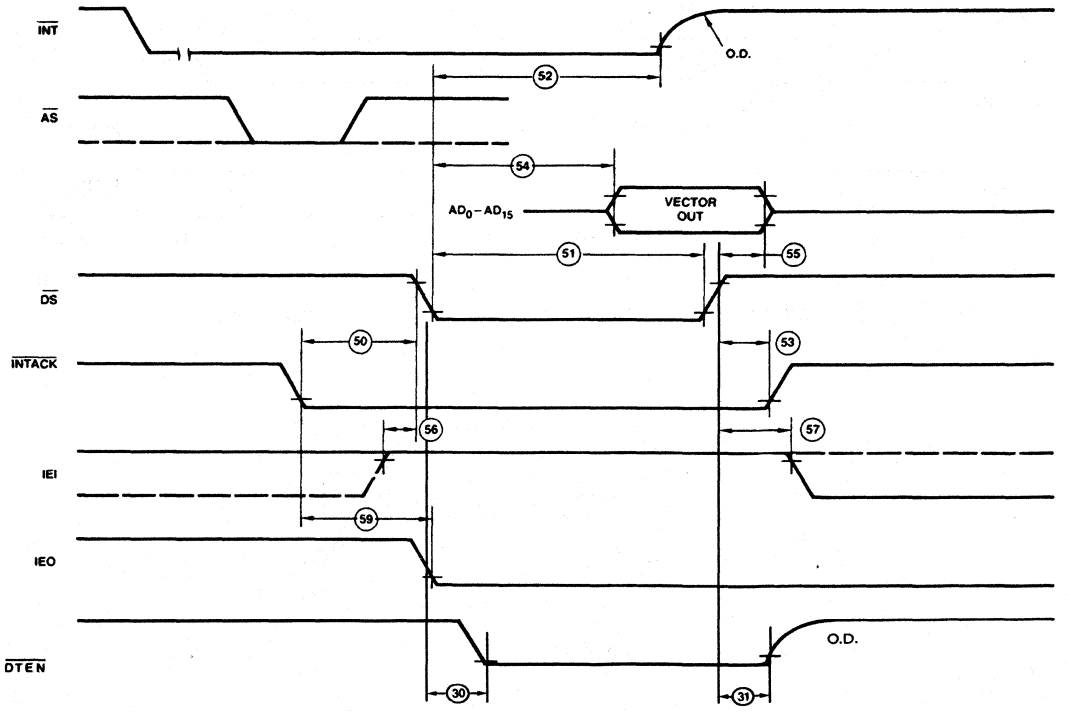
CHAIN DELAY

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 INTERRUPT ACK TIMING – DEVICE ACKNOWLEDGED

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
50	t_S	$\overline{INTACK}$ TO $\overline{DS}$ ↓	230		230		200	
51	t_{PW}	$\overline{DS}$ ↓ TO $\overline{DS}$ ↑ ACK	200		200		150	
52	t_{PD}	$\overline{DS}$ ↓ TO INT ↑		230		230		200
53	t_H	$\overline{INTACK}$ FROM $\overline{DS}$ ↑	0		0		0	
54	t_D	$\overline{DS}$ ↓ TO VECTOR VALID		180		180		150
55	t_H	VECTOR FROM $\overline{DS}$ ↑	0		0		0	
56	t_S	IEI TO $\overline{DS}$ ↓	90		90		80	
57	t_H	IEI FROM $\overline{DS}$ ↑	0		0		0	
59	t_D	$\overline{INTACK}$ TO IEO ↓ (IEI = H)		170		170		150

8052 INTERRUPT ACK TIMING – DEVICE ACKNOWLEDGED



WF004452

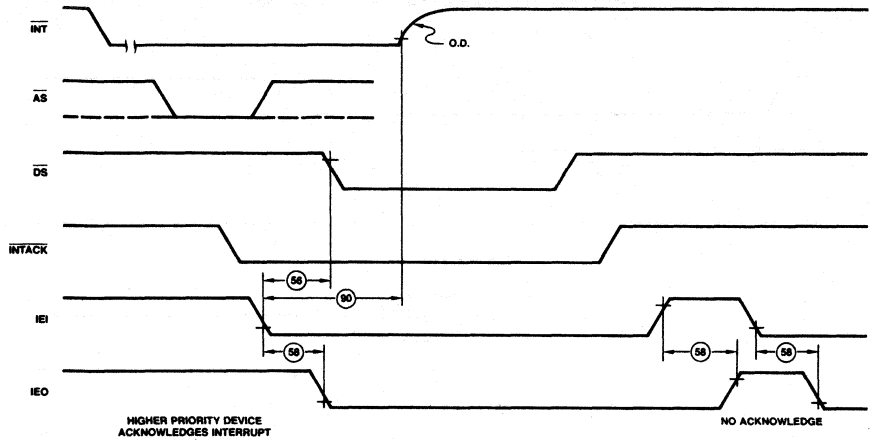
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 INTERRUPT ACK TIMING – LOW PRIORITY

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
56	t_S	IEI TO $\overline{DS}$ ↓	90		90		80	
58	t_D	IEI TO IEO		90		90		80
90	t_D	IEI ↓ TO $\overline{INT}$ ↑ (Note 1)		90		90		80

Note 1: INT terminated by an acknowledge higher on chain.

8052 INTERRUPT ACK TIMING – LOW PRIORITY



WF004460

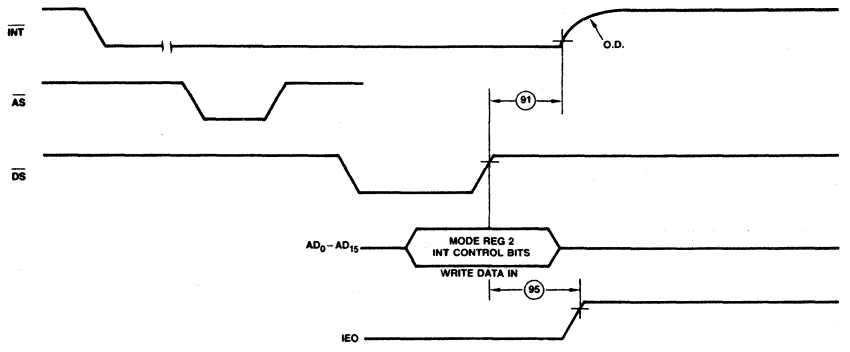
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 NON-VECTORED INT TIMING

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
91	t_D	$\overline{DS}$ ↑ TO $\overline{INT}$ (Write) (Note 1)		90		90		80
95	t_D	$\overline{DS}$ ↑ TO IEO ↑ (Write) (Note 2)		90		90		80

Notes: 1. This parameter describes the termination of an interrupt request via a write to the appropriate bit in Mode Reg 2:
IUSS → 1 IUSV → 1
IES → 0 IES → 0
IPS → 0 IPV → 0
2. This is the release of IEO LOW due to the slave mode reset of the IUS bit in Mode Reg 2.

8052 NON-VECTORED INT TIMING



WF004470

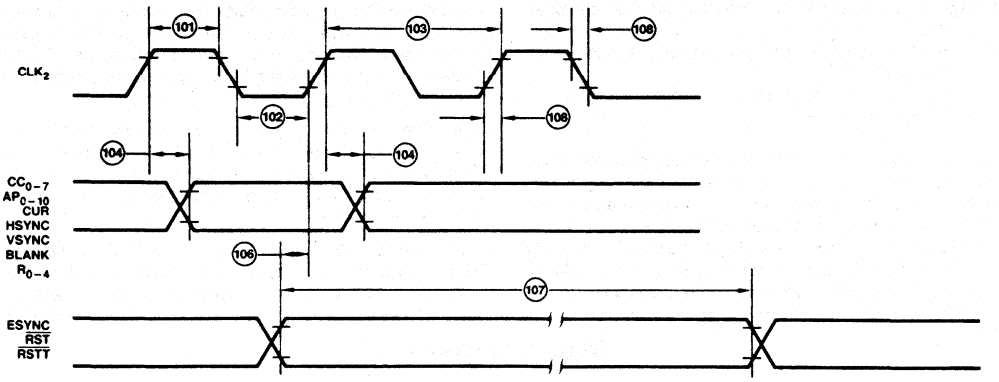
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

8052 VIDEO OUTPUTS AND SYNCHRONIZING INPUT TIMING

Numbers	Parameters	Description	5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
101	t_{PW}	CLK_2 HIGH PULSE WIDTH	85	500	70	500	35	500
102	t_{PW}	CLK_2 LOW PULSE WIDTH	75	500	70	500	35	500
103	t_{CYC}	CLK_2 PERIOD	200	1000	165	1000	100	1000
104	t_{DC}	CLK_2 ↑ TO OUTPUT DELAY		55		55		35
106	t_S	INPUT SETUP TO CLK_2 ↑ (Note 1)	60		60		50	
107	t_W	INPUT PULSE WIDTH (Note 2)	5T		5T		5T	
108	t_R, t_F	CLK_2 RISE, FALL TIME		15		15		10

Notes: 1. Parameter 106 is specified for test purposes only.
2. Parameter 107 is for reset only. T = CLK_2 period.

8052 VIDEO OUTPUTS AND SYNCHRONIZING INPUT TIMING



WF004480

Am8152A/53A

Video System Controller (VSC)

DISTINCTIVE CHARACTERISTICS

- 100MHz Video Dot Rate
- Four-level current driven (75Ω) differential video output
- Digital Video output ECL option (Am8152A provides TTL option)
- On-board crystal driven oscillator
- Proportional Spacing Support (2–17 dots)
- 9-bit dot data parallel input, with expansion capability to seventeen bits
- Trailing blanks (0–3 dots)
- Double Width Characters
- Attribute Support: Character Blink, Underline, Overstrike, Reverse, and Highlight
- Buffered and Synchronized Character Clock Outputs
- Background color selection
- Buffered and Synchronized Vertical and Horizontal Sync Outputs

GENERAL DESCRIPTION

The Am8152A/53A Video System Controller (VSC) provides interface between a CRT controller and a CRT monitor. The basic chip functions are:

- Support proportional and non-proportional character display
- Correctly synchronize and mix character attributes with video signals
- Output the video information in a four-level analog or digital format

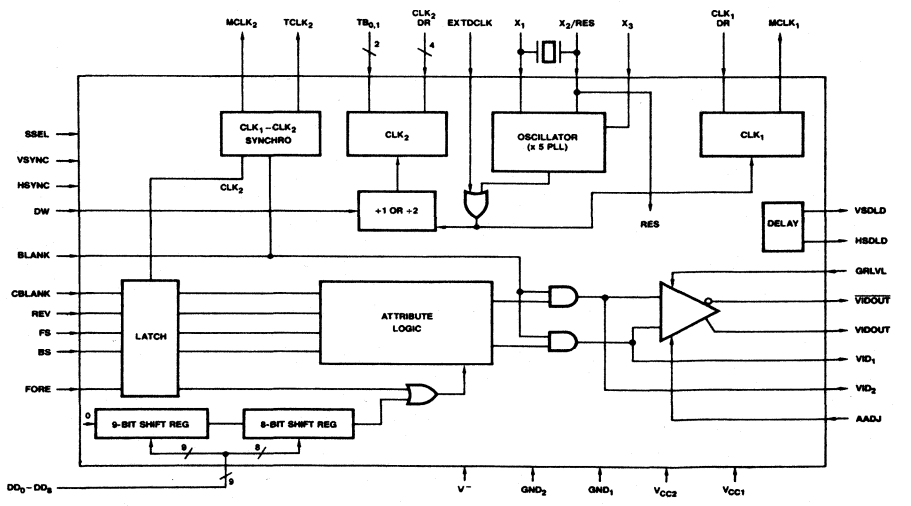
The VSC consists of a parallel-to-serial converter which provides a video bit stream to on-chip attribute logic. This logic, under control of the attribute inputs, operates on the bit stream to generate grey scale video. Video outputs from the VSC are of two forms — analog and digital. The digitally encoded outputs implement four video levels: Blank, Black, Grey and White. Identical information is available in analog form via differential outputs (current driven) into a nominal 75Ω impedance.

The Am8152A/53A also supports proportional spacing using a bit width programmable character clock. Character ROM pixel information is selectable from two to seventeen pixels per character. Up to three blank pixels can be appended to the character ROM input thereby facilitating right justification of text.

The difference between the Am8152A and the Am8153A is in the output scheme. The Am8152A has standard TTL outputs and operates in the 25 – 60MHz range, while the Am8153A has 10K ECL outputs and operates in the 40 – 100MHz range.

The Am8152A/53A is fabricated using AMD's advanced bipolar process with internal ECL logic. The device is available in conventional 48-pin dual in-line packages.

BLOCK DIAGRAM



BD001241

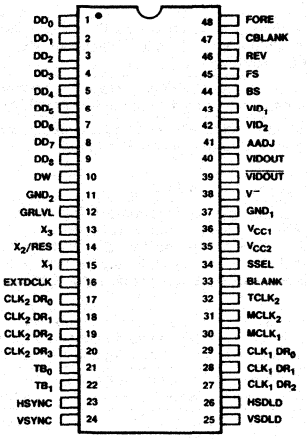
Figure 1.

03888C

Refer to page 7-1 for Essential Information on Military Devices

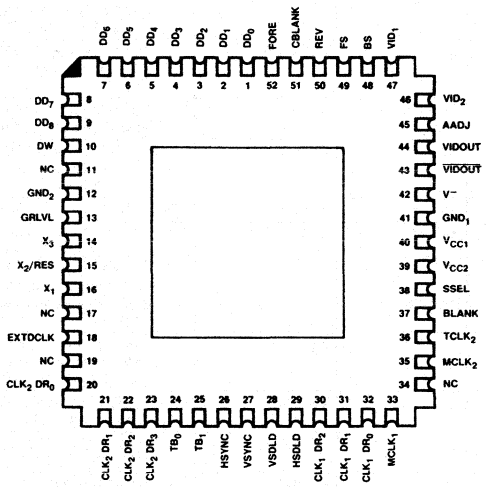
CONNECTION DIAGRAM
Top View

D-48-1



CD001511

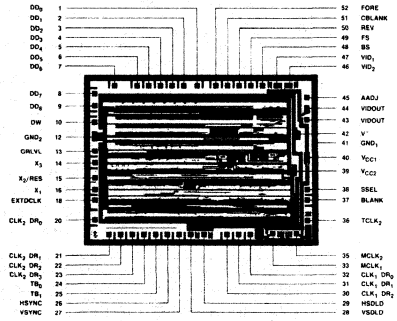
Leadless Chip Carrier
L-52-1



CD004922

Note: Pin 1 is marked for orientation

METALLIZATION AND PAD LAYOUT



ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am8152A/53A **D** **C** **B**

- **B** — Screening Option
  - B = Burn-in
  - Blank = Standard processing
- **C** — Temperature (See Operating Range)
  - C = Commercial (0°C to +70°C)
  - I = Industrial (−40°C to 85°C)
- **D** — Package
  - D = CERDIP 48-Pin
  - L = Leadless Chip
  - J = Plastic Leaded Chip Carrier
- **Am8152A/53A** — Device type

Valid Combinations	
Am8152A	DC
Am8153A	DC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

DIP Pin No.	LCC Pin No.	Name	I/O	Description
30	33	$MCLK_1$	O	$CLOCK_1$ (non-TTL compatible). $MCLK_1$ is a system clock. It is intended to drive the AM8052 horizontal and vertical timing circuitry as well as the DMA operations. $MCLK_1$ output is nominally a square wave divided down from the internal dot clock frequency according to the CLK_1 DR (CLK_1 Divide Ratio) input.
29-27	32-30	CLK_1 DR	I	CLK_1 DIVIDE RATIO. CLK_1 DR are three inputs which control the $MCLK_1$ divide ratio. The three inputs may be programmed to divide the $MCLK_1$ signal by two, four, six, ..., sixteen. For further information, see the table following this section.
31	35	$MCLK_2$	O	$CLOCK_2$ (non-TTL compatibe). $MCLK_2$ is a character display clock. Its function is to control the character code and attribute data output rate from the appropriate Am8052 CRTC's ports.
17-20	20-23	CLK_2DR	I	$CLOCK_2$ DIVIDE RATIO. CLK_2 DR are four inputs which control an internal divider to divide the dot clock frequency by a value from two to seventeen. For further information, see the table following this section.
32	36	$TCLK_2$	O	TTL CLK_2. $TCLK_2$ is a TTL compatible version of $MCLK_2$.
15, 14	16, 15	X_1,X_2/RES	I	X_1, X_2/RESET (X_2 is non-TTL compatible, reset is TTL compatible). X_1, X_2/RES are the external crystal inputs when the on-chip oscillator of the VSC is being used. The external crystal frequency is multiplied by five to produce the on-chip dot clock. If the external dot clock option is used, the X_1 should be tied Low and X_2/RES may be used as a reset input to synchronize multiple VSCs. Note that the reset signal should be synchronous to the external dot clock.
13	14	X_3	I	X_3 (non-TTL compatible). X_3 is used as an input to the on-chip voltage-controlled oscillator. When the on-chip oscillator of VSR is being used, X_3 should be connected to ground by an appropriate capacitor. If the external dot clock option is used, X_3 and X_1 should be tied Low.
24	27	VSYNC	I	VERTICAL SYNC. VSYNC is an input that must be synchronous to either $MCLK_1$ or $MCLK_2$, dependent on the SSEL input. If SSEL is High, VSYNC must be synchronous to $MCLK_1$.
25	28	VSDLD	O	VERTICAL SYNC DELAYED. VSDLD is the delayed output of VSYNC, synchronous to $MCLK_1$ or $MCLK_2$, depending on the setting of SSEL.
23	26	HSYNC	I	HORIZONTAL SYNC DELAYED. HSYNC is an input that must be synchronous to either $MCLK_1$ or $MCLK_2$, dependent upon the SSEL input. If SSEL is Low, HSYNC must be synchronous to $MCLK_2$; if SSEL is High, HSYNC must be synchronous to $MCLK_1$.
26	29	HSDLD	O	HORIZONTAL SYNC. HSDLD is the delayed output of HSYNC, synchronous to $MCLK_1$ or $MCLK_2$, depending upon the setting of SSEL.
34	38	SSEL	I	SYNC SELECT. The SSEL line determines if the VSYNC, HSYNC and BLANK are going to be synchronized to the $MCLK_1$ or $MCLK_2$ signals. A High on SSEL also will resynchronize CLK_2 and CLK_1 during blanking.
33	37	BLANK	I	BLANK. BLANK is an input normally synchronous to $MCLK_1$, although it may be synchronous to $MCLK_2$ in non-proportional spacing applications. The active pulse width of BLANK will usually overlap the inactive-to-active waveforms of HSYNC and VSYNC, as well as the active-to-inactive portion of VSYNC. While BLANK is active, $TCLK_2$/$MCLK_2$ may be forced to synchronize to the $MCLK_1$ clock. When BLANK goes inactive, thue rising edges of $MCLK_1$ and $TCLK_2$/$MCLK_2$ must be synchronized in order to prevent "dot walk" in proportional spacing applications. BLANK active also forces the video output level to "blank" regardless of DD, FORE or other inputs.
47	51	CBLANK	I	CHARACTER BLANK. CBLANK forces video output levels (VID_1, VID_2, VIDOUT and $\overline{VIDOUT}$ to switch to the background color level.
48	52	FORE	I	FOREGROUND VIDEO. The FORE video input is "OR'ed" with the dot data output by the parallel-to-serial shift register to switch to the foreground color level (e.g., to implement underlines).
46	50	REV	I	REVERSE. The REV input causes the foreground color levels to be transposed with the background color level for the total character period (including any tracking blanks).
45	49	FS	I	FOREGROUND SHIFT. The FS input causes the shift in the video output levels to produce a highlight effect. See Table 1.
21, 22	24, 25	TB_0, TB_1	I	TRAILING BLANKS. The TB inputs concatenate "blank" video dots to the tail end of the dot data contained in the parallel-to-serial shift register. TB can be specified to concatenate 0, 1, 2 or 3 dots. The TB value is also added to the CLK_2 DR value to obtain the total. The combination of all CLK_2DR inputs being High (17 dots) and both TB inputs being High (3 trailing blanks) is not allowed. The maximum CLK_2 period is 19 dot periods.
1-9	1-9	DD_0-DD_8	I	DOT DATA. The DD inputs accept parallel character dot matrix information for serial conversion for video output. DD data is accepted at the $TCLK_2$/$MCLK_2$ clock rate. DD_0 is shifted out first.
44	48	BS	I	BACKGROUND SELECT. The BS input specifies the color level of the background video. This input can be overridden by BLANK active.
40, 39	44, 43	VIDOUT, $\overline{VIDOUT}$	O	VIDEO OUTPUT (non-TTL compatible). VIDOUT and $\overline{VIDOUT}$ outputs in a differential mode the composite blank and video dot levels to a nominal 75Ω load impedance from switched current sources.
43, 42	47, 46	VID_1,VID_2	O	VIDEO DIGITAL (8152A-TTL; 8153A-ECL). VID_1 and VID_2 are digitally encoded outputs of the video out. VID_1 is the least significant bit. Encoding is as follows:

	VID_2 (VIDEO)	VID_1 (HIGHLIGHT)
Blank Level	0	0
Blank	0	1
Grey	1	0
White	1	1

PIN DESCRIPTION (Cont.)

DIP Pin No.	LCC Pin No.	Name	I/O	Description
12	13	GRLVL	I	GREY LEVEL. The GRLVL input adjusts the current level output, via the VIDOUT and $\overline{\text{VIDOUT}}$ outputs, of the grey video level. There are two pre-selected grey levels: for GRLVL (High), grey is brighter; for GRLVL (Low), grey is darker.
10	10	DW	I	DOUBLE WIDTH. The DW input, when active HIGH, causes the dot clock supplied to thue $TCLK_2/MCLK_2$ clock divide to be divided by two. This function is used to facilitate doubling the width of a character cell matrix in the horizontal direction. The trailing blank information is also widened during a double width character.
16	18	EXTDCLK	I	EXTERNAL DOT CLOCK (8152A-TTL; 8153A-ECL). EXTDCLK is an external, TTL or ECL compatible dot clock input for use in multiple Am8152A/53A configurations. This signal replaces the internal oscillator function. To enable EXTDCLK, both X_1 and X_3 must be grounded.
41	45	AADJ	I	ANALOG OUTPUTS CURRENT ADJUST (Non-TTL compatible). Analog output current adjust is used for setting the analog video output current to 13.3mA. This is done by connecting AADJ to V^- via an applicable 1% resistor.
36, 35	40, 39	V_{CC1}, V_{CC2}		V_{CC} should be connected to +5 V for both the 8152A and 8153A.
37, 11	41, 12	GND_1, GND_2		GND should be connected to Ground for both the 8152A and 8153A.
38	42	V^-		V^- should be connected to Ground for the 8152A or to −5.2 V for the 8153A.

CLK_1 DR				
2	1	0	A^1	B^1
L	L	L	1	1
L	L	H	2	2
L	H	L	3	3
L	H	H	4	4
H	L	L	5	5
H	L	H	6	6
H	H	L	7	7
H	H	H	8	8

CLK_2 DR					
3	2	1	0	C^1	D^1
L	L	L	L	1	1
L	L	L	H	1	2
L	L	H	L	2	2
L	L	H	H	2	3
L	H	L	L	3	3
L	H	L	H	3	4
L	H	H	L	4	4
L	H	H	H	4	5
H	L	L	L	5	5
H	L	L	H	5	6
H	L	H	L	6	6
H	L	H	H	6	7
H	H	L	L	7	7
H	H	L	H	7	8
H	H	H	L	8	8
H	H	H	H	8	9

Note 1. A, B, C, and D are measured in EXTDCLK periods.

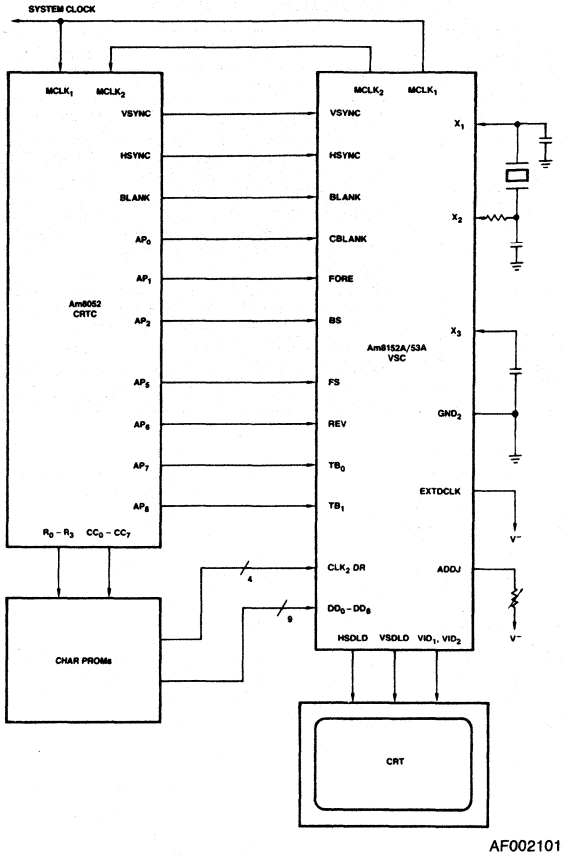


Figure 2. Am8152A/53A Application with Am8052 CRT Controller

DETAILED DESCRIPTION

The Am8152A Video System Controller (VSC) supports both black and white and color video applications for CPUs, CRT controllers, and terminals. The essential functions of the VSC are to support proportional and non-proportional character display, to synchronize and mix character attributes with video, and to output the video in a four level analog or digital format.

PARALLEL PIXEL LOADING

Pixel information that must be serialized for video transmission is loaded into the serial shift register via inputs $DD_0 - DD_8$. Information is loaded on both edges of the $MCLK_2$ character clock, as shown in Figure 3. The information set up on DD(0:7) prior to the falling edge of $MCLK_2$ is loaded into positions $VID_9 - VID_{16}$. Note that DD_8 information is ignored. Information set up on DD(0:8) prior to the rising edge of $MCLK_2$ is loaded into positions $VID_0 - VID_8$. Thus, up to 17 bits of pixel information can be loaded into the shift register. Note that if the character width is nine pixels or less the information captured on the falling edge of the $MCLK_2$ is not used. Any trailing blank insertion only occurs after the total number of pixels for the character have been transmitted.

CLK_2DR (0:3) and TB (0:1) determine the divide ratio for the character clock. The sum of both values specifies the character clock period in dot clocks. During the trailing blank, the VSC shifts out what was loaded into the shift register. Therefore, it is the responsibility of the user to insure that the pixels output during the trailing blank dot period are set to the blank level.

VIDEO OPERATION

Parallel video data is obtained from the character ROM inputs; bits are shifted out serially and mixed with attribute information such as underline, shifted underline, and any other video sources. Video is internally encoded into one of four levels: White, Grey, Black and Blank. White is the highest analog current level, and Blank is the lowest. This information is then output through two ports. One port provides a single current source output into a 75Ω impedance, and the second port outputs either encoded TTL or ECL video on two pins.

There are two distinct blank inputs to the Am8152A/53A. BLANK is the CRTC's horizontal and vertical retrace period input which causes a blank output level to the display. CBLANK is an attribute input to selectively blank a character cell by forcing the video information for the particular character cell period to switch to the selected background color level.

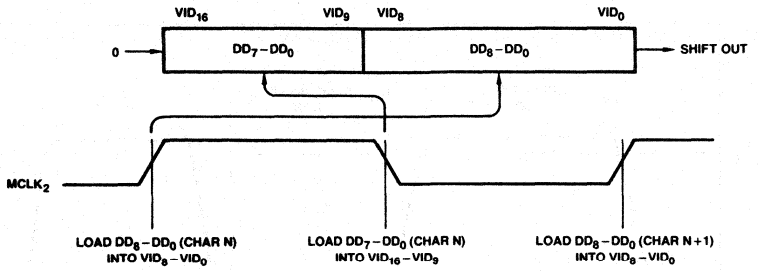


Figure 3. Shift Register Loading

CRYSTAL SPECIFICATION

The crystal used with the VSC may have the following specifications:

Series Resonant
Shunt Capacitance: 7 pf maximum

VIDEO INPUTS/OUTPUTS

Video information may be input in a number of different ways. Table 1 depicts all the combinations of video outputs achievable with each of the various inputs. The background color is determined by a separate pin input allowing either a black or white background. Using the REVERSE VIDEO (REV) input, a grey background can also be selected. The foreground then becomes black or white according to the signal on the foreground SHIFT line. Foreground and video sums can be modified depending on the combination of background, foreground shift, and reverse inputs. The user may apply any of his video inputs to the foreground to obtain a desired effect.

TABLE 1. Am8152A/53A VIDEO ATTRIBUTES

INPUTS				Am8152A/53A VIDEO ATTRIBUTES
BS	FS	REV	$\overline{\text{CBLANK}}$ (DD (0 : 8) + FORE)	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

TB4

SYSTEM TIMING

The CPU clock ($MCLK_1$) output is derived from an on-board oscillator by an externally programmable divide by two or three prescaler and a one-to-eight decoder. The internal oscillator is capable of operating a frequency of up to 100MHz and in a fundamental or third harmonic mode. Figure 4 shows the output waveform of $MCLK_1$ and $MCLK_2$.

The character clock ($MCLK_2$) output to the CRTC is frequency modulated according to the chosen number of dots per character cell. The duty cycle of $MCLK_2$ is 50% (±1 dot clock period) and is derived from an internal crystal driven oscillator whose divide ratio is set by the width of the character ROM plus the number of trailing blanks. A double width input further modifies $MCLK_2$, doubling the character width. During an active BLANK input, $MCLK_2$ is internally resynchronized to $MCLK_1$. This action aligns character cells at the left-end side of the display, thereby eliminating "Dot Walk." The Vertical and Horizontal Sync (VSYNC, HSYNC) inputs from the CRT controller are buffered and delayed by a $MCLK_1$ or $MCLK_2$ clock period to phase correctly with the character video output.

PROPORTIONAL/VARIABLE SPACING

Proportional spacing is achieved by programming on a character-by-character basis, a number of two to twenty dot clock periods per character. The character ROM pixel information is selectable from two to seventeen per character. Up to three trailing blank pixels can be concatenated to the character ROM input, making it easier to provide a straight right margin for right justification of text.

COLOR APPLICATION

The Am8152A/53A may be used for many high-end color display applications. The foreground video and background information is mixed by the Am8152A/53A, and the encoded TTL video output can be used externally to select a color mix for the particular pixel being displayed. The horizontal and vertical synchronization and video blank are output by the Am8152A/53A.

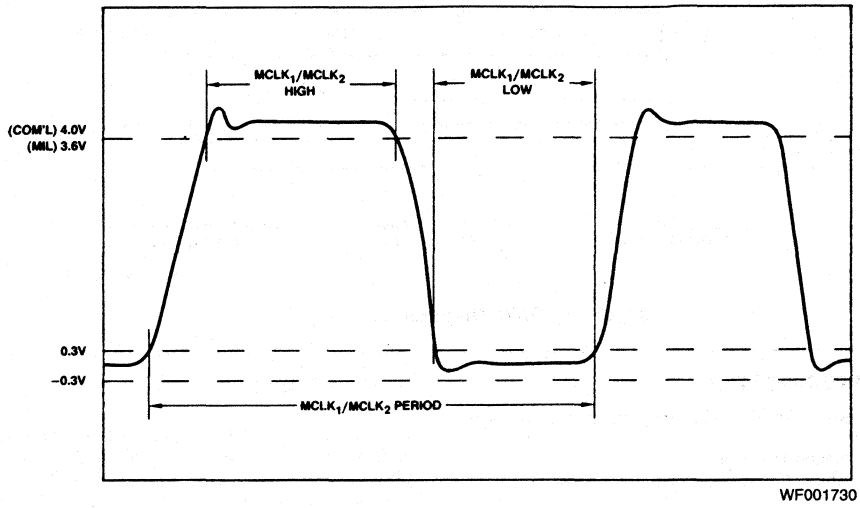
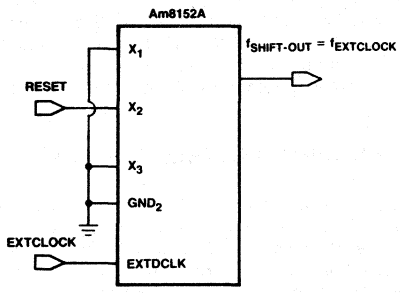
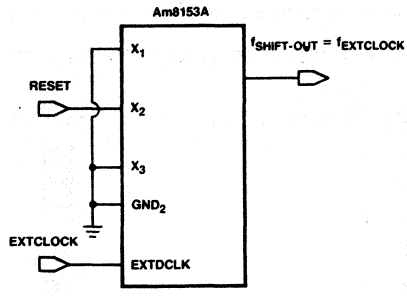


Figure 4. $MCLK_1/MCLK_2$ Output Waveform

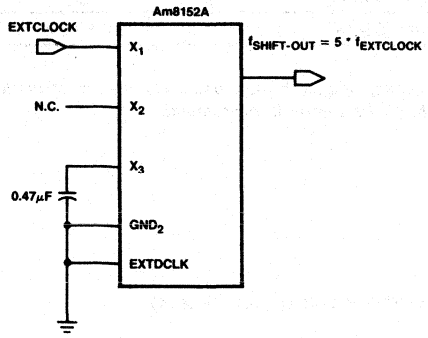
DOTCLOCK GENERATION MODE

EXTERNAL CLOCK FLOW THROUGH MODE

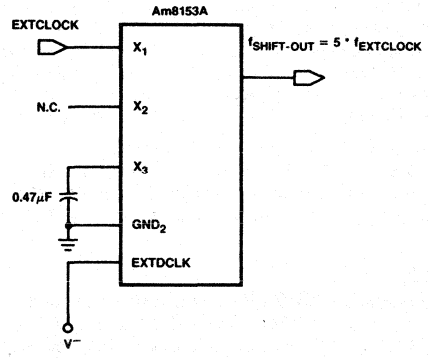




EXTERNAL CLOCK MULTIPLIER MODE

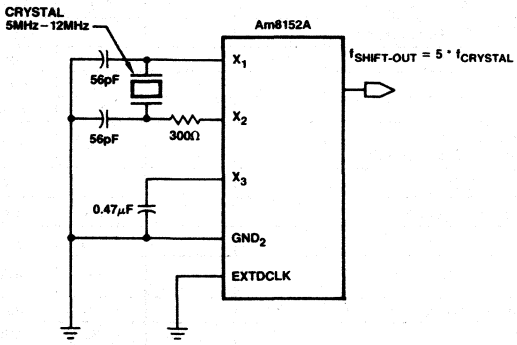


AF002151

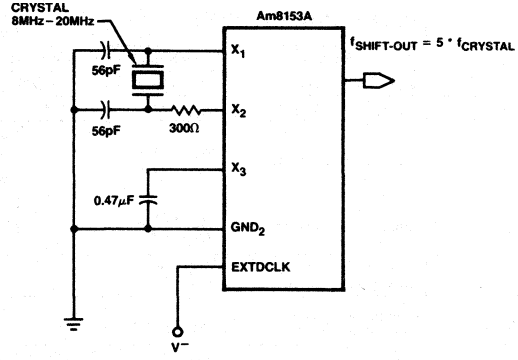


AF002121

CRYSTAL OSCILLATOR MULTIPLIER MODE



AF002161



AF002131

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage to Ground Potential
Continuous ..... −0.5 to +7.0V
DC Voltage Applied to Outputs for
High Output State ..... −0.5V to +V_{CC}
DC Input Voltage ..... −0.5 to +5.5V
DC Output Current into Outputs (See Note 5) 30mA
DC Input Current ..... −30 to +5.0mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	V_{SS}
Am8152ADC	0°C to 70°C	5.0V ±5%	N/A

Operating ranges define those limits over which the functionality of the device is guaranteed.

Am8152A
DC CHARACTERISTICS over operating range unless otherwise specified (See Note 4)
COM'L T_A = 0 to 70°C, V_{CC} = 5.0V ±5%, V^- = 0V

Parameters	Description	Test Conditions (Note 1)				Min	Typ (Note 2)	Max	Units
V_{OH}	Output HIGH Voltage	V_{CC} = Min	$MCLK_1$ $MCLK_2$	I_{OH} = 0.1mA	COM'L	4.0			Volts
			TTL Output	I_{OH} = −1mA I_{OH} = −2.6mA	COM'L	2.4	3.4		Volts
V_{OL}	Output LOW Voltage	V_{CC} = Min		I_{OL} = 0.1mA $MCLK_{1/2}$				0.3	Volts
				I_{OL} = 16mA TTL Outputs				0.5	Volts
V_{IH}	Input HIGH Level	Guaranteed Input HIGH Voltage				2.0			Volts
V_{IL}	Input LOW Level	Guaranteed Input LOW Voltage						0.8	Volts
V_I	Input Clamp Voltage	V_{CC} = Min, I_{IN} = −18mA						−1.2	Volts
I_{IL}	Input LOW Current	V_{CC} = Max V_{IN} = 0.4V		All Inputs (Except RES, EXTDCLK)				−0.4	mA
				RES, EXTDCLK				−1.0	mA
I_{IH}	Input HIGH Current	V_{CC} = Max V_{IN} = 2.7V		All Inputs (Except RES)				+50	µA
				RES				+600	µA
I_I	Input HIGH Current at Max Input Voltage	V_{CC} = Max V_{IN} = 5.5V						+1.0	mA
I_{SC}	Output Short Current Current (Note 3)	V_{CC} = Max		$MCLK_1$, $MCLK_2$		−50		−250	mA
				Others		−40		−130	mA
I_{CC}	Power Supply Current TTL	V_{CC1} = Max V_{CC2} = Max	Over Operating Range @ T_A = 70°C					415 375	mA mA

Notes: 1. For conditions shown as Min or Max use the appropriate value specified under Operating Ranges for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. Except : X_1, X_2, X_3, AADJ, VIDOUT, $\overline{\text{VIDOUT}}$.
5. The maximum rating for VIDOUT and $\overline{\text{VIDOUT}}$ is 15mA. Extended operation at current levels above 15mA will result in impairing the life of the device. Shorting VIDOUT, $\overline{\text{VIDOUT}}$, or AADJ to ground will destroy the device.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage to Ground Potential
Continuous ..... −0.5 to +7.0V
DC Voltage Applied to Outputs for
High Output State ..... −0.5V to +V_{CC}
DC Input Voltage ..... −0.5 to +5.5V
DC Output Current into Outputs (See Note 5) 30mA
DC Input Current ..... −30 to +5.0mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	V_{SS}
Am8153ADC	0°C to 70°C	5.0V ±5%	N/A

Operating ranges define those limits over which the functionality of the device is guaranteed.

Am8153A
DC CHARACTERSTICS over operating range unless otherwise specified (See Note 4)

COM'L T_A = 0 to 70°C, V_{CC} = 5.0V ±5%, V^- = −5.2V ±5%

Parameters	Description	Test Conditions (Note 1)				Min	Typ (Note 2)	Max	Units
V_{OH}	Output HIGH Voltage	V_{CC} = Min	$MCLK_1$ $MCLK_2$	I_{OH} = −0.1mA	COM'L	4.0			Volts
			TTL Outputs	I_{OH} = −1mA I_{OH} = −2.6mA	COM'L	2.4	3.4		Volts
V_{OL}	Output LOW Voltage	V_{CC} = Min		I_{OL} = 0.1mA $MCLK_{1/2}$				0.3	Volts
				I_{OL} = 16mA TTL Outputs				0.5	Volts
V_{IH}	Input HIGH Level	Guaranteed Input HIGH Voltage				2.0			Volts
V_{IL}	Input LOW Level	Guaranteed Input LOW Voltage						0.8	Volts
V_I	Input Clamp Voltage	V_{CC} = Min, I_{IN} = −18mA						−1.2	Volts
I_{IL}	Input LOW Current	V_{CC} = Max V_{IN} = 0.4V		All Inputs (Except RES)				−0.4	mA
				RES				−1.0	mA
I_{IH}	Input HIGH Current	V_{CC} = Max V_{IN} = 2.7V		All Inputs (Except RES)				+50	μA
				RES				+600	μA
I_I	Input HIGH Current at Max Input Voltage	V_{CC} = Max V_{IN} = 5.5V						+1.0	mA
I_{SC}	Output Short Current Current (Note 3)	V_{CC} = Max		$MCLK_1$, $MCLK_2$		−50		−250	mA
				Others		−40		−130	mA
I_{CC}	Power Supply Current ECL	V_{CC1} = Max V_{CC2} = Max	Over Operating Range					410	mA
			@ T_A = 70°C					370	mA
I^-	Power Supply Current ECL	V^- = Max	Over Operating Range					40	mA
			@ T_A = 70°C					37	mA

Notes: 1. For conditions shown as Min or Max use the appropriate value specified under Operating Ranges for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. Except: X_1, X_2, X_3, AADJ, VIDOUT, $\overline{\text{VIDOUT}}$, VID_1, VID_2, EXTDCLK.
5. The maximum rating for VIDOUT and $\overline{\text{VIDOUT}}$ is 15mA. Extended operation at current levels above 15mA will result in impairing the life of the device. Shortening VIDOUT, $\overline{\text{VIDOUT}}$, or AADJ to ground will destroy the device.

Am8152A
SWITCHING CHARACTERISTICS over operating range
(T_A = 0 to 70°C, V_{CC} = 5.0V ±5%, V^- = 0V)

Number	Description		Min	Max	Units
1	$MCLK_1$ Period		100		ns
2	CLK_2 Period		70		ns
3	$MCLK_1$ HIGH (See Note 5)	4.0V	38		ns
4	$MCLK_1$ LOW (See Notes 3 and 5)	0.3V	38		ns
5	$MCLK_2$ HIGH (See Note 6)	4.0V	23		ns
6	$MCLK_2$ LOW (See Notes 3 and 6)	0.3V	23		ns
7	Data to $MCLK_2$/$TCLK_2$ RE (See Note 1)		20		ns
8	$MCLK_2$/$TCLK_2$ to Data Not Valid		0		ns
9	VSYNC/HSYNC to $MCLK_1$ RE Setup (SSEL = HIGH)		20		ns
10	VSYNC/HSYNC to $MCLK_2$ RE Setup (SSEL = LOW)		20		ns
11	$TCLK_2$ RE to $MCLK_2$ RE Delay			8	ns
12	$TCLK_2$ FE to $MCLK_2$ FE Delay			12	ns
13	$MCLK_1$ to VSLD, HSLD (SSEL = HIGH)			$6 + T_D$	ns
14	$TCLK_2$ to VSLD, HSLD (SSEL = LOW)			$6 + T_D$	ns
15	DD(0:7) to $TCLK_2$ FE		20		ns
16	$TCLK_2$ RE to VID_1 VID_2 VAL (See Note 2)			$6 + T_D$	ns
17	BLANK FE to $MCLK_2$ RE Setup (SSEL = LOW)		20		ns
18	BLANK FE to $MCLK_1$ FE Setup (SSEL = HIGH)		20		ns
19	BLANK RE to $MCLK_1$ RE Setup (SSEL = HIGH)		20		ns
20	BLANK RE to $MCLK_2$ RE Setup (SSEL = LOW)		20		ns
22	VID_1 to VID_2 Skew		−5	+5	ns
24	EXTDCLK to $MCLK_1$			20	ns
25	EXTDCLK to $TCLK_2$			13	ns
26	EXTDCLK to $MCLK_2$			23	ns
27	EXTDCLK to VID_1/VID_2			13	ns
28	EXTDCLK to HSDLD/VSDLD (SSEL HI)			13	ns
29	EXTDCLK to HSDLD/VSDLD (SSEL LO)			13	ns
30	EXTDCLK to Data in Setup		9		ns
31	EXTDCLK to Data Not Valid Hold		11		ns
32	EXTDCLK to H/V SYNC Setup		10		ns
33	EXTDCLK Period		16.6		ns
34	EXTDCLK LOW Cycle		5		ns
35	EXTDCLK HIGH Cycle		5.0		ns
36	Reset Pulse Width (High)		10.0		ns
37	Reset Low to EXTDCLK Setup		8.0		ns

Notes: 1. Data includes CBLANK, FORE, REV, FS, DD_0–DD_8, TB0, TB1, BS, CLK_1DR, CLK_2DR, DW.
2. First Pixel of character. T_D is pixel period as defined by oscillator frequency.
3. Max undershoot on these outputs is guaranteed to be −0.3V.
4. T_D is the dot clock period.
5. Guaranteed to 100ns $MCLK_1$ cycle time.
6. Guaranteed to 70ns $MCLK_2$ cycle time (even divide ratio only).

Am8153A

SWITCHING CHARACTERISTICS over operating range

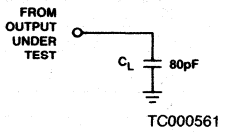
(T_A = 0 to 70°C, V_{CC} = 5.0V ±5%, V^- = −5.2V ±5%)

Number	Description		Min	Max	Units
1	$MCLK_1$ Period		100		ns
2	CLK_2 Period		70		ns
3	$MCLK_1$ HIGH (See Note 5)	4.0V	38		ns
4	$MCLK_1$ LOW (See Notes 3 and 5)	0.3V	38		ns
5	$MCLK_2$ HIGH (See Note 6)	4.0V	23		ns
6	$MCLK_2$ LOW (See Notes 3 and 6)	0.3V	23		ns
7	Data to $MCLK_2$/$TCLK_2$ RE (See Note 1)		$20 + T_D$ (See Note 4)		ns
8	$MCLK_2$/$TCLK_2$ to Data Not Valid		0		ns
9	VSYNC/HSYNC to $MCLK_1$ RE Setup (SSEL = HIGH)		$20 + T_D$		ns
10	VSYNC/HSYNC to $MCLK_2$ RE Setup (SSEL = LOW)		$20 + T_D$		ns
11	$TCLK_2$ RE to $MCLK_2$ RE Delay			8	ns
12	$TCLK_2$ FE to $MCLK_2$ FE Delay			12	ns
13	$MCLK_1$ to VSLD, HSLD (SSEL = HIGH)			6	ns
14	$TCLK_2$ to VSLD, HSLD (SSEL = LOW)			6	ns
15	DD(0:7) to $TCLK_2$ FE		$20 + T_D$		ns
16	$TCLK_2$ RE to VID_1 VID_2 VAL (See Note 2)			6	ns
17	BLANK FE to $MCLK_2$ RE Setup (SSEL = LOW)		$20 + T_D$		ns
18	BLANK FE to $MCLK_1$ FE Setup (SSEL = HIGH)		$20 + T_D$		ns
19	BLANK RE to $MCLK_1$ RE Setup (SSEL = HIGH)		$20 + T_D$		ns
20	BLANK RE to $MCLK_2$ RE Setup (SSEL = LOW)		$20 + T_D$		ns
21	VID_1 to VID_2 Skew		−2	+2	ns
24	EXTDCLK to $MCLK_1$			20	ns
25	EXTDCLK to $TCLK_2$			10	ns
26	EXTDCLK to $MCLK_2$			23	ns
27	EXTDCLK to VID_1/VID_2			8	ns
28	EXTDCLK to HSDLD/VSDLD (SSEL HI)			10	ns
29	EXTDCLK to HSDLD/VSDLD (SSEL LO)			10	ns
30	EXTDCLK to Data in Setup		9		ns
31	EXTDCLK to Data Not Valid Hold		11		ns
32	EXTDCLK to H/V SYNC Setup		10		ns
33	EXTDCLK Period		10		ns
34	EXTDCLK LOW Cycle		5		ns
35	EXTDCLK HIGH Cycle		3.5		ns
36	Reset Pulse Width (High)		10.0		ns
37	Reset Low to EXTDCLK Setup		8.0		ns

Notes: 1. Data includes CBLANK, FORE, REV, FS, DD_0–DD_8, TB0, TB1, BS, CLK_1DR, CLK_2DR, DW.
2. First Pixel of character. T_D is pixel period as defined by oscillator frequency.
3. Max undershoot on these outputs is guaranteed to be −0.3V.
4. T_D is the dot clock period.
5. Guaranteed to 100ns $MCLK_1$ cycle time.
6. Guaranteed to 70ns $MCLK_2$ cycle time (even divide ratio only).

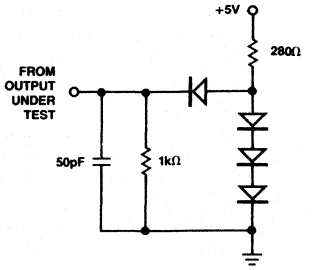
SWITCHING TEST CIRCUIT

$MCLK_1/MCLK_2$ OUTPUT



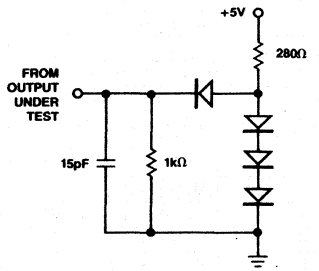
TC000561

TTL OUTPUTS EXCEPT $TCLK_2$



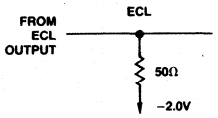
TC000701

$TCLK_2$ OUTPUT



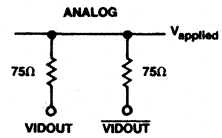
TC001531

ECL OUTPUTS



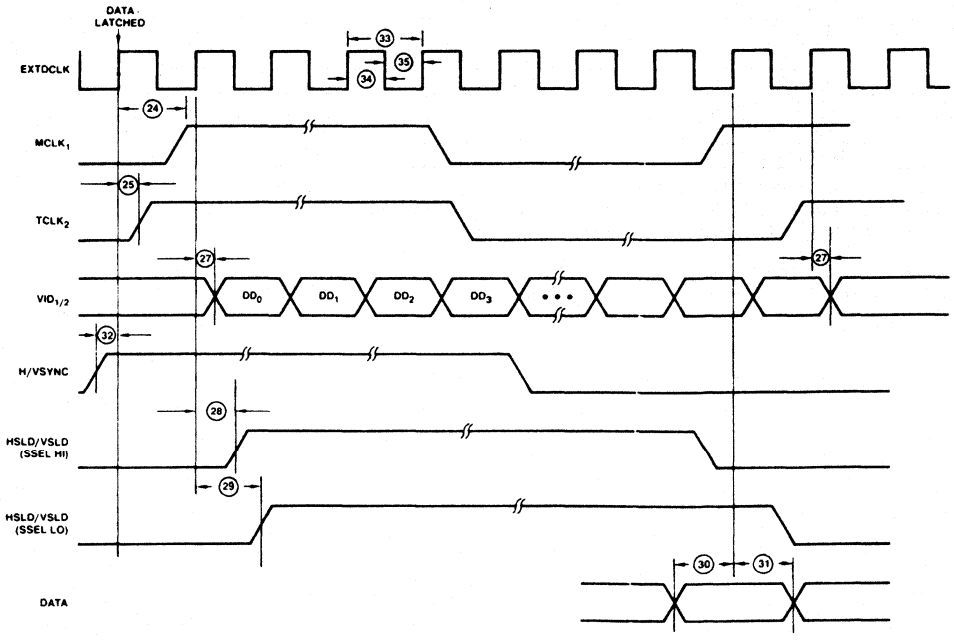
TC000571

ANALOG OUTPUTS



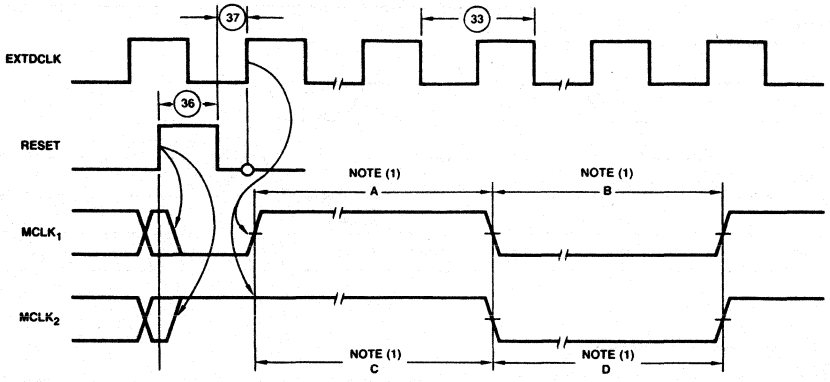
TC000551

Am8152A/53A TIMING
(PARAMETERS MEASURED WITH RESPECT TO EXTDCLK)



WF003212

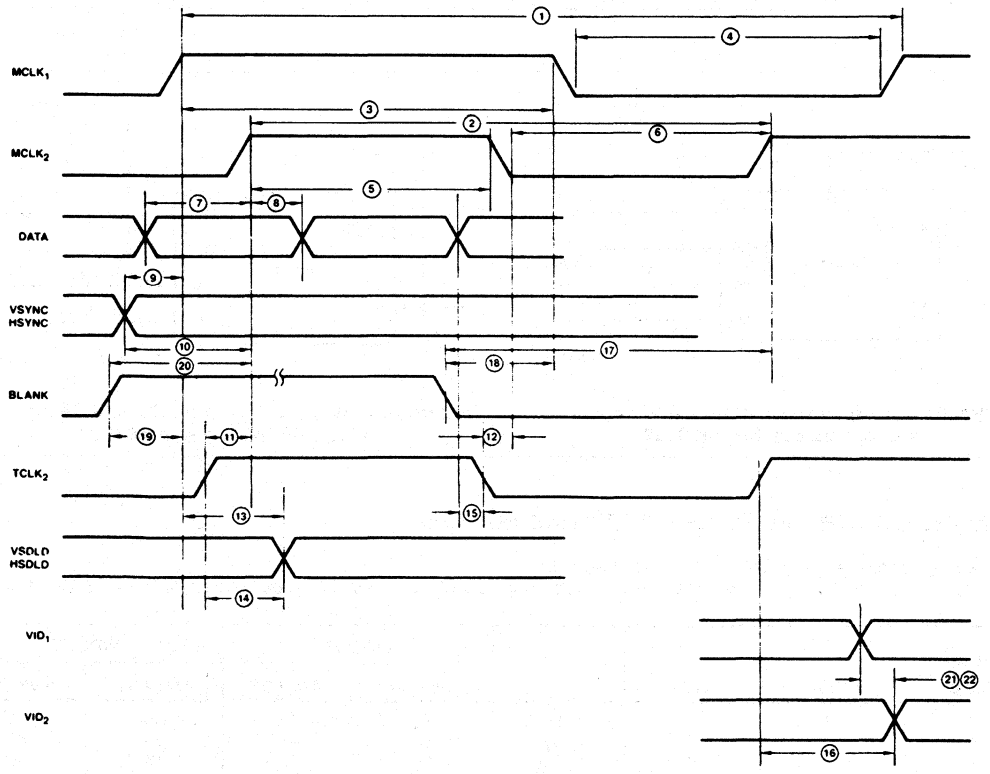
RESET TIMING FOR Am8152A/53A



WF003191

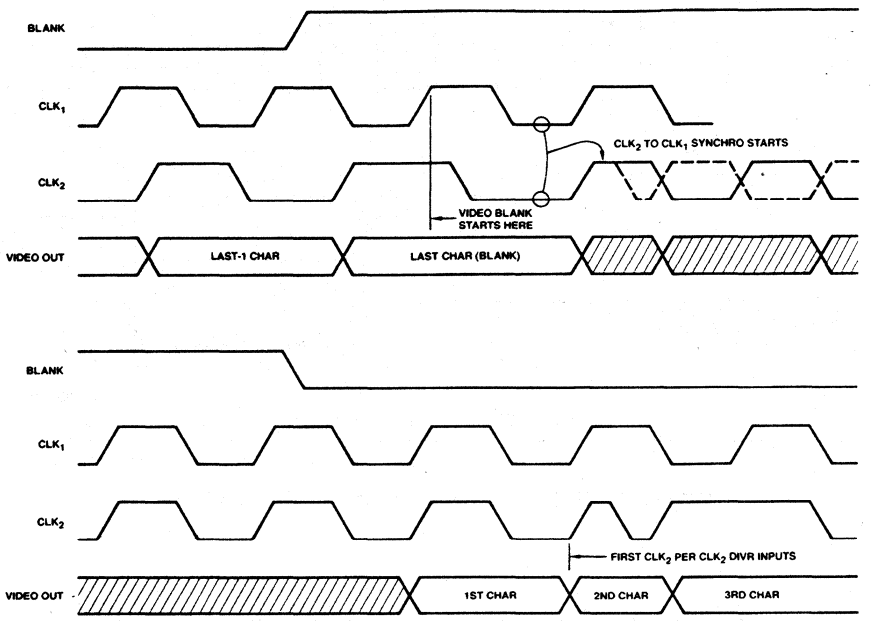
Note 1. See Pin Description section.

SWITCHING TIMING DIAGRAM — $MCLK_1/MCLK_2$

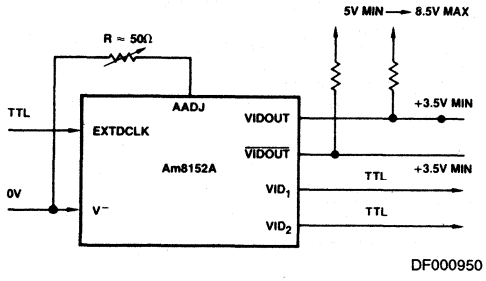


WF001761

VSC CLK_2 SYNCHRONIZATION (ONLY OCCURS IF SSEL IS HIGH)

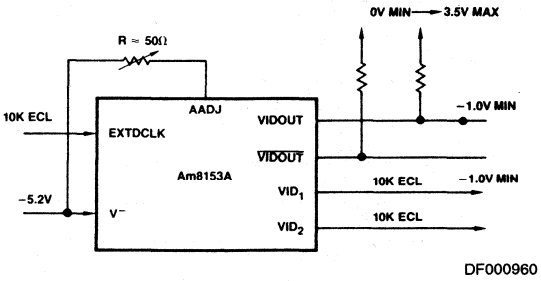


WF001751



DF000950

Figure 5. Analog Video Outputs and Digital Video Outputs for Am8152A



DF000960

Figure 6. Analog Video Outputs and Digital Video Outputs for Am8153A

ANALOG ELECTRICAL CHARACTERISTICS (see notes)

The following conditions apply unless otherwise specified:
COM'L $T_A = 0$ to $+70°C$ $V_{CC} = 5.0V \pm 5\%$ (Min = 4.75V Max = 5.25V) Am8152A: $V^- = 0V$ Am8153A: $V^- = -5.2V \pm 5\%$

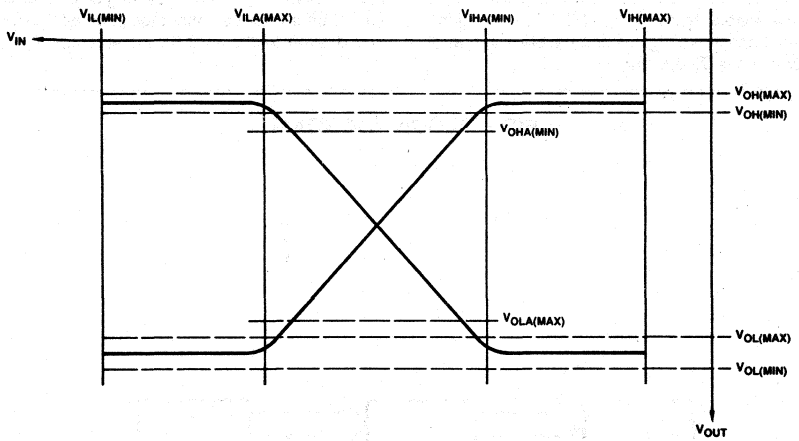
Grey Level	VID_2	VID_1		VIDOUT		$\overline{VIDOUT}$	
				Min (%)	Max (%)	Min (%)	Max (%)
X	I	I	I_{White}	0	0	100	100
I	I	O	I_{Grey1}	37	44	56	63
O	I	O	I_{Grey2}	45	53	47	55
X	O	I	I_{Black}	90.5	92.5	7.5	9.5
X	O	O	I_{Blank}	100	100	0	0

DRIFT OVER OPERATING CONDITIONS
(For particular part)

Grey Level	VID_2	VID_1		VIDOUT	$\overline{VIDOUT}$
X	I	I	I_{White}	0	0
I	I	O	I_{Grey1}	2% Max	2% Max
O	I	O	I_{Grey2}	2% Max	2% Max
X	O	I	I_{Black}	1% Max	1% Max
X	O	O	I_{Blank}	0	0

Notes:
1. Test Condition: Normal I_{White} for $\overline{VIDOUT}$ + 13.3mA.
2. Positive current flowing into VIDOUT/$\overline{VIDOUT}$.
3. t_R, t_F = 5ns Max.
4. $\overline{VIDOUT}$ output currents normalized to I_{White}. VIDOUT output currents normalized to I_{Blank}.
5. Min/Max values for VIDOUT and $\overline{VIDOUT}$ account for variation of different devices.
6. Am8152A
   V Pull-Up: 8.5V ≥ V Pull-Up ≥ V_{CC}
   VIDOUT/$\overline{VIDOUT}$: (V Pull-Up) ≥ VIDOUT/$\overline{VIDOUT}$ ≥ (V_{CC} – 1V)
7. Am8153A
   V Pull-Up: 3.5V ≥ V Pull-Up ≥ 0V
   VIDOUT/$\overline{VIDOUT}$: (V Pull-Up) ≥ VIDOUT/$\overline{VIDOUT}$ ≥ –1.0V

Am8153A 10K ECL SPECIFICATIONS



WF003201

DC CHARACTERISTICS

COM'L T_A = 0 to 70°C V_{CC} = 5.0V ±5% GND = 0V V^- = –5.2V ±5% (Max = –5.46, Min = –4.94)

	Parameters	Test Conditions	0°C	25°C	70°C	Unit
10K ECL Outputs	V_{OH} (Max)	50Ω to –2V	–840	–780	–720	mV
	V_{OH} (Min)		–1000	–930	–900	mV
	V_{OHA} (Min)		–1020	–950	–920	mV
VID_1 and VID_2	V_{OLA} (Max)	50Ω to –2V	–1645	–1600	–1605	mV
	V_{OL} (Max)		–1665	–1620	–1625	mV
	V_{OL} (Min)		–1870	–1850	–1830	mV
10K ECL Input	V_{IH} (Max)		–840	–780	–720	mV
	V_{IHA} (Min)		–1145	–1105	–1045	mV
EXTDCLK	V_{ILA} (Max)		–1490	–1475	–1450	mV
	V_{IL} (Min)		–1870	–1850	–1830	mV
	I_{IH}	V^- = Max, V_{IN} = V_{IH} (Max)	200	200	200	µA
	I_{IL}	V^- = Max, V_{IN} = V_{IL} (Min)	150	150	150	µA

Z8060 FIFO

Buffer Unit and FIFO Expander

DISTINCTIVE CHARACTERISTICS

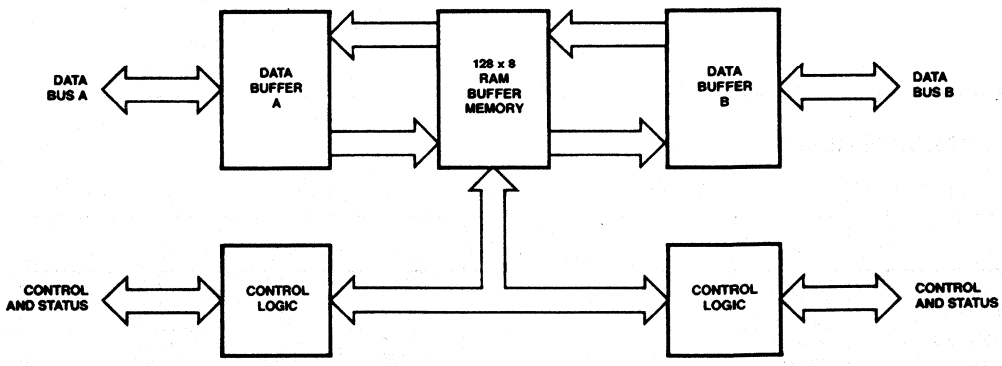
- Bidirectional, asynchronous data transfer capability
- Large 128-bit-by-8-bit buffer memory
- Two-wire, interlocked handshake protocol
- 3-state data outputs
- Wire-ORing of empty and full outputs for sensing of multiple-unit buffers
- Connects any number of FIFOs in series to form buffer of any desired length
- Connects any number of FIFOs in parallel to form buffer of any desired width

GENERAL DESCRIPTION

The Z8060* First-In, First-Out (FIFO) buffer unit consists of a 128-bit-by-8-bit memory, bidirectional data transfer and handshake logic. The structure of the FIFO unit is similar to that of other available buffer units. FIFO is a general-purpose unit; its handshake logic is compatible with that of other members of the Z8000 family.

FIFOs can be cascaded end-to-end without limit to form a parallel 8-bit buffer of any desired length (in 128-byte increments). Any number of single- or multiple-unit FIFO serial buffers can be connected in parallel to form buffers of any desired width (in 8-bit increments).

FIFO BLOCK DIAGRAM



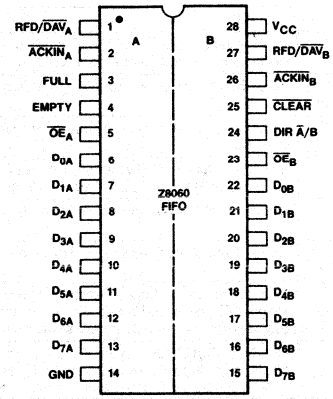
BD003311

*Z8000 is a trademark of Zilog, Inc.

02128B

Refer to page 7-1 for Essential Information on Military Devices

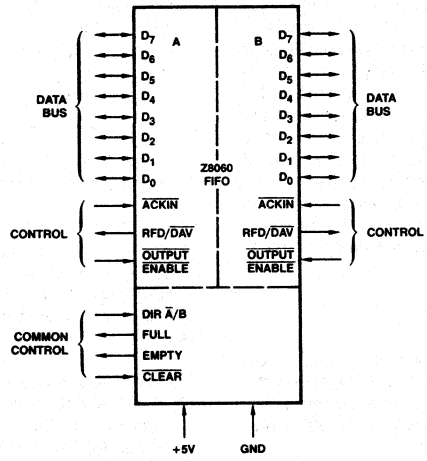
CONNECTION DIAGRAM
Top View
D-28, P-28



CD005131

Note: Pin 1 is marked for orientation

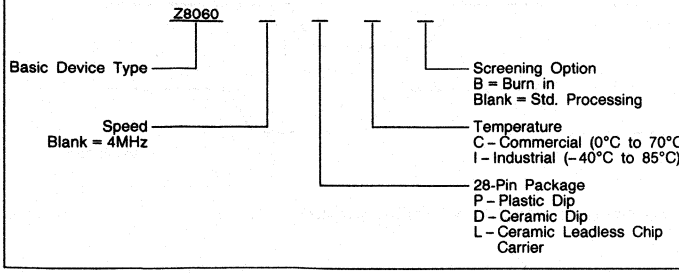
LOGIC SYMBOL



LS001171

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Z8060	DC, DCB DI, DIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}		+5V Power Supply.
14	GND		Ground.
2, 26	$\overline{ACKIN}$	I active LOW	Acknowledge Input. This line signals the FIFO that output data has been received by peripherals or that input data is valid.
25	$\overline{CLEAR}$	I active LOW	Clear Buffer. When set to LOW, this line causes all data to be cleared from the FIFO buffer.
6 – 13	D_0-D_7	I/O	Data Bus (bidirectional). These bidirectional lines are used by the FIFO to receive and to transmit data.
24	DIR $\overline{A}$/B	I	Direction Input $\overline{A}$/B (two control states). A HIGH on this line signals that input data is to be received at port B. A LOW on this line signals that input data is to be received at port A.
4	EMPTY	O active HIGH	Buffer Status (open-drain). A HIGH on this line indicates that the FIFO buffer is empty.
3	FULL	O active HIGH	Buffer Status (open-drain). A HIGH on this line indicates that the FIFO buffer is full.
5, 23	$\overline{OE}_A$, $\overline{OE}_B$	I active LOW	Output Enable A, Output Enable B. When LOW, $\overline{OE}_A$ enables the bus drivers for port A; when HIGH, $\overline{OE}_A$ causes the bus drivers to float to a high-impedance level. Input $\overline{OE}_B$ controls the bus drivers for port B in the same manner as $\overline{OE}_A$ controls those for port A.
1, 27	RFD/$\overline{DAV}$	O	Ready-for-Data/Data Available (outputs RFD, active HIGH, $\overline{DAV}$ active LOW). RFD, when HIGH, signals to the peripherals involved that the FIFO is ready to receive data. $\overline{DAV}$, when LOW, signals to the peripherals involved that FIFO has data available to send.

DETAILED DESCRIPTION

Interlocked 2-Wire Handshake

In interlocked 2-wire handshake operation, the action of FIFO must be acknowledged by the other half of the handshake before the next action can occur. In an Output Handshake mode, the FIFO indicates that new data is available only after the external device has indicated that it is ready for the data. In an Input Handshake mode, the FIFO does not indicate that it is ready for new data until the data source indicates that the previous byte of the data is no longer available, thereby acknowledging the acceptance of the last byte. This control feature allows the FIFO, with no external logic, to directly interface with the port of any CPU in the Z8 Family — a CIO, a UPC, an FIO, or another FIFO. The timing for the input and output handshake operations is shown in Figures 1 and 2, respectively.

Resetting or Clearing the FIFO

The $\overline{CLEAR}$ is used to initialize and clear the FIFO. A Low level on this input clears all data from the FIFO, allows the EMPTY output to go HIGH and forces both outputs RFD/$\overline{DAV}_A$ and RFD/$\overline{DAV}_B$ HIGH. A HIGH level on $\overline{CLEAR}$ allows the data to transfer through the FIFO.

Bidirectional Transfer Control

The FIFO has bidirectional data transfer capability under control of the DIR $\overline{A}$/B input. When DIR $\overline{A}$/B is set LOW, port A becomes input handshake and port B becomes output handshake; data transfers are then made from port A to port B. Setting DIR $\overline{A}$/B HIGH reverses the handshake assignments and the direction of transfer. This bidirectional control is illustrated in Table 1.

TABLE 1. BIDIRECTIONAL CONTROL FUNCTION TABLE

DIR $\overline{A}$/B	Port A Handshake	Port B Handshake	Transfer
0	Input	Output	A to B
1	Output	Input	B to A

The FIFO buffer must be empty before the direction of transfer is changed; otherwise, the results of the change will be unpredictable. If FIFO status is unknown when a transfer direction change is to be made, the recommended procedure is:

(1) Force and hold $\overline{CLEAR}$ LOW,
(2) Set DIR $\overline{A}$/B to the level required for the desired direction, and then
(3) Force $\overline{CLEAR}$ HIGH.

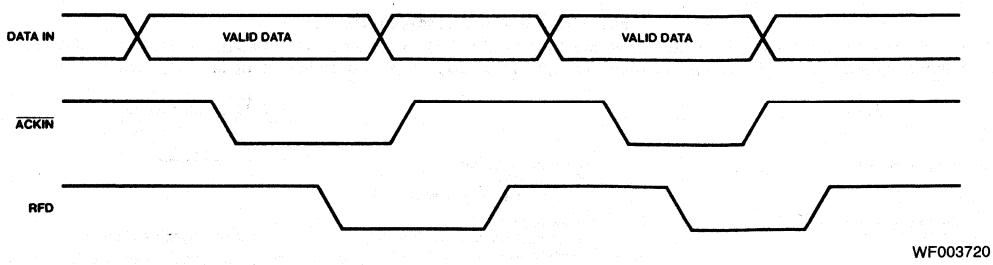


Figure 1. Two-Wire Interlocked Handshake Timing (Input)

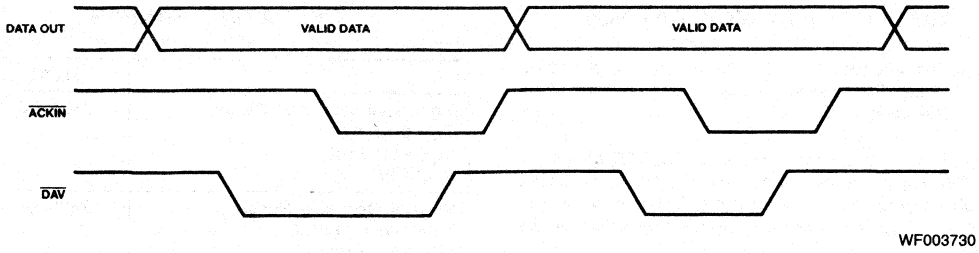


Figure 2. Two-Wire Interlocked Handshake Timing (Output)

EMPTY and FULL Operation

The EMPTY and FULL output lines can be wire-ORed with the EMPTY and FULL lines of other FIFOs and FIOs. This capability enables the user to determine the EMPTY/FULL status of a buffer consisting of multiple FIFOs, FIOs, or a combination of both. Table 2 shows the various states of EMPTY and FULL.

TABLE 2. SIGNALS EMPTY AND FULL OPERATION TABLE

Number of Bytes in FIFO	Empty	Full
0	HIGH	LOW
1-127	LOW	LOW
128	LOW	HIGH

Interconnection Example

A simplified block diagram showing the manner in which FIFOs can be interconnected to extend a FIO buffer is shown in Figure 3.

Output Enable Operation

The FIFO provides a separate Output Enable ($\overline{OE}$) signal for each port of the buffer. An $\overline{OE}$ output is valid only when its port is in the Output Handshake mode. The control of this output function is shown in Table 3. Signal $\overline{OE}$ operates with lines DIR $\overline{A}$/B. A HIGH on a valid $\overline{OE}$ line 3-states its port's data bus but does not affect the handshake operation. A LOW level on a valid $\overline{OE}$ enables the data bus outputs if its port is in the Output Handshake mode. Note that the handshake operation is unaffected by the Output Enable pin.

TABLE 3. OUTPUT CONTROL FUNCTION TABLE

DIR $\overline{A}$/B	$\overline{OE}_A$	$\overline{OE}_B$	Function
0	X	0	Disable Port A Output Enable Port B Output
0	X	1	Disable Port A Output Disable Port B Output
1	0	X	Enable Port A Output Disable Port B Output
1	1	X	Disable Port A Output Disable Port B Output

Note: X = Don't Care.

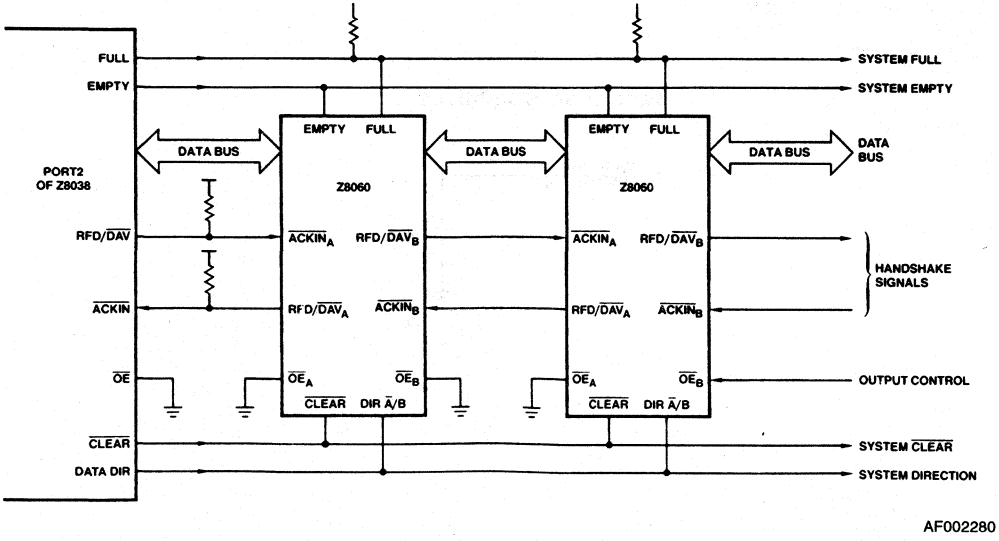


Figure 3. Typical Interconnection (Simplified Diagram)

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage to any Pin Relative to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.8W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	Z8060
Commercial Operating Range T_A = 0 to +70°C V_{CC} = 5V ±5% V_{SS} = 0V	Z8060DC
Industrial Operating Range T_A = −40 to +85°C V_{CC} = 5V ±10% V_{SS} = 0V	Z8060DI

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (Note 1)

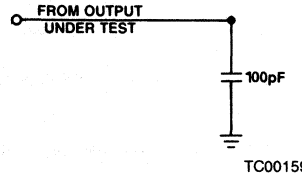
Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input LOW Voltage		−0.3		+0.8	Volts
V_{IH}	Input HIGH Voltage	Standard Temp	2.0		V_{CC} +0.3	Volts
V_{OL}	Output LOW Voltage	I_{OL} = 3.2mA			0.5	Volts
		I_{OL} = 2.0mA			0.4	
V_{OH}	Output HIGH Voltage	I_{OH} = −250µA	2.4			Volts
I_{OZL}	Output Leakage Current	V_{OUT} = 0.4V			10	µA
I_{OZH}	Output Leakage Current	V_{OUT} = V_{CC}			10	µA
I_I	Input Leakage Current				±10	µA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.			10	pF
$C_{I/O}$	I/O Capacitance				20	pF
C_{OUT}	Output Capacitance				15	pF
I_{CC}	Power Supply Current	V_{CC} = MAX			250	mA

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

+4.75V ≤ V_{CC} ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ +70°C

Test Load Conditions



TC001591

FIFO 2-WIRE HANDSHAKE TIMING

Timing for 2-wire interlock handshake operation is shown in Figure 1. The symbol, description and values for the numbered parameters (Figure 1) are given in Switching Characteristics.

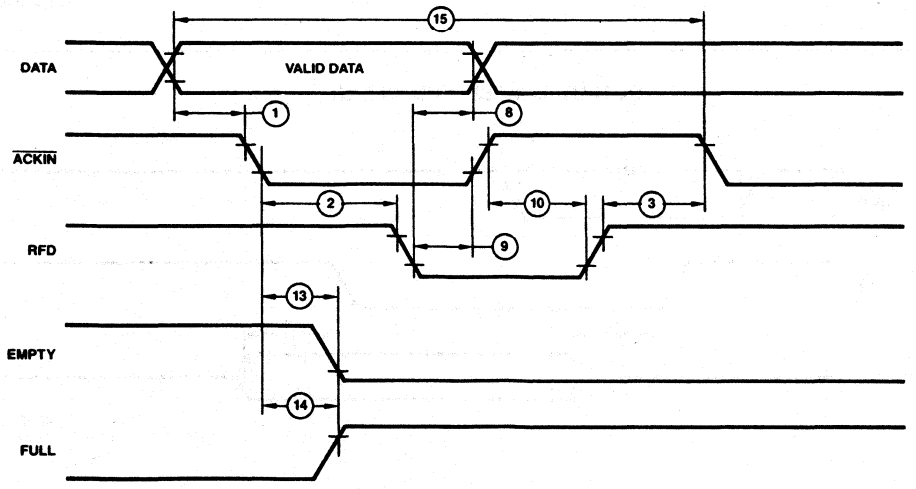
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Parameters	Description	Min	Max	Units
1	TsDI(ACK)	Data Input to $\overline{ACKIN}\downarrow$ to Setup Time	50		ns
2	TdACKf(RFD)	$\overline{ACKIN}\downarrow$ to RFD↓ Delay	0	500	ns
3	TdRFDr(ACK)	RFD↑ to $\overline{ACKIN}\downarrow$ Delay	0		ns
4	TsDO(DAV)	Data Out to $\overline{DAV}\downarrow$ Setup Time	25		ns
5	TdDAVf(ACK)	DAV↓ to $\overline{ACKIN}\downarrow$ Delay	0		ns
6	ThDO(ACK)	Data Out to $\overline{ACKIN}\uparrow$ Hold Time	50		ns
7	TdACK(DAV)	$\overline{ACKIN}\downarrow$ to $\overline{DAV}\uparrow$ Delay	0	500	ns
8	ThDI(RFD)	Data input to RFD ↓ Hold Time	0		ns
9	TdRFDf(ACK)	RFD↓ to $\overline{ACKIN}\uparrow$ Delay	0		ns
10	TdACKr(RFD)	$\overline{ACKIN}$ ↑ to RFD ↑ Delay	0	400	ns
11	TdDAVr(ACK)	DAV ↑ to $\overline{ACKIN}$ ↑	0		ns
12	TdACKr(DAV)	$\overline{ACKIN}$ ↑ to $\overline{DAV}$ ↓	0	800	ns
13	TdACKINf(EMPTY)	(Input) $\overline{ACKIN}$ ↓ to EMPTY ↓ Delay		600	ns
		(Output) $\overline{ACKIN}$ ↓ to EMPTY ↑ Delay			
14	TdACKINf(FULL)	(Input) $\overline{ACKIN}$ ↓ to FULL ↑ Delay		600	ns
		(Output) $\overline{ACKIN}$ ↓ to FULL ↓ Delay			
15	ACKIN Clock Rate	(Input)		1.0	MHz
16	TdACKINf($\overline{DAV}$f)	(Bubble Time)		800	ns
17	TwCLR	Width of Clear to Reset FIFO	700		ns
18	TdOE(DO)	$\overline{OE}$ ↓ to Data Bus Driven	0	150	ns
19	TdOE(DRZ)	$\overline{OE}$ ↑ to Data Bus Float		100	ns

Note: All timing references assume 2.0V for a logic 1 and 0.8V for a logic 0.

SWITCHING WAVEFORMS

INPUT TIMING

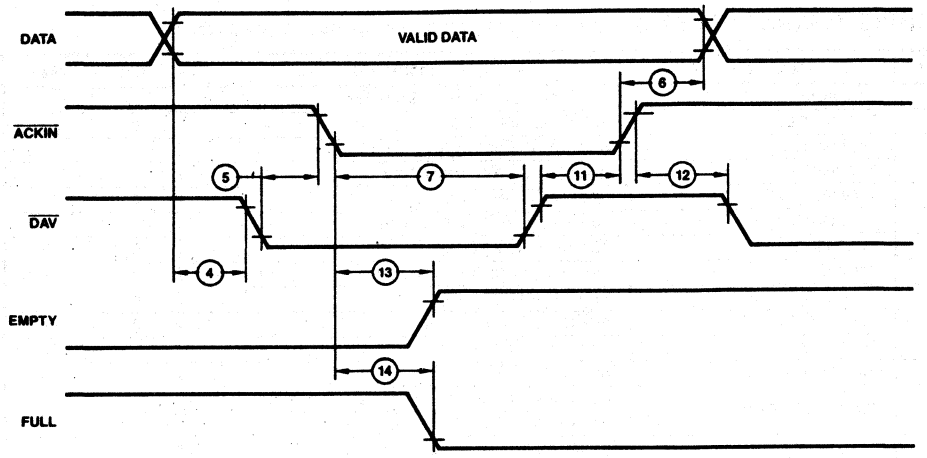


WF003740

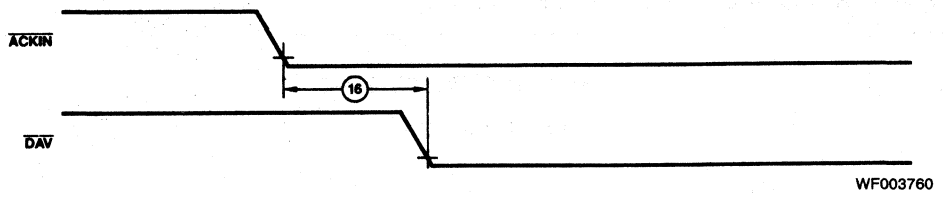
Figure 4. Timing Diagrams

SWITCHING WAVEFORMS (Cont.)

OUTPUT TIMING



ACKNOWLEDGE INPUT DATA TO DATA AVAILABLE TIME (BUBBLE TIME)



OUTPUT ENABLE AND CLEAR

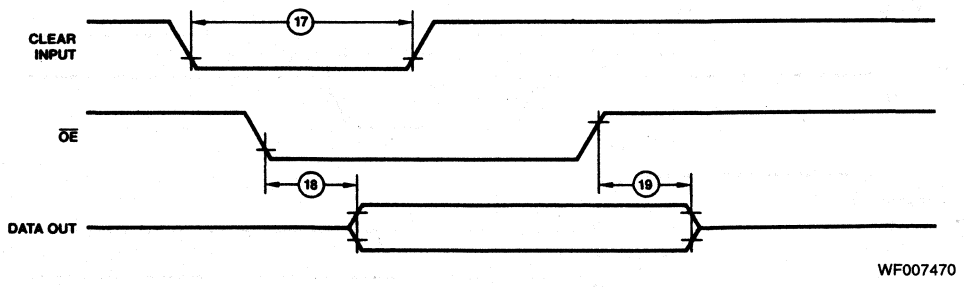


Figure 4. Timing Diagrams (Cont.)

Am9511A

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- 2, 3 and 4MHz operation; fixed point 16-bit and 32-bit operations
- Floating point 32-bit operations; binary data formats
- Add, Subtract, Multiply and Divide; trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation; float to fixed and fixed to float conversions
- Stack-oriented operand storage; DMA or programmed I/O data transfers
- End signal simplifies concurrent processing; Synchronous/Asynchronous operations
- General purpose 8-bit data bus interface; standard 24-pin package
- +12 volt and +5 volt power supplies; advanced N-channel silicon gate MOS technology

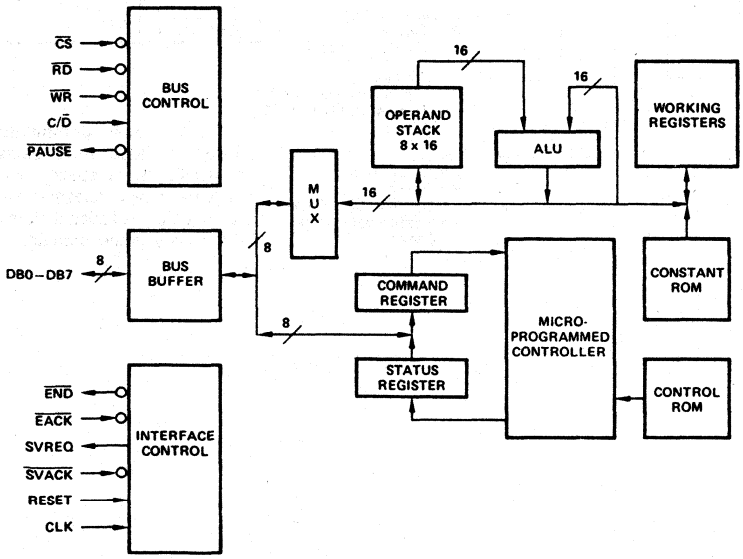
GENERAL DESCRIPTION

The Am9511A Arithmetic Processing Unit (APU) is a monolithic MOS/LSI device that provides high performance fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. It may be used to enhance the computational capability of a wide variety of processor-oriented systems.

All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack, and a command is issued to perform operations on the data in the stack. Results are then available to be retrieved from the stack, or additional commands may be entered.

Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.

BLOCK DIAGRAM

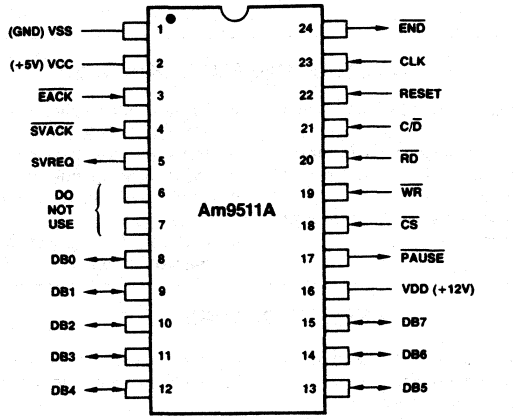


BD003340

01892B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-24-2



CD005170

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am9511A – –1 D C B

- Device Type
- Speed: Blank = 2MHz; –1 = 3MHz; –4 = 4MHz
- Package: D = Cerdip 24-pin; P = Plastic; L = Leadless Chip Carrier; J = Plastic Leaded Chip Carrier
- Temperature: C = Commercial 0°C to 70°C; I = Industrial –40°C to 85°C
- Screening Option: B = Burn In; Blank = Std. Processing

Valid Combinations	
Am9511A	DC, DCB, DI, DIB,
Am9511A-1	LC, LCB, LI, LIB,
Am9511A-4	/BJA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
2	V_{CC}		+5V Power Supply.
16	V_{DD}		+12V Power Supply.
1	V_{SS}		Ground.
23	CLK	I	(Clock). An external timing source connected to the CLK input provides the necessary clocking. The CLK input can be asynchronous to the $\overline{RD}$ and $\overline{WR}$ control signals.
22	RESET	I	(Reset). A HIGH on this input causes initialization. Reset terminates any operation in progress and clears the status register to zero. The internal stack pointer is initialized, and the contents of the stack may be affected, but the command register is not affected by the reset operation. After a reset the $\overline{END}$ output will be HIGH, and the SVREQ output will be LOW. For proper initialization, the RESET input must be HIGH for at least five CLK periods following stable power supply voltages and stable clock.
21	C/$\overline{D}$	I	(Command/Data Select). The C/$\overline{D}$ input, with the $\overline{RD}$ and $\overline{WR}$ inputs, determines the type of transfer to be performed on the data bus as follows:

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Function
L	H	L	Push data byte into the stack
L	L	H	Pop data byte from the stack
H	H	L	Enter command byte from the data bus
H	L	H	Read Status
X	L	L	Undefined

L = LOW
H = HIGH
X = DON'T CARE

24	$\overline{END}$	O	(End of Execution). A LOW on this output indicates that execution of the current command is complete. This output will be cleared HIGH by activating the $\overline{EACK}$ input LOW or performing any read or write operation or device initialization using the RESET. If $\overline{EACK}$ is tied LOW, the $\overline{END}$ output will be a pulse (see $\overline{EACK}$ description). This is an open drain output and requires a pull up to +5V. Reading the status register while a command execution is in progress is allowed. However, any read or write operation clears the flip-flop that generates the $\overline{END}$ output. Thus, such continuous reading could conflict with internal logic setting the $\overline{END}$ flip-flop at the completion of command execution.
3	$\overline{EACK}$	I	(End Acknowledge). This input when LOW makes the $\overline{END}$ output go HIGH. As mentioned earlier, LOW on the $\overline{END}$ output signals completion of a command execution. The $\overline{END}$ output signal is derived from an internal flip-flop which is clocked at the completion of a command. This flip-flop is clocked to the reset state when $\overline{EACK}$ is LOW. Consequently, if the $\overline{EACK}$ is tied LOW, the $\overline{END}$ output will be a pulse that is approximately one CLK period wide.
5	SVREQ	O	(Service Request). A HIGH on this output indicates completion of a command. In this sense, this output is same as the $\overline{END}$ output. However, whether the SVREQ output will go HIGH at the completion of a command or not is determined by a service request bit in the command register. This bit must be 1 for SVREQ to go HIGH. The SVREQ can be cleared (i.e., go LOW) by activating the $\overline{SVACK}$ input LOW or initializing the device using the RESET. Also, the SVREQ will be automatically cleared after completion of any command that has the service request bit as 0.
4	$\overline{SVACK}$	I	(Service Acknowledge). A LOW on this input activates the reset input of the flip-flop generating the SVREQ output. If the $\overline{SVACK}$ input is permanently tied LOW, it will conflict with the internal setting of the flip-flop to generate the SVREQ output. Thus the SVREQ indication cannot be relied upon if the $\overline{SVACK}$ is tied LOW.
8-15	DB0-DB7	I/O	(Bidirectional Data Bus). These eight bidirectional lines are used to transfer command, status and operand information between the device and the host processor. DB0 is the least significant, and DB7 is the most significant bit position. HIGH on the data bus line corresponds to 1, and LOW corresponds to 0. When pushing operands on the stack using the data bus, the least significant byte must be pushed first and most significant byte last. When popping the stack to read the result of an operation, the most significant byte will be available on the data bus first, and the least significant byte will be the last. Moreover, for pushing operands and popping results, the number of transactions must be equal to the proper number of bytes appropriate for the chosen format. Otherwise, the internal byte pointer will not be aligned properly. The Am9511A single precision format requires 2 bytes; double precision and floating-point formats require 4 bytes.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
18	$\overline{CS}$	I	(Chip Select). This input must be LOW to accomplish any read or write operation to the Am9511A. To perform a write operation, data is presented on DB0 through DB7 lines, C/$\overline{D}$ is driven to an appropriate level and the $\overline{CS}$ input is made LOW. However, actual writing into the Am9511A cannot start until $\overline{WR}$ is made LOW. After initiating the write operation by a $\overline{WR}$ HIGH-to-LOW transition, the $\overline{PAUSE}$ output will go LOW momentarily (TPPWW). The $\overline{WR}$ input can go HIGH after $\overline{PAUSE}$ goes HIGH. The data lines, C/$\overline{D}$ input and the $\overline{CS}$ input can change when appropriate hold time requirements are satisfied. See Write Timing diagram for details. To perform a read operation, an appropriate logic level is established on the C/$\overline{D}$ input, and $\overline{CS}$ is made LOW. The read operation does not start until the $\overline{RD}$ input goes LOW. $\overline{PAUSE}$ will go LOW for a period of TPPWR. When $\overline{PAUSE}$ goes back HIGH again, it indicates that the read operation is complete, and the required information is available on the DB0 through DB7 lines. This information will remain on the data lines as long as the $\overline{RD}$ input is LOW. The $\overline{RD}$ input can return HIGH anytime after $\overline{PAUSE}$ goes HIGH. The $\overline{CS}$ input and C/$\overline{D}$ inputs can change anytime after $\overline{RD}$ returns HIGH. See Read Timing diagram for details.
20	$\overline{RD}$	I	(Read). A LOW on this input is used to read information from an internal location and gate that information on to the data bus. The $\overline{CS}$ input must be LOW to accomplish the read operation. The C/$\overline{D}$ input determines what internal location is of interest. See C/$\overline{D}$, $\overline{CS}$ input descriptions and Read Timing diagram for details. If the END output was LOW, performing any read operation will make the $\overline{END}$ output go HIGH after the HIGH-to-LOW transition of the $\overline{RD}$ input (assuming $\overline{CS}$ is LOW).
19	$\overline{WR}$	I	(Write). A LOW on this input is used to transfer information from the data bus into an internal location. The $\overline{CS}$ must be LOW to accomplish the write operation. The C/$\overline{D}$ determines which internal location is to be written. See C/$\overline{D}$, $\overline{CS}$ input descriptions and Write Timing diagram for details. If the $\overline{END}$ output was LOW, performing any write operation will make the $\overline{END}$ output go HIGH after the LOW-to-HIGH transition of the $\overline{WR}$ input (assuming $\overline{CS}$ is LOW).
17	$\overline{PAUSE}$	O	(Pause). This output is a handshake signal used while performing read or write transactions with the Am9511A. A LOW at this output indicates that the Am9511A has not yet completed its information transfer with the host over the data bus. During a read operation, after $\overline{CS}$ went LOW, the $\overline{PAUSE}$ will become LOW shortly (TRP) after $\overline{RD}$ goes LOW. $\overline{PAUSE}$ will return high only after the data bus contains valid output data. The $\overline{CS}$ and $\overline{RD}$ should remain LOW when $\overline{PAUSE}$ is LOW. The RD may go high anytime after $\overline{PAUSE}$ goes HIGH. During a write operation, after $\overline{CS}$ went LOW, the $\overline{PAUSE}$ will be LOW for a very short duration (TPPWN) after $\overline{WR}$ goes LOW. Since the minimum of TPPWW is 0, the $\overline{PAUSE}$ may not go LOW at all for fast devices. $\overline{WR}$ may go HIGH anytime after $\overline{PAUSE}$ goes HIGH.

DETAILED DESCRIPTION

Major functional units of the Am9511A are shown in the block diagram. The Am9511A employs a microprogram controlled stack oriented architecture with 16-bit wide data paths.

The Arithmetic Logic Unit (ALU) receives one of its operands from the Operand Stack. This stack is an 8-word by 16-bit 2-port memory with last in-first out (LIFO) attributes. The second operand to the ALU is supplied by the internal 16-bit bus. In addition to supplying the second operand, this bidirectional bus also carries the results from the output of the ALU when required. Writing into the Operand Stack takes place from this internal 16-bit bus when required. Also connected to this bus are the Constant ROM and Working Registers. The ROM provides the required constants to perform the mathematical operations (Chebyshev Algorithms), while the Working Registers provide storage for the intermediate values during command execution.

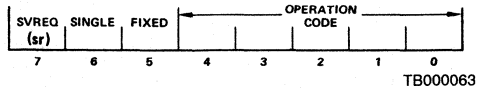
Communication between the external world and the Am9511A takes place on eight bidirectional input/output lines DB0 through DB7 (Data Bus). These signals are gated to the internal eight-bit bus through appropriate interface and buffer circuitry. Multiplexing facilities exist for bidirectional communication between the internal eight- and sixteen-bit buses. The Status Register and Command Register are also accessible via the eight-bit bus.

The Am9511A operations are controlled by the microprogram contained in the Control ROM. The Program Counter supplies the microprogram addresses and can be partially loaded from the Command Register. Associated with the Program Counter is the Subroutine Stack where return addresses are held during subroutine calls in the microprogram. The Microinstruction Register holds the current microinstruction being executed. This register facilitates pipelined microprogram execution. The Instruction Decode logic generates various internal control signals needed for the Am9511A operation.

The Interface Control logic receives several external inputs and provides handshake related outputs to facilitate interfacing the Am9511A to microprocessors.

Command Format

Each command entered into the Am9511A consists of a single 8-bit byte having the format illustrated below:



TB000063

Bits 0 – 4 select the operation to be performed as shown in the table. Bits 5 – 6 select the data format for the operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of the data to be operated on by fixed point commands (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are indicated; if bit 6 is a 0, double-precision (32-bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1, the service request output (SVREQ) will go HIGH at the conclusion of the command and will remain HIGH until reset by a LOW level on the service acknowledge pin ($\overline{SVACK}$) or until completion of execution of a succeeding command where bit 7 is 0. Each command issued to the Am9511A requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, SVREQ remains LOW.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description
7	6	5	4	3	2	1	0		
FIXED-POINT 16-BIT									
sr	1	1	0	1	1	0	0	SADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	0	1	SSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	1	0	SMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	1	1	1	0	1	1	0	SMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	1	1	0	1	1	1	1	SDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FIXED-POINT 32-BIT									
sr	0	1	0	1	1	0	0	DADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	0	1	DSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	1	0	DMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	0	1	1	0	1	1	0	DMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	0	1	0	1	1	1	1	DDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FLOATING-POINT 32-BIT									
sr	0	0	1	0	0	0	0	FADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	0	1	FSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	0	FMUL	Multiply NOS by TOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	1	FDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
DERIVED FLOATING-POINT FUNCTIONS									
sr	0	0	0	0	0	0	1	SQRT	Square Root of TOS. Result in TOS.
sr	0	0	0	0	0	1	0	SIN	Sine of TOS. Result in TOS.
sr	0	0	0	0	0	1	1	COS	Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	0	0	TAN	Tangent of TOS. Result in TOS.
sr	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS. Result in TOS.
sr	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS. Result in TOS.
sr	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS. Result in TOS.
sr	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS. Result in TOS.
sr	0	0	0	1	0	1	0	EXP	Exponential (e^x) of TOS. Result in TOS.
sr	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS. Result in NOS. Pop Stack.
DATA MANIPULATION COMMANDS									
sr	0	0	0	0	0	0	0	NOP	No Operation.
sr	0	0	1	1	1	1	1	FIXS	Convert TOS from floating-point to 16-bit fixed-point format.
sr	0	0	1	1	1	1	0	FIXD	Convert TOS from floating-point to 32-bit fixed-point format.
sr	0	0	1	1	1	0	1	FLTS	Convert TOS from 16-bit fixed-point to floating-point format.
sr	0	0	1	1	1	0	0	FLTD	Convert TOS from 32-bit fixed-point to floating-point format.
sr	1	1	1	0	1	0	0	CHSS	Change sign of 16-bit fixed-point operand on TOS.
sr	0	1	1	0	1	0	0	CHSD	Change sign of 32-bit fixed-point operand on TOS.
sr	0	0	1	0	1	0	1	CHSF	Change sign of floating-point operand on TOS.
sr	1	1	1	0	1	1	1	PTOS	Push 16-bit fixed-point operand on TOS to NOS. (Copy.)
sr	0	1	1	0	1	1	1	PTOD	Push 32-bit fixed-point operand on TOS to NOS. (Copy.)
sr	0	0	1	0	1	1	1	PTOF	Push floating-point operand on TOS to NOS. (Copy.)
sr	1	1	1	1	0	0	0	POPS	Pop 16-bit fixed-point operand from TOS. NOS becomes TOS.
sr	0	1	1	1	0	0	0	POPD	Pop 32-bit fixed-point operand from TOS. NOS becomes TOS.
sr	0	0	1	1	0	0	0	POPF	Pop floating-point operand from TOS. NOS becomes TOS.
sr	1	1	1	1	0	0	1	XCHS	Exchange 16-bit fixed-point operands TOS and NOS.
sr	0	1	1	1	0	0	1	XCHD	Exchange 32-bit fixed-point operands TOS and NOS.
sr	0	0	1	1	0	0	1	XCHF	Exchange floating-point operands TOS and NOS.
sr	0	0	1	1	0	1	0	PUPI	Push floating-point constant "π" onto TOS. Previous TOS becomes NOS.

Notes:
1. TOS means Top of Stack. NOS means Next on Stack.
2. AMD Application Brief "Algorithm Details for the Am9511A APU" provides detailed descriptions of each command function, including data ranges, accuracies, stack configurations, etc.
3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived functions may destroy several stack locations. See Application Brief for details.
4. The trigonometric functions handle angles in radians, not degrees.
5. No remainder is available for the fixed-point divide functions.
6. Results will be undefined for any combination of command coding bits not specified in this table.

Command Initiation

After properly positioning the required operands on the stack, a command may be issued. The procedure for initiating a command execution is as follows:

1. Enter the appropriate command on the DB0 – DB7 lines.
2. Establish HIGH on the C/$\overline{D}$ input.
3. Establish LOW on the $\overline{CS}$ input.
4. Establish LOW on the $\overline{WR}$ input after an appropriate set-up time (see Timing diagrams).
5. Sometime after the HIGH-to-LOW level transition of $\overline{WR}$ input, the $\overline{PAUSE}$ output will become LOW. After a delay of TPPWW, it will go HIGH to acknowledge the write operation. The WR input can return to HIGH anytime after $\overline{PAUSE}$ going HIGH. The DB0 – DB7, C/$\overline{D}$ and $\overline{CS}$ inputs are allowed to change after the hold time requirements are satisfied (see Timing diagram).

An attempt to issue a new command while the current command execution is in progress is allowed. Under these circumstances, the $\overline{PAUSE}$ output will not go HIGH until the current command execution is completed.

Operand Entry

The Am9511A commands operate on the operands located at the TOS and NOS, and results are returned to the stack at NOS and then popped to TOS. The operands required for the Am9511A are one of three formats – single precision fixed-point (2 bytes), double precision fixed-point (4 bytes) or floating-point (4 bytes). The result of an operation has the

same format as the operands except for float to fix or fix to float commands.

Operands are always entered into the stack least significant byte first and most significant byte last. The following procedure must be followed to enter operands onto the stack:

1. The lower significant operand byte is established on the DB0 – DB7 lines.
2. A LOW is established on the C/$\overline{D}$ input to specify that data is to be entered into the stack.
3. The $\overline{CS}$ input is made LOW.
4. After appropriate set-up time (see Timing diagrams), the $\overline{WR}$ input is made LOW. The $\overline{PAUSE}$ output will become LOW.
5. Sometime after this event, the $\overline{PAUSE}$ will return HIGH to indicate that the write operation has been acknowledged.
6. Anytime after the $\overline{PAUSE}$ output goes HIGH, the $\overline{WR}$ input can be made HIGH. The DB0 – DB7, C/$\overline{D}$ and $\overline{CS}$ inputs can change after appropriate hold time requirements are satisfied (see Timing diagrams).

The above procedure must be repeated until all bytes of the operand are pushed onto the stack. It should be noted that for single precision fixed-point operands, 2 bytes should be pushed and 4 bytes must be pushed for double precision fixed-point or floating-point. Not pushing all the bytes of a quantity will result in byte pointer misalignment.

The Am9511A stack can accommodate 8 single precision fixed-point quantities or 4 double precision fixed-point or floating-point quantities. Pushing more quantities than the capacity of the stack will result in loss of data which is usual with any LIFO stack.

Data Removal

Result from an operation will be available at the TOS. Results can be transferred from the stack to the data bus by reading the stack. When the stack is popped for results, the most significant byte is available first and the least significant byte last. A result is always of the same precision as the operands that produced it except for format conversion commands. Thus when the result is taken from the stack, the total number of bytes popped out should agree with the precision – single precision results are 2 bytes, and double precision and floating-point results are 4 bytes. The following procedure must be used for reading the result from the stack:

1. A LOW is established on the C/$\overline{D}$ input.
2. The $\overline{CS}$ input is made LOW.
3. After appropriate set-up time (see Timing diagrams), the $\overline{RD}$ input is made LOW. The $\overline{PAUSE}$ will become LOW.
4. Sometime after this, $\overline{PAUSE}$ will return HIGH, indicating that the data is available on the DB0 – DB7 lines. This data will remain on the DB0 – DB7 lines as long as the $\overline{RD}$ input remains LOW.
5. Anytime after $\overline{PAUSE}$ goes HIGH, the $\overline{RD}$ input can return HIGH to complete transaction.
6. The $\overline{CS}$ and C/$\overline{D}$ inputs can change after appropriate hold time requirements are satisfied (see Timing diagram).
7. Repeat this procedure until all bytes appropriate for the precision of the result are popped out.

Reading of the stack does not alter its data; it only adjusts the byte pointer. If more data is popped than the capacity of the stack, the internal byte pointer will wrap around and older data will be read again, consistent with the LIFO stack.

Status Read

The Am9511A status register can be read without any regard to whether a command is in progress or not. The only implication that has to be considered is the effect this might have on the END output discussed in the signal descriptions.

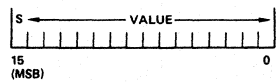
The following procedure must be followed to accomplish status register reading:

1. Establish HIGH on the C/$\overline{D}$ input.
2. Establish LOW on the $\overline{CS}$ input.
3. After appropriate set-up time (see Timing diagram) $\overline{RD}$ input is made LOW. The $\overline{PAUSE}$ will become LOW.
4. Sometime after the HIGH-to-LOW transition of $\overline{RD}$ input, the $\overline{PAUSE}$ will become HIGH, indicating that status register contents are available on the DB0 – DB7 lines. The status data will remain on DB0 – DB7 as long as $\overline{RD}$ input is LOW.
5. The $\overline{RD}$ input can be returned HIGH anytime after $\overline{PAUSE}$ goes HIGH.
6. The C/$\overline{D}$ input and $\overline{CS}$ input can change after satisfying appropriate hold time requirements (see Timing diagram).

Data Formats

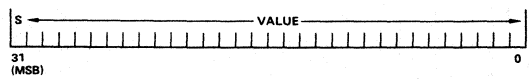
The Am9511A Arithmetic Processing Unit handles operands in both fixed-point and floating-point formats. Fixed-point operands may be represented in either single (16-bit operands) or double precision (32-bit operands), and are always represented as binary, two's complement values.

16-BIT FIXED-POINT FORMAT



TB00064M

32-BIT FIXED-POINT FORMAT



TB00065M

The sign (positive or negative) of the operand is located in the most significant bit (MSB). Positive values are represented by a sign bit of zero (S = 0). Negative values are represented by the two's complement of the corresponding positive value with a sign bit equal to 1 (S = 1). The range of values that may be accommodated by each of these formats is −32,767 to +32,767 for single precision and −2,147,483,647 to +2,147,483,647 for double precision.

Floating point binary values are represented in a format that permits arithmetic to be performed in a fashion analogous to operations with decimal values expressed in scientific notation.

$$(5.83 \times 10^2)(8.16 \times 10^1) = (4.75728 \times 10^4)$$

In the decimal system, data may be expressed as values between 0 and 10 times 10 raised to a power that effectively shifts the implied decimal point right or left the number of places necessary to express the result in conventional form (e.g., 47,572.8). The value-portion of the data is called the mantissa. The exponent may be either negative or positive.

The concept of floating point notation has both a gain and a loss associated with it. The gain is the ability to represent the significant digits of data with values spanning a large dynamic range limited only by the capacity of the exponent field. For

example, in decimal notation if the exponent field is two digits wide and the mantissa is five digits, a range of values (positive or negative) from 1.000×10^{-99} to $9.9999 \times 10^{+99}$ can be accommodated. The loss is that only the significant digits of the value can be represented. Thus there is no distinction in this representation between the values 123451 and 123452, for example, since each would be expressed as: 1.2345×10^{5}. The sixth digit has been discarded. In most applications where the dynamic range of values to be represented is large, the loss of significance, and hence accuracy of results, is a minor consideration. For greater precision a fixed point format could be chosen, although with a loss of potential dynamic range.

The Am9511 is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between .5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows:

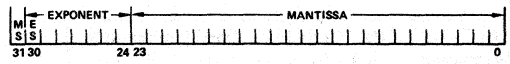
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100.5 expressed in this form is 0.11001001×2^{7}. The decimal equivalent of this value may be computed by summing the components (powers of two) of the mantissa and then multiplying by the exponent as shown below:

$$\begin{aligned}\text{value} &= (2^{-1} + 2^{-2} + 2^{-5} + 2^{-8}) \times 2^{7} \\ &= (0.5 + 0.25 + 0.03125 + 0.00290625) \times 128 \\ &= 0.78515625 \times 128 \\ &= 100.5\end{aligned}$$

Floating Point Format

The format for floating-point values in the Am9511A is given below. The mantissa is expressed as a 24-bit (fractional) value; the exponent is expressed as an unbiased two's complement 7-bit value having a range of −64 to +63. The most significant bit is the sign of the mantissa (0 = positive, 1 = negative), for a total of 32 bits. The binary point is assumed to be to the left of the most significant mantissa bit (bit 23). All floating-point data values must be normalized. Bit 23 must be equal to 1, except for the value zero, which is represented by all zeros.



TB000066

The range of values that can be represented in this format is $\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero.

Status Register

The Am9511A contains an eight bit status register with the following bit assignments:

BUSY	SIGN	ZERO	ERROR CODE	CARRY
7	6	5	4 3 2 1	0

BUSY: Indicates that Am9511A is currently executing a command (1 = Busy).

SIGN: Indicates that the value on the top of stack is negative (1 = Negative).

ZERO: Indicates that the value on the top of stack is zero (1 = Value is zero).

ERROR
CODE: This field contains an indication of the validity of the result of the last operation. The error codes are:
0000 – No error
1000 – Divide by error
0100 – Square root or log of negative number
1100 – Argument of inverse sine, cosine, or e^{x} too large
XX10 – Underflow
XX01 – Overflow

CARRY: Previous operation resulted in carry or borrow from most significant bit (1 = Carry/Borrow, 0 = No Carry/No Borrow).

If the BUSY bit in the status register is a one, the other status bits are not defined; if zero, indicating not busy, the operation is complete, and the other status bits are defined as given above.

Table 1.

Command Mnemonic	Hex Code (sr = 1)	Hex Code (sr = 0)	Execution Cycles	Summary Description
16-BIT FIXED-POINT OPERATIONS				
SADD	EC	6C	16-18	Add TOS to NOS. Result to NOS. Pop Stack.
SSUB	ED	6D	30-32	Subtract TOS from NOS. Result to NOS. Pop Stack.
SMUL	EE	6E	84-94	Multiply NOS by TOS. Lower result to NOS. Pop Stack.
SMUU	F6	76	80-98	Multiply NOS by TOS. Upper result to NOS. Pop Stack.
SDIV	EF	6F	84-94	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FIXED-POINT OPERATIONS				
DADD	AC	2C	20-22	Add TOS to NOS. Result to NOS. Pop Stack.
DSUB	AD	2D	38-40	Subtract TOS from NOS. Result to NOS. Pop Stack.
DMUL	AE	2E	194-210	Multiply NOS by TOS. Lower result to NOS. Pop Stack.
DMUU	B6	36	182-218	Multiply NOS by TOS. Upper result to NOS. Pop Stack.
DDIV	AF	2F	196-210	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FLOATING-POINT PRIMARY OPERATIONS				
FADD	90	10	54-368	Add TOS to NOS. Result to NOS. Pop Stack.
FSUB	91	11	70-370	Subtract TOS from NOS. Result to NOS. Pop Stack.
FMUL	92	12	146-168	Multiply NOS by TOS. Result to NOS. Pop Stack.
FDIV	93	13	154-184	Divide NOS by TOS. Result to NOS. Pop Stack.
32-BIT FLOATING-POINT DERIVED OPERATIONS				
SQRT	81	01	782-870	Square Root of TOS. Result to TOS.
SIN	82	02	3796-4808	Sine of TOS. Result to TOS.
COS	83	03	3840-4878	Cosine of TOS. Result to TOS.
TAN	84	04	4894-5886	Tangent of TOS. Result to TOS.
ASIN	85	05	6230-7938	Inverse Sine of TOS. Result to TOS.
ACOS	86	06	6304-8284	Increase Cosine of TOS. Result to TOS.
ATAN	87	07	4992-6536	Inverse Tangent of TOS. Result to TOS.
LOG	88	08	4474-7132	Common Logarithm of TOS. Result to TOS
LN	89	09	4296-6956	Natural Logarithm of TOS. Result to TOS.
EXP	8A	0A	3794-4878	e raised to power in TOS. Result to TOS.
PWR	8B	0B	8290-12032	NOS raised to power in TOS. Result to NOS. Pop Stack.
DATA AND STACK MANIPULATION OPERATIONS				
NOP	80	00	4	No Operation. Clear or set SVREQ.
FIXS	9F	1F	90-214	Convert TOS from floating-point format to fixed-point format.
FIXD	9E	1E	90-336	
FLTS	9D	1D	62-156	Convert TOS from fixed-point format to floating-point format.
FLTD	9C	1C	56-342	
CHSS	F4	74	22-24	Change sign of fixed-point operand on TOS.
CHSD	B4	34	26-28	
CHSF	95	15	16-20	Change sign of floating-point operandon TOS.
PTOS	F7	77	16	Push stack. Duplicate NOS in TOS.
PTOD	B7	37	20	
PTOF	97	17	20	
POPS	F8	78	10	Pop stack. Old NOS becomes new TOS. Old TOS rotates to bottom.
POPD	B8	38	12	
POPF	98	18	12	
XCHS	F9	79	18	Exchange TOS and NOS.
XCHD	B9	39	26	
XCHF	99	19	26	
PUPI	9A	1A	16	Push floating-point constant π onto TOS. Previous TOS becomes NOS.

COMMAND DESCRIPTIONS

This section contains detailed descriptions of the APU commands. They are arranged in alphabetical order by command mnemonic. In the descriptions, TOS means Top Of Stack, and NOS means Next On Stack.

All derived functions except Square Root use Chebyshev polynomial approximating algorithms. This approach is used to help minimize the internal microprogram, to minimize the maximum error values and to provide a relatively even distribution of errors over the data range. The basic arithmetic operations are used by the derived functions to compute the various Chebyshev terms. The basic operations may produce error codes in the status register as a result.

Execution times are listed in terms of clock cycles and may be converted into time values by multiplying by the clock period used. For example, an execution time of 44 clock cycles when running at a 3MHz rate translates to 14 microseconds ($44 \times 32\mu s = 14\mu s$). Variations in execution cycles reflect the data dependency of the algorithms.

In some operations exponent overflow or underflow may be possible. When this occurs, the exponent returned in the result will be 128 greater or smaller than its true value.

Many of the functions use portions of the data stack as scratch storage during development of the results. Thus, previous values in those stack locations will be lost. Scratch locations destroyed are listed in the command descriptions and shown with the crossed-out locations in the Stack Contents After diagram.

Table 1 is a summary of all the Am9511A commands. It shows the hex codes for each command, the mnemonic abbreviation, a brief description and the execution time in clock cycles. The commands are grouped by functional classes.

The command mnemonics in alphabetical order are shown below in Table 2.

Table 2.
Command Mnemonics in Alphabetical Order.

ACOS	ARCCOSINE
ASIN	ARCSINE
ATAN	ARCTANGENT
CHSD	CHANGE SIGN DOUBLE
CHSF	CHANGE SIGN FLOATING
CHSS	CHANGE SIGN SINGLE
COS	COSINE
DADD	DOUBLE ADD
DDIV	DOUBLE DIVIDE
DMUL	DOUBLE MULTIPLY LOWER
DMUU	DOUBLE MULTIPLY UPPER
DSUB	DOUBLE SUBTRACT
EXP	EXPONENTATION (e^X)
FADD	FLOATING ADD
FDIV	FLOATING DIVIDE
FIXD	FIX DOUBLE
FIXS	FIX SINGLE
FLTD	FLOAT DOUBLE
FLTS	FLOAT SINGLE
FMUL	FLOATING MULTIPLY
FSUB	FLOATING SUBTRACT
LOG	COMMON LOGARITHM
LN	NATURAL LOGARITHM
NOP	NO OPERATION
POPD	POP STACK DOUBLE
POPF	POP STACK FLOATING
POPS	POP STACK SINGLE
PTOD	PUSH STACK DOUBLE
PTOF	PUSH STACK FLOATING
PTOS	PUSH STACK SINGLE
PUPI	PUSH π
PWR	POWER (X^Y)
SADD	SINGLE ADD
SDIV	SINGLE DIVIDE
SIN	SINE
SMUL	SINGLE MULTIPLY LOWER
SMUU	SINGLE MULTIPLY UPPER
SQRT	SQUARE ROOT
SSUB	SINGLE SUBTRACT
TAN	TANGENT
XCHD	EXCHANGE OPERANDS DOUBLE
XCHF	EXCHANGE OPERANDS FLOATING
XCHS	EXCHANGE OPERANDS SINGLE

ACOS

32-BIT FLOATING-POINT INVERSE COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	0

Hex Coding: 86 with sr = 1
06 with sr = 0

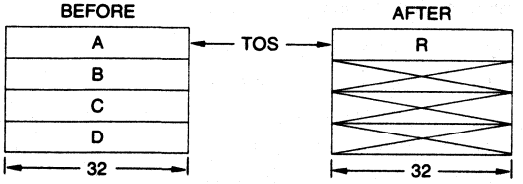
Execution Time: 6304 to 8284 k cycles clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse cosine of A. The result R is a value in radians between 0 and π. Initial operands A, B, C and D are lost. ACOS will accept all input data values within the range of −1.0 to +1.0. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ACOS exhibits a maximum relative error of 2.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000012

ATAN

32-BIT FLOATING-POINT INVERSE TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	1

Hex Coding: 87 with sr = 1
07 with sr = 0

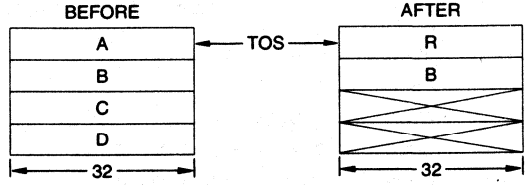
Execution Time: 4992 to 6536 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse tangent of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, C, and D are lost. Operand B is unchanged.
ATAN will accept all input data values that can be represented in the floating point format.

Accuracy: ATAN exhibits a maximum relative error of 3.0×10^{-7} over the input data range.

Status Affected: Sign, Zero

STACK CONTENTS



TB000013

ASIN

32-BIT FLOATING-POINT INVERSE SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	1

Hex Coding: 85 with sr = 1
05 with sr = 0

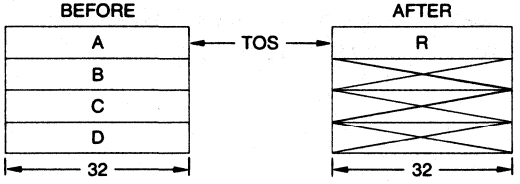
Execution Time: 6230 to 7938 k cycles clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse sine of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, B, C, and D are lost. ASIN will accept all input data values within the range of −1.0 to +1.0. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ASIN exhibits a maximum relative error of 4.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000014

CHSD

32-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	0	0

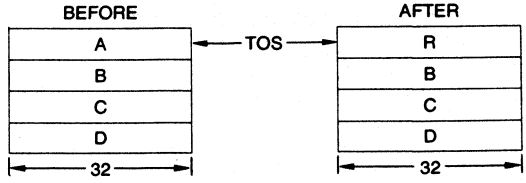
Hex Coding: B4 with sr = 1
34 with sr = 0

Execution Time: 26 to 28 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. Other entries in the stack are not disturbed.
Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



TB000047

CHSF

32-BIT FLOATING-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	0	1

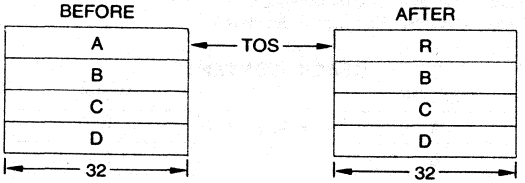
Hex Coding: 95 with sr = 1
15 with sr = 0

Execution Time: 16 to 20 clock cycles

Description:
The sign of the mantissa of the 32-bit floating-point operand A at the TOS is inverted. The result R replaces A at the TOS. Other stack entries are unchanged.
If A is input as zero (mantissa MSB = 0), no change is made.

Status Affected: Sign, Zero

STACK CONTENTS



TB000048

CHSS

16-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	0	0

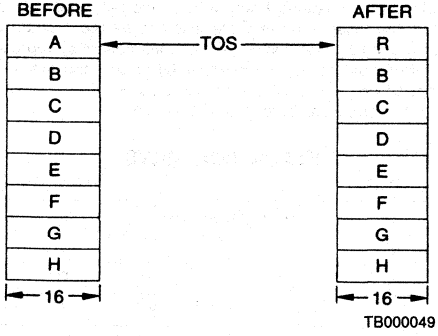
Hex Coding: F4 with sr = 1
74 with sr = 0

Execution Time: 22 to 24 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. All other operands are unchanged.
Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000049

COS

32-BIT FLOATING-POINT COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	1

Hex Coding: 83 with sr = 1
03 with sr = 0

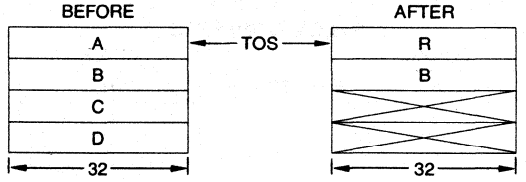
Execution Time: 3840 to 4878 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point cosine of A. A is assumed to be in radians. Operands A, C and D are lost. B is unchanged.
The COS function can accept any input data value that can be represented in the data format. All input values are range reduced to fall within an interval of $-\pi/2$ to $+\pi/2$ radians.

Accuracy: COS exhibits a maximum relative error of 5.0×10^{-7} for all input data values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



TB000015

DADD

32-BIT FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	0

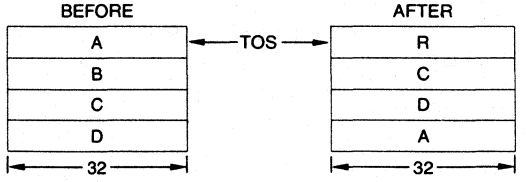
Hex Coding: AC with sr = 1
2C with sr = 0

Execution Time: 20 to 22 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is added to the 32-bit fixed-point two's complement integer operand B at the NOS. The result R replaces operand B, and the Stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged. If the addition generates a carry, it is reported in the status register.
If the result is too large to be represented by the data format, the least significant 32 bits of the result are returned, and overflow status is reported.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



TB000050

DDIV

32-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	1

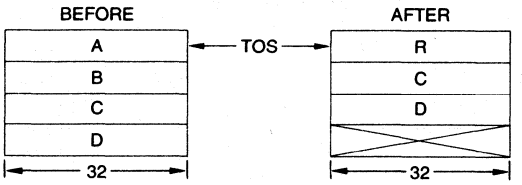
Hex Coding: AF with sr = 1
2F with sr = 0

Execution Time: 196 to 210 clock cycles when A ≠ 0
18 clock cycles when A = 0.

Description:
The 32-bit fixed-point two's complement integer operand B at NOS is divided by the 32-bit fixed-point two's complement integer operand A at the TOS. The 32-bit integer quotient R replaces B, and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. Operands C and D are unchanged.
If A is zero, R is set equal to B and the divided-by-zero error status will be reported. If either A or B is the most negative value possible in the format, R will be meaningless and the overflow error status will be reported.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000016

DMUU

32-BIT FIXED-POINT MULTIPLY, UPPER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	0

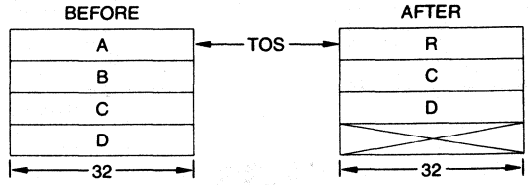
Hex Coding: B6 with sr = 1
36 with sr = 0

Execution Time: 182 to 218 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit most significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.
If A or B was the most negative value possible in the format, overflow status is set and R is meaningless.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000017

DMUL

32-BIT FIXED-POINT MULTIPLY, LOWER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	0

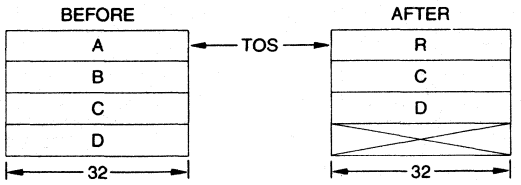
Hex Coding: AE with sr = 1
2E with sr = 0

Execution Time: 194 to 210 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit least significant half of the product R replaces B, and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.
The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB000018

DSUB

32-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	1

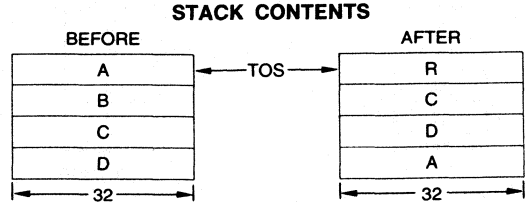
Hex Coding: AD with sr = 1
2D with sr = 0

Execution Time: 38 to 40 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is subtracted from the 32-bit fixed-point two's complement integer operand B at the NOS. The difference R replaces operand B, and the stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged.
If the subtraction generates a borrow, it is reported in the carry status bit. If A is the most negative value that can be represented in the format, the overflow status is set. If the result cannot be represented in the data format range, the overflow bit is set, and the 32 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Overflow

STACK CONTENTS



TB000051

EXP

32-BIT FLOATING-POINT e^X

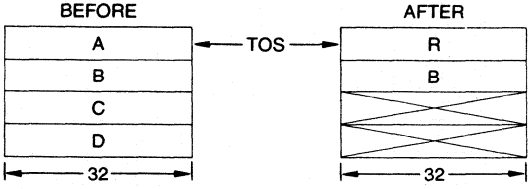
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	0

Hex Coding: 8A with sr = 1
0A with sr = 0

Execution Time: 3794 to 4878 clock cycles for $|A| \leq 1.0 \times 2^5$
34 clock cycles for $|A| > 1.0 \times 2^5$

Description:
The base of natural logarithms, e, is raised to an exponent value specified by the 32-bit floating-point operand A at the TOS. The result R of e^A replaces A. Operands A, C and D are lost. Operand B is unchanged.
EXP accepts all input data values within the range of $-1.0 \times 2^{+5}$ to $+1.0 \times 2^{+5}$. Input values outside this range will return a code of 1100 in the error field of the status register.
Accuracy: EXP exhibits a maximum relative error of 5.0×10^{-7} over the valid input data range.
Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB00019M

FADD

32-BIT FLOATING-POINT ADD

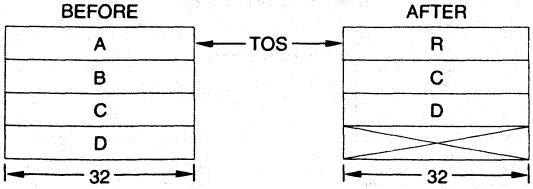
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	0

Hex Coding: 90 with sr = 1
10 with sr = 0

Execution Time: 54 to 368 clock cycles for A ≠ 0
24 clock cycles for A = 0

Description:
32-bit floating-point operand A at the TOS is added to 32-bit floating-point operand B at the NOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent alignment before the addition and normalization of the result accounts for the variation in execution time. Exponent overflow and underflow are reported in the status register, in which case the mantissa is correct and the exponent is offset by 128.
Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB00021M

FDIV

32-BIT FLOATING-POINT DIVIDE

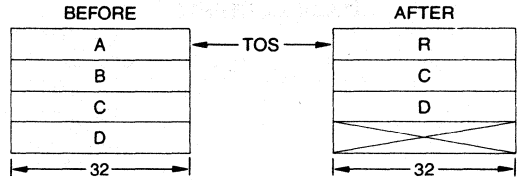
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	1

Hex Coding: 93 with sr = 1
13 with sr = 0

Execution Time: 154 to 184 clock cycles for A ≠ 0
22 clock cycles for A = 0

Description:
32-bit floating-point operand B at the NOS is divided by 32-bit floating-point operand A at the TOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
If operand A is zero, R is set equal to B and the divide-by-zero error is reported in the status register. Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.
Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000020

FIXD

32-BIT FLOATING-POINT TO 32-BIT FIXED-POINT CONVERSION

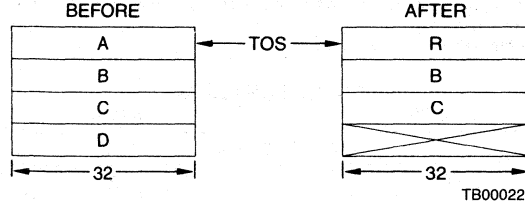
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	0

Hex Coding: 9E with sr = 1
1E with sr = 0

Execution Time: 90 to 336 clock cycles

Description:
32-bit floating-point operand A at the TOS is converted to a 32-bit fixed-point two's complement integer. The result R replaces A. Operands A and D are lost. Operands B and C are unchanged.
If the integer portion of A is larger than 31 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.
Status Affected: Sign, Zero, Overflow

STACK CONTENTS



TB00022M

FIXS

32-BIT FLOATING-POINT TO 16-BIT FIXED-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	1

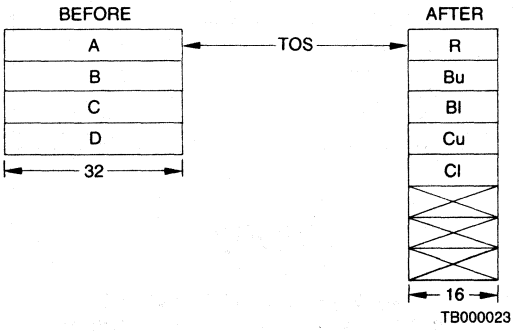
Hex Coding: 9F with sr = 1
1F with sr = 0

Execution Time: 90 to 214 clock cycles

Description:
32-bit floating-point operand A at the TOS is converted to a 16-bit fixed-point two's complement integer. The result R replaces the lower half of A, and the stack is moved up by two bytes so that R occupies the TOS. Operands A and D are lost. Operands B and C are unchanged, but appear as upper (u) and lower (l) halves on the 16-bit wide stack if they are 32-bit operands.
If the integer portion of A is larger than 15 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



FLTS

16-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	1

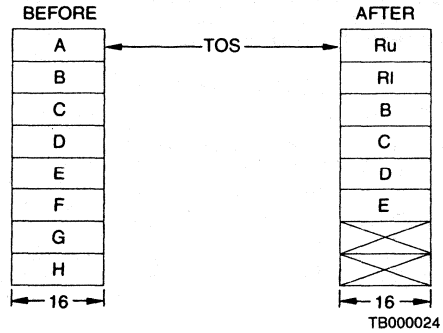
Hex Coding: 9D with sr = 1
1D with sr = 0

Execution Time: 62 to 156 clock cycles

Description:
16-bit fixed-point two's complement integer A at the TOS is converted to a 32-bit floating-point number. The lower half of the result R (Rl) replaces A; the upper half (Ru) replaces H, and the stack is moved down so that Ru occupies the TOS. Operands A, F, G and H are lost. Operands B, C, D and E are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



FLTD

32-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	0

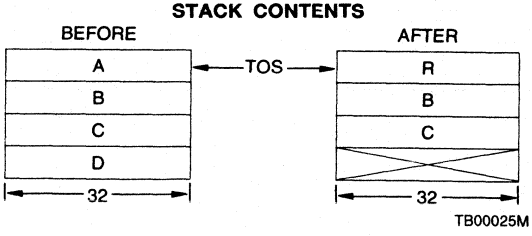
Hex Coding: 9C with sr = 1
1C with sr = 0

Execution Time: 56 to 342 clock cycles

Description:
32-bit fixed-point two's complement integer operand A at the TOS is converted to a 32-bit floating-point number. The result R replaces A at the TOS. Operands A and D are lost. Operands B and C are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



FMUL

32-BIT FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	0

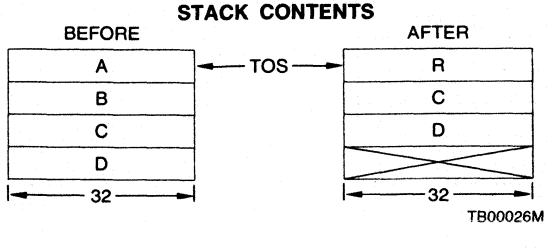
Hex Coding: 92 with sr = 1
12 with sr = 0

Execution Time: 146 to 168 clock cycles

Description:
32-bit floating-point operand A at the TOS is multiplied by the 32-bit floating-point operand B at the NOS. The normalized result R replaces B, and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



FSUB

32-BIT FLOATING-POINT SUBTRACTION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	1

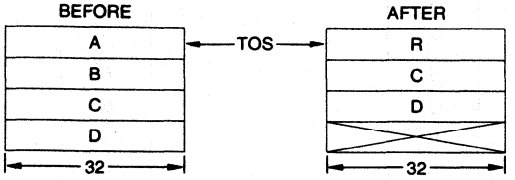
Hex Coding: 91 with sr = 1
11 with sr = 0

Execution Time: 70 to 370 clock cycles for A ≠ 0
26 clock cycles for A = 0

Description:
32-bit floating-point operand A at the TOS is subtracted from 32-bit floating-point operand B at the NOS. The normalized difference R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent alignment before the subtraction and normalization of the result account for the variation in execution time.
Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



TB000027

LOG

32-BIT FLOATING-POINT COMMON LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	0

Hex Coding: 88 with sr = 1
08 with sr = 0

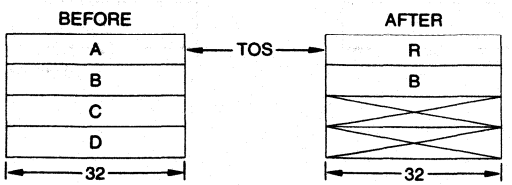
Execution Time: 4474 to 7132 clock cycles for A > 0
20 clock cycles for A ≤ 0

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point common logarithm (base 10) of A. Operands A, C and D are lost. Operand B is unchanged.
The LOG function accepts any positive input data value that can be represented by the data format. If LOG of a non-positive value is attempted, an error status of 0100 is returned.

Accuracy: LOG exhibits a maximum absolute error of 2.0×10^{-7} for the input range from 0.1 to 10, and a maximum relative error of 2.0×10^{-7} for positive values less than 0.1 or greater than 10.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000029

LN

32-BIT FLOATING-POINT NATURAL LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	1

Hex Coding: 89 with sr = 1
09 with sr = 0

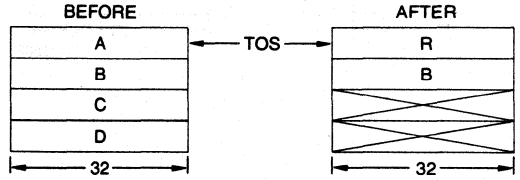
Execution Time: 4298 to 6956 clock cycles for A > 0
20 clock cycles for A ≤ 0

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point natural logarithm (base e) of A. Operands A, C and D are lost. Operand B is unchanged.
The LN function accepts all positive input data value that can be represented by the data format. If LN of a non-positive value is attempted, an error status of 0100 is returned.

Accuracy: LN exhibits a maximum absolute error of 2×10^{-7} for the input range from e^{-1} to e, and a maximum relative error of 2.0×10^{-7} for positive values less than e^{-1} or greater than e.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000028

NOP

NO OPERATION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	0

Hex Coding: 80 with sr = 1
00 with sr = 0

Execution Time: 4 clock cycles

Description:
The NOP command performs no internal data manipulations. It may be used to set or clear the service request interface line without changing the contents of the stack.

Status Affected: The status byte is cleared to all zeroes.

POPD

32-BIT STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	0

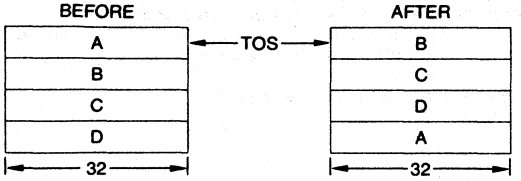
Hex Coding: B8 with sr = 1
38 with sr = 0

Execution Time: 12 clock cycles

Description:
The 32-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged. POPD and POPF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000082

POPS

16-BIT STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	1	0	0	0

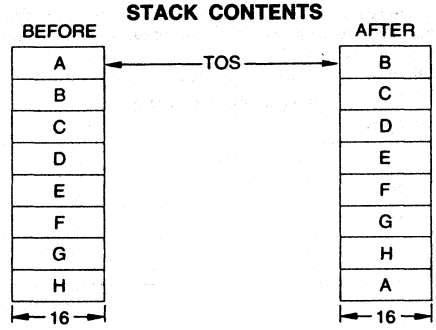
Hex Coding: F8 with sr = 1
78 with sr = 0

Execution Time: 10 clock cycles

Description:
The 16-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000052

POPF

32-BIT STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	0	0

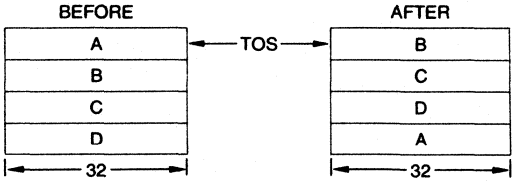
Hex Coding: 98 with sr = 1
18 with sr = 0

Execution Time: 12 clock cycles

Description:
The 32-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged. POPF and POPD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000053

PTOD

PUSH 32-BIT TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	1

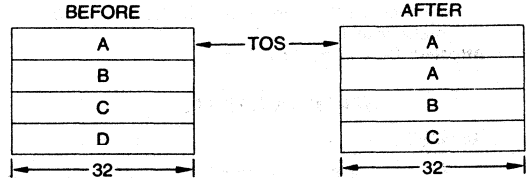
Hex Coding: B7 with sr = 1
37 with sr = 0

Execution Time: 20 clock cycles

Description:
The 32-bit stack is moved down, and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOD and PTOF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000054

PTOF

PUSH 32-BIT TOS ONTO STACK

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	0	1	0	1	1	1

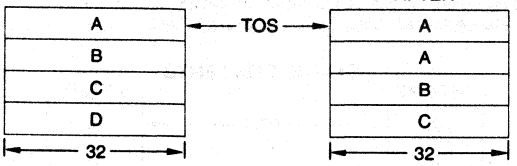
Hex Coding: 97 with sr = 1
17 with sr = 0

Execution Time: 20 clock cycles

Description:
The 32-bit stack is moved down, and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOF and PTOD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000055

PUPI

PUSH 32-BIT FLOATING-POINT π

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	0	1	1	0	1	0

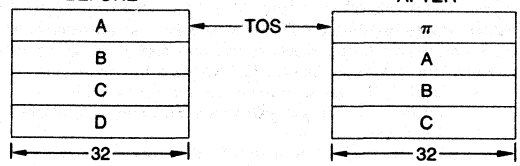
Hex Coding: 9A with sr = 1
1A with sr = 0

Execution Time: 16 clock cycles

Description:
The 32-bit stack is moved down so that the previous TOS occupies the new NOS location. 32-bit floating-point constant π is entered into the new TOS location. Operand D is lost. Operands A, B and C are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000056

PTOS

PUSH 16-BIT TOS ONTO STACK

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	1	0	1	1	1

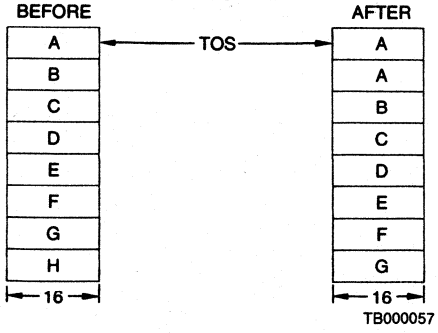
Hex Coding: F7 with sr = 1
77 with sr = 0

Execution Time: 16 clock cycles

Description:
The 16-bit stack is moved down, and the previous TOS is copied into the new TOS location. Operand H is lost, and all other operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000057

PWR

32-BIT FLOATING-POINT X^Y

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	1

Hex Coding: 8B with sr = 1
0B with sr = 0

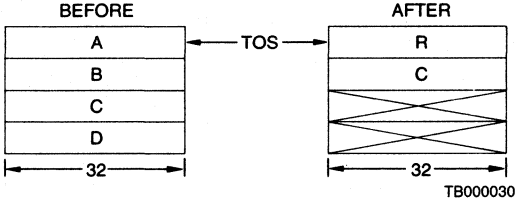
Execution Time: 8290 to 12032 clock cycles

Description:
32-bit floating-point operand B at the NOS is raised to the power specified by the 32-bit floating-point operand A at the TOS. The result R of B^A replaces B, and the stack is moved up so that R occupies the TOS. Operands A, B, and D are lost. Operand C is unchanged.
The PWR function accepts all input data values that can be represented in the data format for operand A and all positive values for operand B. If operand B is non-positive, an error status of 0100 will be returned. The EXP and LN functions are used to implement PWR using the relationship $B^A = EXP [A(LN B)]$. Thus, if the term [A(LN B)] is outside the range of $-1.0 \times 2^{+5}$, to $+1.0 \times 2^{+5}$, an error status of 1100 will be returned. Underflow and overflow conditions can occur.

Accuracy: The error performance for PWR is a function of the LN and EXP performance as expressed by:
$|(\text{Related Error})_{PWR}| = |(\text{Relative Error})_{EXP} + |A(\text{Absolute Error})_{LN}|$
The maximum relative error for PWR occurs when A is at its maximum value while [A(LN B)] is near 1.0×2^5 and the EXP error is also at its maximum. For most practical applications, the relative error for PWR will be less than 7.0×10^{-7}.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000030

SADD

16-BIT FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	0

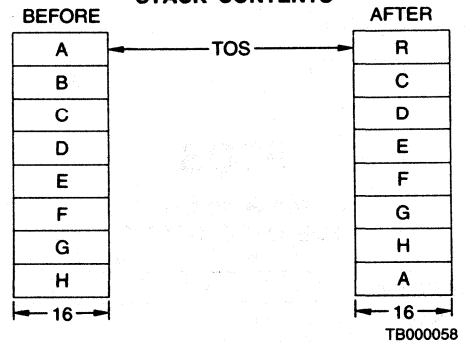
Hex Coding: EC with sr = 1
6C with sr = 0

Execution Time: 16 to 18 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is added to 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.
If the addition generates a carry bit, it is reported in the status register. If an overflow occurs, it is reported in the status register, and the 16 least significant bits of the result are returned.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



TB000058

SDIV

16-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	1	1

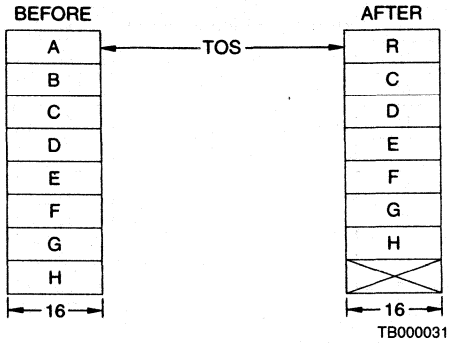
Hex Coding: EF with sr = 1
6F with sr = 0

Execution Time: 84 to 94 clock cycles for $A \neq 0$
14 clock cycles for A = 0

Description:
16-bit fixed-point two's complement integer operand B at the NOS is divided by 16-bit fixed-point two's complement integer operand A at the TOS. The 16-bit quotient R replaces B, and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. All other operands are unchanged.
If A is zero, R will be set equal to B and the divide-by-zero error status will be reported.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000031

SIN

32-BIT FLOATING-POINT SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	0

Hex Coding: 82 with sr = 1
02 with sr = 0

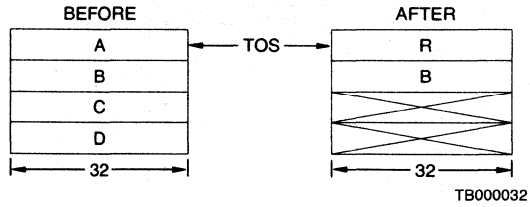
Execution Time: 3796 to 4808 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycles for $|A| \leq 2^{-12}$ radians

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point sine of A. A is assumed to be in radians. Operands A, C and D are lost. Operand B is unchanged.
The SIN function will accept any input data value that can be represented by the data format. All input values are range reduced to fall within the interval $-\pi/2$ to $+\pi/2$ radians.

Accuracy: SIN exhibits a maximum relative error of 5.0×10^{-7} for input values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



TB000032

SMUL

16-BIT FIXED-POINT MULTIPLY, LOWER

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	0	1	1	1	0

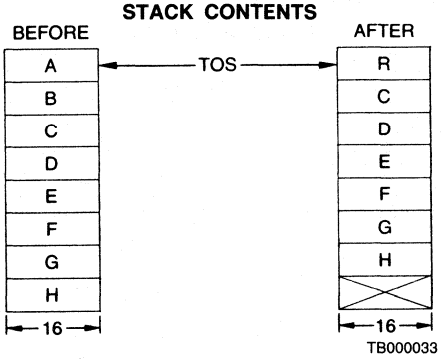
Hex Coding: EE with sr = 1
6E with sr = 0

Execution Time: 84 to 94 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit least significant half of the product R replaces B, and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. All other operands are unchanged. The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SMUU

16-BIT FIXED-POINT MULTIPLY, UPPER

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	1	0	1	1	0

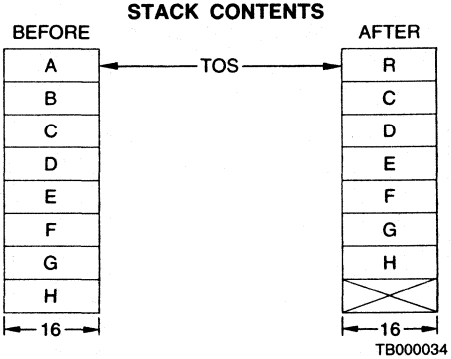
Hex Coding: F6 with sr = 1
76 with sr = 0

Execution Time: 80 to 98 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit most significant half of the product R replaces B, and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. All other operands are unchanged.
If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SQRT

32-BIT FLOATING-POINT SQUARE ROOT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	1

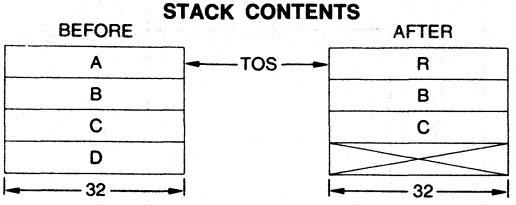
Hex Coding: 81 with sr = 1
01 with sr = 0

Execution Time: 782 to 870 clock cycles

Description:
32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point square root of A. Operands A and D are lost. Operands B and C are unchanged.
SQRT will accept any non-negative input data that can be represented by the data format. If A is negative an error code of 0100 will be returned in the status register.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



TB000035

SSUB

16-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	1

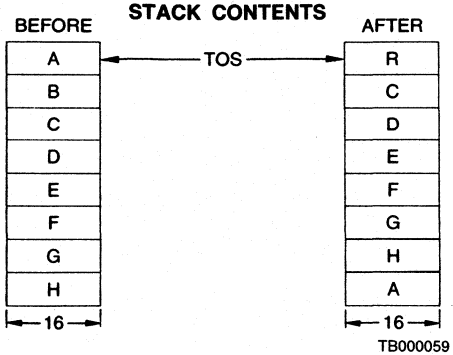
Hex Coding: ED with sr = 1
6D with sr = 0

Execution Time: 30 to 32 clock cycles

Description:
16-bit fixed-point two's complement integer operand A at the TOS is subtracted from 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B, and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.
If the subtraction generates a borrow, it is reported in the carry status bit. If A is the most negative value that can be represented in the format, the overflow status is set. If the result cannot be represented in the format range, the overflow status is set and the 16 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



TB000059

TAN

32-BIT FLOATING-POINT TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	0

Hex Coding: 84 with sr = 1
04 with sr = 0

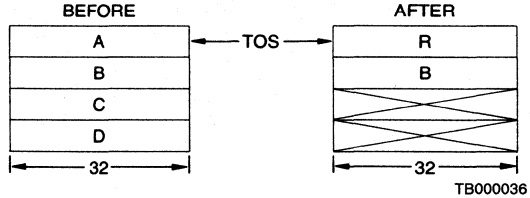
Execution Time: 4894 to 5886 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycle for $|A| \leq 2^{-12}$ radians

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point tangent of A. Operand A is assumed to be in radians. A, C and D are lost. B is unchanged.
The TAN function will accept any input data value that can be represented in the data format. All input data values are range-reduced to fall within $-\pi/4$ to $+\pi/4$ radians. TAN is unbounded for input values near odd multiples of $\pi/2$, and in such cases, the overflow bit is set in the status register. For angles smaller than 2^{-12} radians, TAN returns A as the tangent of A.

Accuracy: TAN exhibits a maximum relative error of 5.0×10^{-7} for input data values in the range of -2π to $+2\pi$ radians except for data values near odd multiples of $\pi/2$.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



TB000036

XCHD

EXCHANGE 32-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	1

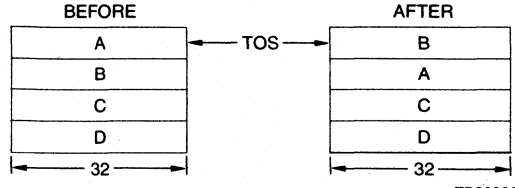
Hex Coding: B9 with sr = 1
39 with sr = 0

Execution Time: 26 clock cycles

Description:
32-bit operand A at the TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000060

XCHF

EXCHANGE 32-BIT STACK OPERANDS

Binary Coding:

7	6	5	4	3	2	1	0
sr	0	0	1	1	0	0	1

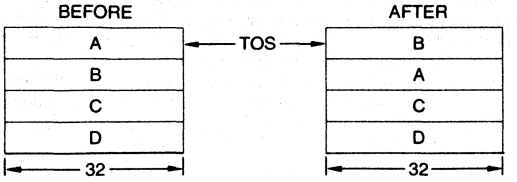
Hex Coding: 99 with sr = 1
19 with sr = 0

Execution Time: 26 clock cycles

Description:
32-bit operand A at all TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



TB000061

XCHS

EXCHANGE 16-BIT STACK OPERANDS

Binary Coding:

7	6	5	4	3	2	1	0
sr	1	1	1	1	0	0	1

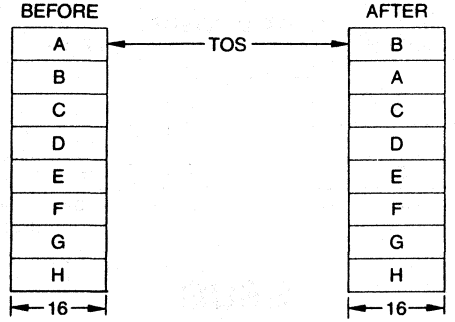
Hex Coding: F9 with sr = 1
79 with sr = 0

Execution Time: 18 clock cycles

Description:
16-bit operand A at the TOS and 16-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



TB000062

APPLICATIONS INFORMATION

The diagram in Figure 2 shows the interface connections for the Am9511A APU with operand transfers handled by an Am9517A DMA controller and CPU coordination handled by an Am9519A Interrupt Controller. The APU interrupts the CPU to indicate that a command has been completed. When the performance enhancements provided by the DMA and Interrupt operations are not required, the APU interface can be simplified as shown in Figure 1. The Am9511A APU is designed with a general purpose 8-bit data bus and interface control so that it can be conveniently used with any general 8-bit processor.

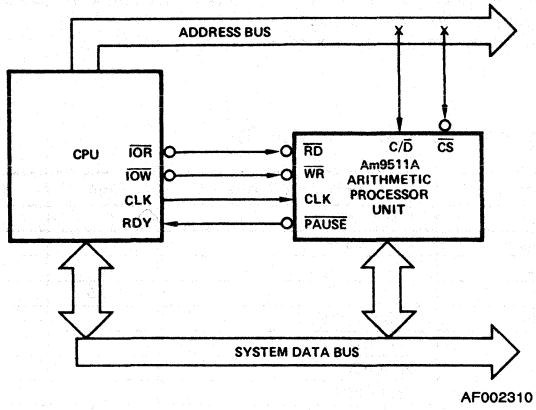


Figure 1. Am9511A Minimum Configuration Example.

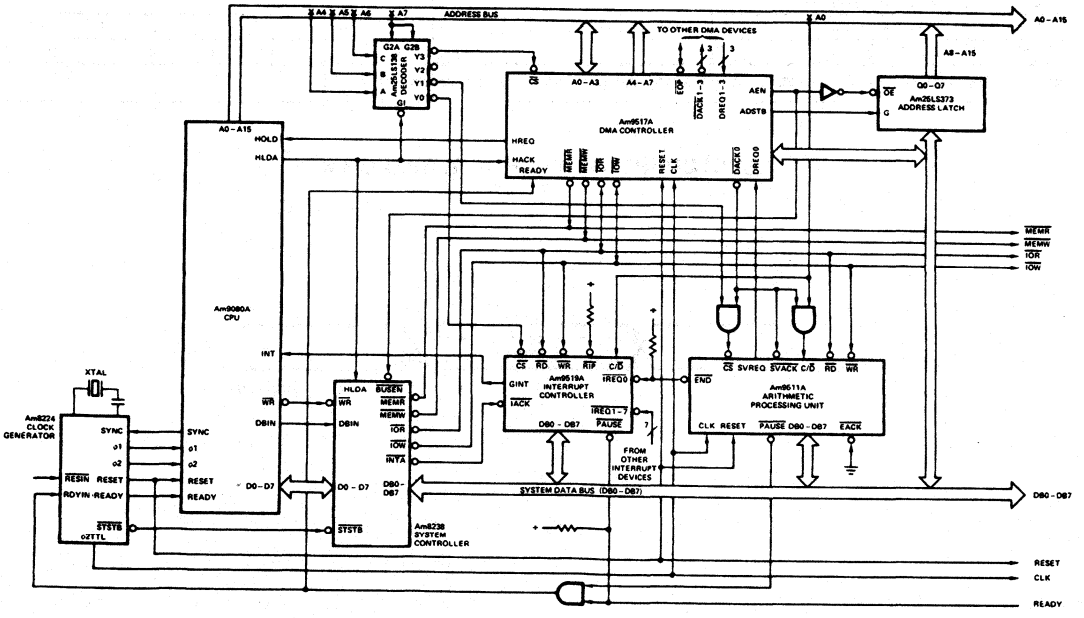


Figure 2. Am9511A High Performance Configuration Example.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
VDD with Respect to VSS ..... −0.5V to +15.0V
VCC with Respect to VSS ..... −0.5V to +7.0V
All Signal Voltages
with Respect to VSS ..... −5.0V to +7.0V
Power Dissipation (Package Limitation) 2.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	T_A	V_{SS}	V_{CC}	V_{DD}
Commercial	0°C to 70°C	0V	5V ±5%	12 ±5%
Industrial	−40°C to 85°C	0V	5V ±10%	12 ±10%

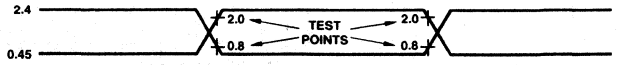
Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
VOH	Output HIGH Voltage	IOH = −200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			10	μA
		VO = VCC			10	
ICC	VCC Supply Current	T_A = +25°C		50	90	mA
		T_A = 0°C			95	
		T_A = −55°C			100	
IDD	VDD Supply Current	T_A = +25°C		50	90	mA
		T_A = 0°C			95	
		T_A = −55°C			100	
CO	Output Capacitance	fc = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

Note: 1. Typical values are for T_A = 25°C, normal supply voltages and normal processing parameters.

SWITCHING TEST INPUT WAVEFORM



WF004060

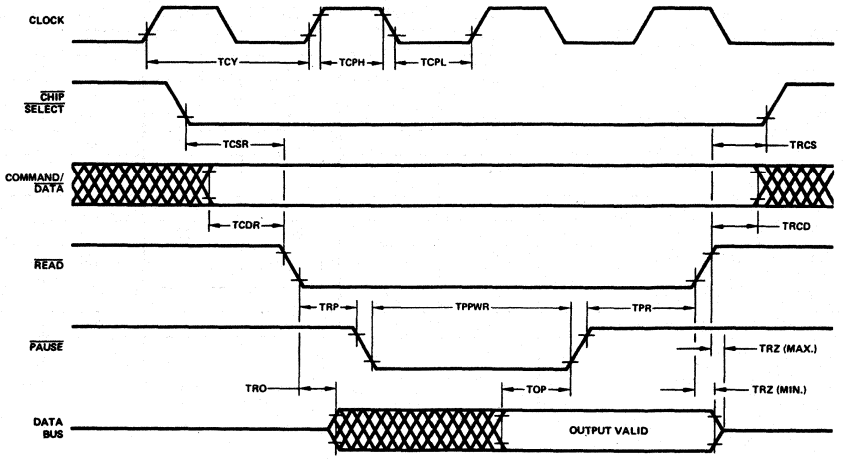
SWITCHING CHARACTERISTICS over operating range (Note 1)

Parameters	Description		Am9511A Min	Am9511A Max	Am9511A-1 Min	Am9511A-1 Max	Am9511A-4 Min	Am9511A-4 Max	Units
TAPW	$\overline{EACK}$ LOW Pulse Width		100		75		50		ns
TCDR	C/$\overline{D}$ to $\overline{RD}$ LOW Set-up Time		0		0		0		ns
TCDW	C/$\overline{D}$ to $\overline{WR}$ LOW Set-up Time		0		0		0		ns
TCPH	Clock Pulse HIGH Width		200		140		100		ns
TCPL	Clock Pulse LOW Width		240		160		120		ns
TCSR	$\overline{CS}$ LOW to $\overline{RD}$ LOW Set-up Time		0		0		0		ns
TCSW	$\overline{CS}$ LOW to $\overline{WR}$ LOW Set-up Time		0		0		0		ns
TCY	Clock Period		480	5000	320	3300	250	2500	ns
TDW	Data Bus Stable to $\overline{WR}$ HIGH Set-up Time		150		100 (Note 9)		100		ns
TEAE	$\overline{EACK}$ LOW to $\overline{END}$ HIGH Delay			200		175		150	ns
TEPW	$\overline{END}$ LOW Pulse Width (Note 4)		400		300		200		ns
TOP	Data Bus Output Valid to $\overline{PAUSE}$ HIGH Delay		0		0		0		ns
TPPWR	$\overline{PAUSE}$ LOW Pulse Width Read (Note 5)	Data	3.5TCY + 50	5.5TCY + 300	3.5TCY + 50	5.5TCY + 200	3.5TCY + 50	5.5TCY + 200	ns
		Status	1.5TCY + 50	3.5TCY + 300	1.5TCY + 50	3.5TCY + 200	1.5TCY + 50	3.5TCY + 200	
TPPWW	$\overline{PAUSE}$ LOW Pulse Width Write (Note 8)			50		50		50	ns
TPR	$\overline{PAUSE}$ HIGH to $\overline{RD}$ HIGH Hold Time		0		0		0		ns
TPW	$\overline{PAUSE}$ HIGH to $\overline{WR}$ HIGH Hold Time		0		0		0		ns
TRCD	$\overline{RD}$ HIGH to C/$\overline{D}$ Hold Time		0		0		0		ns
TRCS	$\overline{RD}$ HIGH to $\overline{CS}$ HIGH Hold Time		0		0		0		ns
TRO	$\overline{RD}$ LOW to Data Bus ON Delay		50		50		25		ns
TRP	$\overline{RD}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 6)			150		100 (Note 9)		100	ns
TRZ	$\overline{RD}$ HIGH to Data Bus OFF Delay		50	200	50	150	25	100	ns
TSAPW	$\overline{SVACK}$ LOW Pulse Width		100		75		50		ns
TSAR	$\overline{SVACK}$ LOW to SVREQ LOW Delay			300		200		150	ns
TWCD	$\overline{WR}$ HIGH to C/$\overline{D}$ Hold Time		60		30		30		ns
TWCS	$\overline{WR}$ HIGH to $\overline{CS}$ HIGH Hold Time		60		30		30		ns
TWD	$\overline{WR}$ HIGH to Data Bus Hold Time		20		20		20		ns
TWI	Write Inactive Time	Command	3TCY		3TCY		3TCY		ns
		Data	4TCY		4TCY		4TCY		
TWP	$\overline{WR}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 6)			150		100 (Note 9)		100	ns

Notes:
1. Typical values are for $T_A = 25°C$, nominal supply voltages and nominal processing parameters.
2. Switching parameters are listed in alphabetical order.
3. Test conditions assume transition times of 20ns or less, output loading of one TTL gate plus 100pF and timing reference levels of 0.8V and 2.0V.
4. $\overline{END}$ low pulse width is specified for $\overline{EACK}$ tied to VSS. Otherwise TEAE applies.
5. Minimum values shown assume no previously entered command is being executed for the data access. If a previously entered command is being executed, $\overline{PAUSE}$ LOW Pulse Width is the time to complete execution plus the time shown. Status may be read at any time without exceeding the time shown.
6. $\overline{PAUSE}$ is pulled low for both command and data operations.
7. TEX is the execution time of the current command (see the Command Execution Times table).
8. $\overline{PAUSE}$ low pulse width is less than 50ns when writing into the data port or the control port as long as the duty requirement (TWI) is observed and no previous command is being executed. TWI may be safely violated up to 500ns as long as the extended TPPWW that results is observed. If a previously entered command is being executed, $\overline{PAUSE}$ LOW Pulse Width is the time to complete execution plus the time shown.
9. 150ns for Military grade.

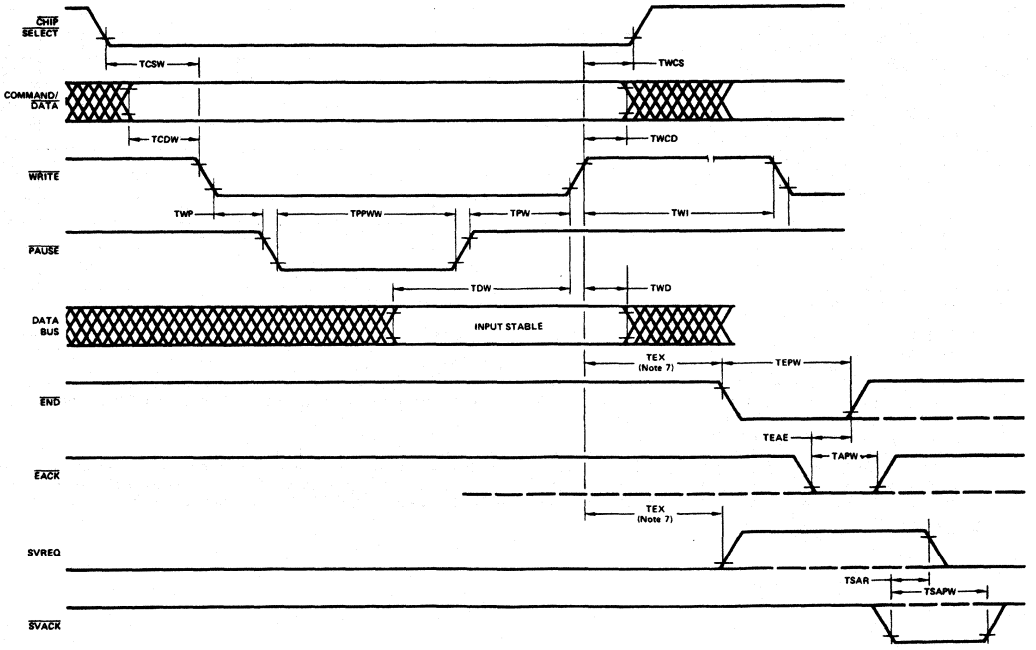
SWITCHING WAVEFORMS

READ OPERATIONS



WF004070

WRITE OPERATIONS



WF004080

Am9512

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- Single (32-bit) and double (64-bit) precision capability
- Add, subtract, multiply and divide functions
- Compatible with proposed IEEE format
- Easy interfacing to microprocessors
- 8-bit data bus
- Standard 24-pin package
- 12V and 5V power supplies
- Stack oriented operand storage
- Direct memory access or programmed I/O Data Transfers
- End of execution signal
- Error interrupt
- All inputs and outputs TTL level compatible
- Advanced N-channel silicon gate MOS technology

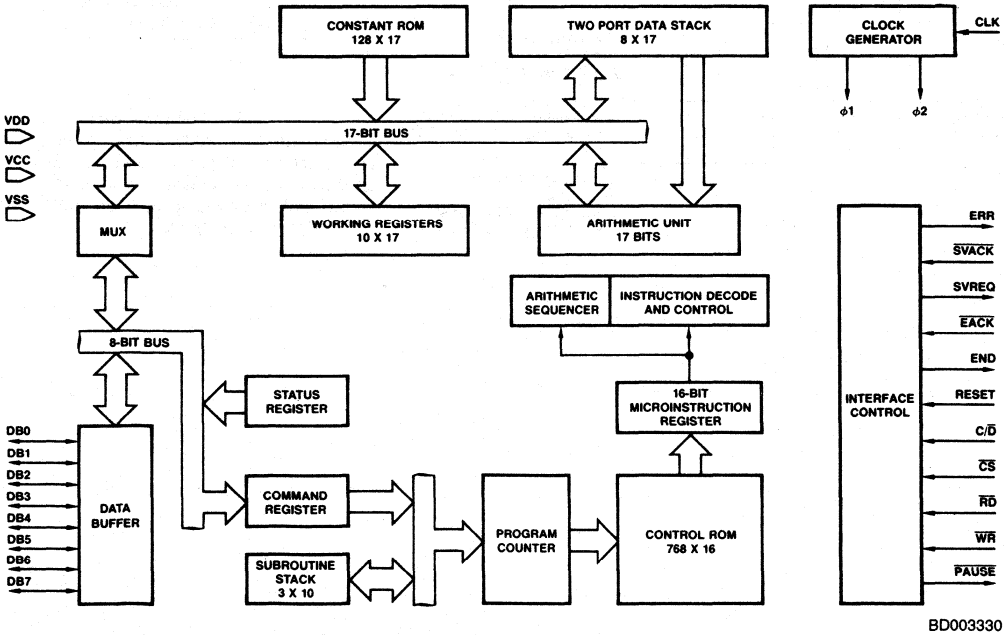
GENERAL DESCRIPTION

The Am9512 is a high performance floating-point processor unit (FPU). It provides single precision (32-bit) and double precision (64-bit) add, subtract, multiply and divide opertions. It can be easily interfaced to enhance the computational capabilities of the host microprocessor.

The operand, result, status and command information transfers take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack by the host processor, and a command is issued to perform an operation on the data stack. The results of this operation are available to the host processor by popping the stack.

Information transfers between the Am9512 and the host processor can be handled by using programmed I/O or direct memory access techniques. After completing an operation, the Am9512 activates an "end of execution" signal that can be used to interrupt the host processor.

BLOCK DIAGRAM

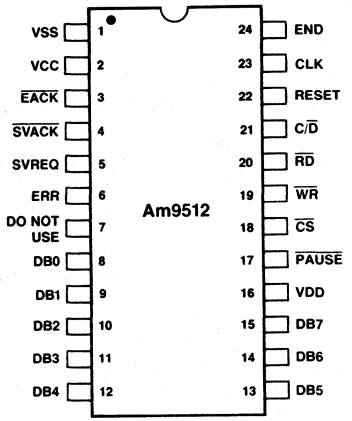


BD003330

02047B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

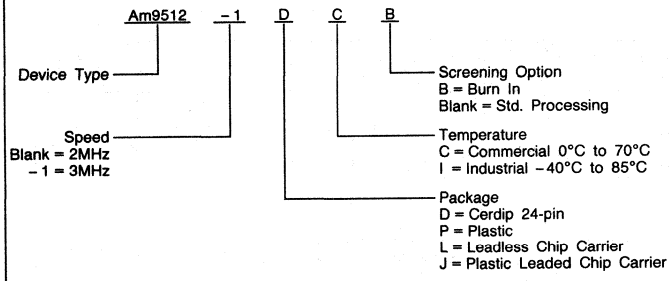


CD005160

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9512	DC, DCB, DI, DIB,
Am9512-1	LC, LCB, LI, LIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
23	CLK	I	(Clock). An external timing source connected to the CLK input provides the necessary clocking.
22	RESET	I	(Reset). A HIGH on this input causes initialization. Reset terminates any operation in progress, and clears the status register to zero. The internal stack pointer is initialized, and the contents of the stack may be affected. After a reset, the END output, the ERR output and the SVREQ output will be LOW. For proper initialization, RESET must be HIGH for at least five CLK periods following stable power supply voltages and stable clock.
21	C/$\overline{D}$	I	(Command/Data Select). The C/$\overline{D}$ input together with the $\overline{RD}$ and $\overline{WR}$ inputs determines the type of transfer to be performed on the data bus as follows:

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Function
L	H	L	Push data byte onto the stack
L	L	H	Pop data byte from the stack
H	H	L	Enter command
H	L	H	Read Status
X	L	L	Undefined

L = LOW
H = HIGH
X = DON'T CARE

24	END	O	(End of Execution). A HIGH on this output indicates that execution of the current command is complete. This output will be cleared LOW by activating the $\overline{EACK}$ input LOW or performing any read or write operation or device initialization using the RESET. If $\overline{EACK}$ is tied LOW, the END output will be a pulse (see $\overline{EACK}$ description). Reading the status register while a command execution is in progress is allowed. However, any read or write operation clears the flip-flop that generates the END output. Thus, such continuous reading could conflict with internal logic setting of the END flip-flop at the end of command execution.
3	$\overline{EACK}$	I	(End Acknowledge). This input when LOW makes the END output go LOW. As mentioned earlier, HIGH on the END output signals completion of a command execution. The END signal is derived from an internal flip-flop which is clocked at the completion of a command. This flip-flop is clocked to the reset state when $\overline{EACK}$ is LOW. Consequently, if $\overline{EACK}$ is tied LOW, the END output will be a pulse that is approximately one CLK period wide.
5	SVREQ	O	(Service Request). A HIGH on this output indicates completion of a command. In this sense, this output is the same as the END output. However, the Service Bit in the Command Register determines whether the SVREQ output will go HIGH at the completion of a command. This bit must be 1 for SVREQ to go HIGH. The SVREQ can be cleared (i.e., go LOW) by activating the $\overline{SVACK}$ input LOW or initializing the device using the RESET. Also, the SVREQ will be automatically cleared after completion of any command that has the service request bit as 0.
4	$\overline{SVACK}$	I	(Service Acknowledge). A LOW on this input clears SVREQ. If the $\overline{SVACK}$ input is permanently tied LOW, it will conflict with the internal setting of the SVREQ output. Thus, the SVREQ indication cannot be relied upon if the $\overline{SVACK}$ is tied LOW.
8-15	DB0–DB7	I/O	(Data Bus). These eight bidirectional lines are used to transfer command, status and operand information between the device and the host processor. DB0 is the least significant and DB7 is the most significant bit position. HIGH on a data bus line corresponds to 1 and LOW corresponds to 0. When pushing operands onto the stack using the data bus, the least significant byte must be pushed first and the most significant byte last. When popping the stack to read the result of an operation, the most significant byte will be available on the data bus first and the least significant byte will be the last. Moreover, for pushing operands and popping results, the number of transactions must be equal to the proper number of bytes appropriate for the chosen format. Otherwise, the internal byte pointer will not be aligned properly. The Am9512 single precision format requires 4 bytes, and double precision format requires 8 bytes.
6	ERR	O	(Error). This output goes HIGH to indicate that the current command execution resulted in an error condition. The error conditions are: attempt to divide by zero, exponent overflow and exponent underflow. The ERR output is cleared LOW on read status register operation or upon RESET. The ERR output is derived from the error bits in the status register. These error bits will be updated internally at an appropriate time during a command execution. Thus, ERR output going HIGH may not correspond with the completion of a command. Reading of the status register can be performed while a command execution is in progress. However, it should be noted that reading the status register clears the ERR output. Thus, reading the status register while a command execution is in progress may result in an internal conflict with the ERR output.
18	$\overline{CS}$	I	(Chip Select). This input must be LOW to accomplish any read or write operation to the Am9512. To perform a write operation, appropriate data is presented on DB0 through DB7 lines, appropriate logic level on the C/$\overline{D}$ input, and the $\overline{CS}$ input is made LOW. Whenever $\overline{WR}$ and $\overline{RD}$ inputs are both HIGH and $\overline{CS}$ is LOW, $\overline{PAUSE}$ goes LOW. However, actual writing into the Am9512 cannot start until $\overline{WR}$ is made LOW. After initiating the write operation by the HIGH-to-LOW transition on the $\overline{WR}$ input, the $\overline{PAUSE}$ output will go HIGH, indicating the write operation has been acknowledged. The $\overline{WR}$ input can go HIGH after $\overline{PAUSE}$ goes HIGH. The data lines, C/$\overline{D}$ input and the $\overline{CS}$ input can change when appropriate hold time requirements are satisfied. See Write Timing diagram for details. To perform a read operation, an appropriate logic level is established on the C/$\overline{D}$ input and $\overline{CS}$ is made LOW. The $\overline{PAUSE}$ output goes LOW because $\overline{WR}$ and $\overline{RD}$ inputs are HIGH. The read operation does not start until the $\overline{RD}$ input goes LOW. $\overline{PAUSE}$ will go HIGH, indicating that the read operation is complete and the required information is available on the DB0 through DB7 lines. This information will remain on the data lines as long as $\overline{RD}$ is LOW. The $\overline{RD}$ input can return HIGH anytime after $\overline{PAUSE}$ goes HIGH. The $\overline{CS}$ input and C/$\overline{D}$ input can change anytime after $\overline{RD}$ returns HIGH. See Read Timing diagram for details. If the $\overline{CS}$ is tied LOW permanently, $\overline{PAUSE}$ will remain LOW until the next Am9512 read or write access.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
20	$\overline{RD}$	I	(Read). A LOW on this input is used to read information from an internal location and gate that information onto the data bus. The $\overline{CS}$ input must be LOW to accomplish the read operation. The C/$\overline{D}$ input determines what internal location is of interest. See C/$\overline{D}$, $\overline{CS}$ input descriptions and Read Timing diagram for details. If the END output was HIGH, performing any read operation will make the END output go LOW after the HIGH-to-LOW transition of the $\overline{RD}$ input (assuming $\overline{CS}$ is LOW). If the ERR output was HIGH, performing a status register read operation will make the ERR output LOW. This will happen after the HIGH-to-LOW transition of the RD input (assuming $\overline{CS}$ is LOW).
19	$\overline{WR}$	I	(Write). A LOW on this input is used to transfer information from the data bus into an internal location. The $\overline{CS}$ must be LOW to accomplish the write operation. The C/$\overline{D}$ determines which internal location is to be written. See C/$\overline{D}$, $\overline{CS}$ input descriptions and Write Timing diagram for details. If the END output was HIGH, performing any write operation will make the END output go LOW after the LOW-to-HIGH transition of the $\overline{WR}$ input (assuming $\overline{CS}$ is LOW).
17	$\overline{PAUSE}$	O	(Pause). This output is a handshake signal used while performing read or write transactions with the Am9512. If the $\overline{WR}$ and $\overline{RD}$ inputs are both HIGH, the $\overline{PAUSE}$ output goes LOW with the $\overline{CS}$ input in anticipation of a transaction. If $\overline{WR}$ goes LOW to initiate a write transaction with proper signals established on the DB0 - DB7, C/$\overline{D}$ inputs, the $\overline{PAUSE}$ will return HIGH, indicating that the write operation has been accomplished. The $\overline{WR}$ can be made HIGH after this event. On the other hand, if a read operation is desired, the $\overline{RD}$ input is made LOW after activating $\overline{CS}$ LOW and establishing proper C/$\overline{D}$ input. (The $\overline{PAUSE}$ will go LOW in response to $\overline{CS}$ going LOW.) The $\overline{PAUSE}$ will return HIGH indicating completion of read. The $\overline{RD}$ can return HIGH after this event. It should be noted that a read or write operation can be initiated without any regard to whether a command execution is in progress or not. Proper device operation is assured by obeying the $\overline{PAUSE}$ output indication as described.
2	V_{CC}		+5V Power Supply.
16	V_{DD}		+12V Power Supply.
1	V_{SS}		Ground.

DETAILED DESCRIPTION

Major functional units of the Am9512 are shown in the block diagram. The Am9512 employs a microprogram controlled stack oriented architecture with 17-bit wide data paths.

The Arithmetic Unit receives one of its operands from the Operand Stack. This stack is an eight word by 17-bit two port memory with last in – first out (LIFO) attributes. The second operand to the Arithmetic Unit is supplied by the internal 17-bit bus. In addition to supplying the second operand, this bidirectional bus also carries the results from the output of the Arithmetic Unit when required. Writing into the Operand Stack takes place from this internal 17-bit bus when required. Also, connected to this bus are the Constant ROM and Working Registers. The ROM provides the required constants to perform the mathematical operations while the Working Registers provide storage for the intermediate values during command execution.

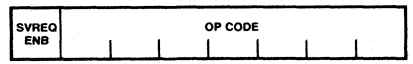
Communication between the external world and the Am9512 takes place on eight bidirectional input/output lines, DB0 through DB7 (Data Bus). These signals are gated to the internal 8-bit bus through appropriate interface and buffer circuitry. Multiplexing facilities exist for bidirectional communication between the internal eight and 17-bit buses. The Status Register and Command Register are also located on the 8-bit bus.

The Am9512 operations are controlled by the microprogram contained in the Control ROM. The Program Counter supplies the microprogram addresses and can be partially loaded from the Command Register. Associated with the Program Counter is the Subroutine Stack where return addresses are held during subroutine calls in the microprogram. The Microinstruction Register holds the current microinstruction being executed. The register facilitates pipelined microprogram execution. The Instruction Decode logic generates various internal control signals needed for the Am9512 operation.

The Interface Control logic receives several external inputs and provides handshake related outputs to facilitate interfacing the Am9512 to microprocessors.

Command Format

The Operation of the Am9512 is controlled from the host processor by issuing instructions called commands. The command format is shown below:



DF001330

The command consists of 8 bits; the least significant 7 bits specify the operation to be performed as detailed in the accompanying table. The most significant bit is the Service Request Enable bit. This bit must be a 1 if SVREQ is to go high at end of executing a command.

The Am9512 commands fall into three categories: single precision arithmetic, double precision arithmetic and data manipulation. There are four arithmetic operations that can be performed with single precision (32-bit) or double precision (64-bit) floating-point numbers: add, subtract, multiply and divide. These operations require two operands. The Am9512 assumes that these operands are located in the internal stack as Top of Stack (TOS) and Next on Stack (NOS). The result will always be returned to the previous NOS which becomes the new TOS. Results from an operation are of the same precision and format as the operands. The results will be rounded to preserve the accuracy. The actual data formats and rounding procedures are described in a later section. In addition to the arithmetic operations, the Am9512 implements eight data manipulating operations. These include changing the sign of a double or single precision operand located in TOS, exchanging single precision operands located at TOS and NOS, as well as copying and popping single or double precision operands. See also the sections on status register and operand formats.

The Execution times of the Am9512 commands are all data dependent. Table 2 shows one example of each command execution time.

Table 1. Command Decoding Table.

Command Bits								Mnemonic	Description
7	6	5	4	3	2	1	0		
X	0	0	0	0	0	0	1	SADD	Add TOS to NOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	0	1	0	SSUB	Subtract TOS from NOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	0	1	1	SMUL	Multiply NOS by TOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	1	0	0	SDIV	Divide NOS by TOS Single Precision and result to NOS. Pop stack.
X	0	0	0	0	1	0	1	CHSS	Change sign of TOS Single Precision operand.
X	0	0	0	0	1	1	0	PTOS	Push Single Precision operand on TOS to NOS.
X	0	0	0	0	1	1	1	POPS	Pop Single Precision operand from TOS. NOS becomes TOS.
X	0	0	0	1	0	0	0	XCHS	Exchange TOS with NOS Single Precision.
X	0	1	0	1	1	0	1	CHSD	Change sign of TOS Double Precision operand.
X	0	1	0	1	1	1	0	PTOD	Push Double Precision operand on TOS to NOS.
X	0	1	0	1	1	1	1	POPD	Pop Double Precision operand from TOS. NOS becomes TOS.
X	0	0	0	0	0	0	0	CLR	CLR status.
X	0	1	0	1	0	0	1	DADD	Add TOS to NOS Double Precision and result to NOS. Pop stack.
X	0	1	0	1	0	1	0	DSUB	Subtract TOS from NOS Double Precision and result to NOS. Pop Stack.
X	0	1	0	1	0	1	1	DMUL	Multiply NOS by TOS Double Precision and result to NOS. Pop Stack.
X	0	1	0	1	1	0	0	DDIV	Divide NOS by TOS Double Precision and result to NOS. Pop Stack.

Note: X = Don't Care Operation for bit combinations not listed above is undefined.

Table 2. Am9512 Execution Time in Cycles.

Single Precision			
	Min	Typ	Max
Add	58	220	512
Subtract	56	220	512
Multiply	192	220	254
Divide	228	240	264

Double Precision			
	Min	Typ	Max
Add	578	1200	3100
Subtract	578	1200	3100
Multiply	1720	1770	1860
Divide	4560	4920	5120

Note: Typical for add and subtract, assumes the operands are within six decimal orders of magnitude. Max is derived from the maximun execution time of 1000 executions with random 32-bit or 64-bit patterns.

Table 3. Some Execution Examples.

Command	TOS	NOS	Result	Clock periods
SADD	3F800000	3F800000	40000000	58
SSUB	3F800000	3F800000	00000000	56
SMUL	40400000	3FC00000	40900000	198
SDIV	40000000	3F800000	3F000000	228
CHSS	3F800000	–	BF800000	10
PTOS	3F800000	–	–	16
POPS	3F800000	–	–	14
XCHS	3F800000	4000000	–	26
CHSD	3FF0000000000000	–	BFF0000000000000	24
PTOD	3FF0000000000000	–	–	40
POPD	3FF0000000000000	–	–	26
CLR	3FF0000000000000	–	–	4
DADD	3FF00000A0000000	8000000000000000	3FF00000A0000000	578
DSUB	3FF00000A0000000	8000000000000000	3FF00000A0000000	578
DMUL	BFF0000000000000	3FF8000000000000	C002000000000000	1748
DDIV	BFF8000000000000	3FF8000000000000	BFF0000000000000	4560

Note: TOS, NOS and Result are in hexadecimal; Clock period is in decimal.

Command Initiation

After properly positioning the required operands in the stack, a command may be issued. The procedure for initiating a command execution is as follows:

1. Establish appropriate command on the DB0-DB7 lines.
2. Establish HIGH on the C/$\overline{D}$ input.
3. Establish LOW on the $\overline{CS}$ input. Whenever $\overline{WR}$ and $\overline{RD}$ inputs are HIGH, the PAUSE output follows the $\overline{CS}$ input. Hence, $\overline{PAUSE}$ will become LOW.
4. Establish LOW on the $\overline{WR}$ input after an appropriate set-up time (see Timing diagrams).
5. Sometime after the HIGH-to-LOW level transition of $\overline{WR}$ input, the $\overline{PAUSE}$ output will become HIGH to acknowledge the write operation. The $\overline{WR}$ input can return to HIGH anytime after $\overline{PAUSE}$ goes HIGH. The DB0-DB7, C/$\overline{D}$ and $\overline{CS}$ inputs are allowed to change after the hold time requirements are satisfied (see Timing diagram).

An attempt to issue a new command while the current command execution is in progress is allowed. Under these circumstances, the $\overline{PAUSE}$ output will not go HIGH until the current command execution is completed.

Operand Entry

The Am9512 commands operate on the operands located at the TOS and NOS, and results are returned to the stack at NOS and then popped to TOS. The operands required for the Am9512 are one of two formats – single precision floating-point (4 bytes) or double precision floating-point (8 bytes). The result of an operation has the same format as the operands. In other words, operations using single precision quantities always result in a single precision result, while operations involving double precision quantities will result in double precision result.

Operands are always entered into the stack least significant byte first and most significant byte last. The following procedure must be followed to enter operands into the stack:

1. The lower significant operand byte is established on the DB0-DB7 lines.
2. A LOW is established on the C/$\overline{D}$ input to specify that data is to be entered into the stack.
3. The $\overline{CS}$ input is made LOW. Whenever the $\overline{WR}$ and $\overline{RD}$ inputs are HIGH, the $\overline{PAUSE}$ output will follow the $\overline{CS}$ input. Thus $\overline{PAUSE}$ output will become LOW.
4. After appropriate set-up time (see Timing diagrams), the $\overline{WR}$ input is made LOW.
5. After this event, $\overline{PAUSE}$ will return HIGH to indicate that the write operation has been acknowledged.
6. Anytime after the $\overline{PAUSE}$ output goes HIGH, the $\overline{WR}$ input can be made HIGH. The DB0-DB7, C/$\overline{D}$ and $\overline{CS}$ inputs can change after appropriate hold time requirements are satisfied (see Timing diagrams).

The above procedure must be repeated until all bytes of the operand are pushed onto the stack. It should be noted that for single precision operands 4 bytes should be pushed and 8 bytes must be pushed for double precision. Not pushing all the bytes of a quantity will result in byte pointer misalignment.

The Am9512 stack can accomodate 4 single precision quantities or 2 double precision quantities. Pushing more quantities than the capacity of the stack will result in loss of data, which is usual with any LIFO stack.

Removing the Results

Result from an operation will be available at the TOS. Results can be transferred from the stack to the data bus by reading the stack. When the stack is popped for results, the most significant byte is available first and the least significant byte last. A result is always of the same precision as the operands that produced it. Thus, when the result is taken from the stack, the total number of bytes popped out should be appropriate with the precision – single precision results are 4 bytes and double precision results are 8 bytes. The following procedure must be used for reading the result from the stack:

1. A LOW is established on the C/$\overline{D}$ input.
2. The $\overline{CS}$ input is made LOW. When $\overline{WR}$ and $\overline{RD}$ inputs are both HIGH, the $\overline{PAUSE}$ output follows the $\overline{CS}$ input, thus $\overline{PAUSE}$ will be LOW.
3. After appropriate set-up time (see Timing diagrams), the $\overline{RD}$ input is made LOW.
4. Sometime after this, $\overline{PAUSE}$ will return HIGH, indicating that the data is available on the DB0-DB7 lines. This data will remain on the DB0-DB7 lines. This data will remain on the DB0-DB7 lines as long as the $\overline{RD}$ input remains LOW.
5. Anytime after $\overline{PAUSE}$ goes HIGH, the $\overline{RD}$ input can return HIGH to complete transaction.
6. The $\overline{CS}$ and C/$\overline{D}$ inputs can change after appropriate hold time requirements are satisfied (see Timing diagram).
7. Repeat this procedure until all bytes appropriate for the precision of the result are popped out.

Reading of the stack does not alter its data; it only adjusts the byte pointer. If more data is popped than the capacity of the stack, the internal byte pointer will wrap around and older data will be read again, consistent with the LIFO stack.

Reading Status Register

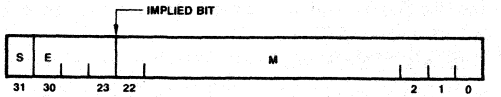
The Am9512 status register can be read without any regard to whether a command is in progress or not. The only implication that has to be considered is the effect this might have on the END and ERR outputs discussed in the signal descriptions.

The following procedure must be followed to accomplish status register reading:

1. Establish HIGH on the C/$\overline{D}$ input.
2. Establish LOW on the $\overline{CS}$ input. Whenever $\overline{WR}$ and $\overline{RD}$ inputs are HIGH, $\overline{PAUSE}$ will follow the $\overline{CS}$ input. Thus, $\overline{PAUSE}$ will go LOW.
3. After appropriate set-up time (see Timing diagram) $\overline{RD}$ is made LOW.
4. Sometime after the HIGH-to-LOW transition of $\overline{RD}$, $\overline{PAUSE}$ will become HIGH, indicating that status register contents are available on the DB0-DB7 lines. These lines will contain this information as long as $\overline{RD}$ is LOW.
5. The $\overline{RD}$ input can be returned HIGH anytime after $\overline{PAUSE}$ goes HIGH.
6. The C/$\overline{D}$ input and $\overline{CS}$ input can change after satisfying approprite hold time requirements (see Timing diagram).

Data Formats

The Am9512 handles floating-point quantities in two different formats – single precision and double precision. The single precision quantities are 32-bits long as shown below.



DF001340

Bit 31:

S = Sign of the mantissa. 1 represents negative and 0 represents positive.

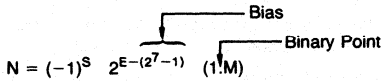
Bits 23-30

E = These 8-bits represent a biased exponent. The bias is $2^7 - 1 = 127$.

Bits 0-22

M = 23-bit mantissa. Together with the sign bit, the mantissa represents a signed fraction in sign-magitude notation. There is an implied 1 beyond the most significant bit (bit 22) of the mantissa. In other words, the mantissa is assumed to be a 24-bit normalized quantity, and the most significant bit, which will always be 1 due to normalization, is implied. The Am9512 restores this implied bit internally before performing arithmetic, normalizes the result, and strips the implied bit before returning the results to the external data bus. The binary point is between the implied bit and bit 22 of the mantissa.

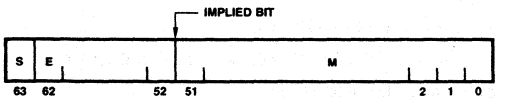
The quantity N represented by the above notation is:



TB000083

Provided E ≠ 0 or all 1's.

A double precision quantity consists of the mantissa sign bit(s), an 11-bit biased exponent (E), and a 52-bit mantissa (M). The bias for double precision quantities is $2^{10} - 1$. The double precision format is illustrated below.



DF001360

Bit 63:

S = Sign of the mantissa. 1 represents negative and 0 represents positive.

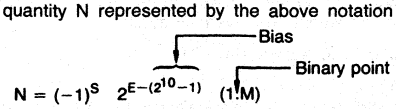
Bits 52-62

E = These 11 bits represent a biased exponent. The bias is $2^{10} - 1 = 1023$.

Bit 0-51

M = 52-bit mantissa. Together with the sign bit, the mantissa represents a signed fraction in sign-magnitude notation. There is an implied 1 beyond the most significant bit (bit 51) of the mantissa. In other words, the mantissa is assumed to be a 53-bit normalized quantity, and the most significant bit, which will always be a 1 due to normalization, is implied. The Am9512 restores this implied bit internally before performing arithmetic, normalizes the result, and strips the implied bit before returning the result to the external data bus. The binary point is between the implied bit and bit 51 of the mantissa.

The quantity N represented by the above notation is:

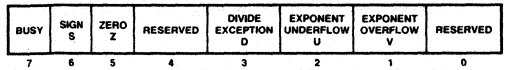


TB000009

Provided E ≠ 0 or all 1's.

Status Register

The Am9512 contains an 8-bit status register with the following format.



DF001350

Bit 0 and bit 4 are reserved. Occurrence of exponent overflow (V), exponent underflow (U) and divide exception (D) are indicated by bits 1, 2 and 3 respectively. An attempt to divide by zero is the only divide exception. Bits 5 and 6 represent a zero result and the sign of a result respectively. Bit 7 (Busy) of the status register indicates if the Am9512 is currently busy executing a command. All the bits are initialized to zero upon reset. Also, executing a CLR (Clear Satus) command will result in all zero status register bits. A zero in Bit 7 indicates that the Am9512 is not busy and a new command may be initiated. As soon as a new command is issued, Bit 7 becomes 1 to indicate the device is busy and remains 1 until the command execution is complete – at which time it will become 0. As soon as a new command is issued, status register bits 0, 1, 2, 3, 4, 5 and 6 are cleared to zero. The status bits will be set as required during the command execution. Hence, as long as bit 7 is 1, the remainder of the status register bit indications should not be relied upon unless the ERR occurs. The following is a detailed bit description.

Bit 0 Reserved

Bit 1 Exponent Overflow (V): When 1, this bit indicates that exponent overflow has occurred. Cleared to zero otherwise.

Bit 2 Exponent Underflow (U): When 1, this bit indicates that exponent underflow has occurred. Cleared to zero otherwise.

Bit 3 Divide Exception (D): When 1, this bit indicates that an attempt to divide by zero is made. Cleared to zero otherwise.

Bit 4 Reserved

Bit 5 Zero (Z): When 1, this bit indicates that the result returned to TOS after a command is all zeros. Cleared to zero otherwise.

Bit 6 Sign (S): When 1, this bit indicates that the result returned to TOS is negative. Cleared to zero otherwise.

Bit 7 Busy: When 1, this bit indicates the Am9512 is in the process of executing a command. It will become zero after the command execution is complete.

All other status register bits are valid when the Busy bit is zero.

Algorithms of Floating-Point Arithmetic

1. Floating-Point to Decimal Conversion

As an introduction to floating-point arithmetic, a brief description of the Decimal equivalent of the Am9512 floating-point format should help the reader to understand and verify the validity of the arithmetic operations. The Am9512 single precision format is used for the following discussions. With a minor modification of the field lengths, the discussion would also apply to the double precision format.

There are three parts in a floating-point number:

a. The sign – the sign applies to the sign of the number. Zero means the number is positive or zero. One means the number is negative.

b. The exponent – the exponent represents the magnitude of the number. The Am9512 single precision format has an excess 127_{10} notation, which means the code representation is 127_{10} higher than the actual value. The following are a few examples of actual versus coded exponent.

Actual	Coded
$+127_{10}$	$+254_{10}$
0	127_{10}
-126_{10}	$+1_{10}$

c. The mantissa – the mantissa is a 23-bit value with the binary point to the left of the most significant bit. There is a hidden 1 to the left of the binary point so the mantissa is always less than 2 and greater than or equal to 1.
To find the Decimal equivalent of the floating-point number, the mantissa is multiplied by 2 to the power of the actual exponent. The number is negated if the sign bit = 1. The following are two examples of conversion:

Example 1

Floating Point No. = 0 10000011 11000000000000000000000B
Sign — Exponent — Mantissa

TB000010

Coded Exponent = 1 0 0 0 0 0 1 1 B
Actual Exponent = 1 0 0 0 0 0 1 1 B – 0 1 1 1 1 1 1 1 B = 0 0 0 0 0 1 0 0
B = 4_{10}
Mantissa = 1.1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B
= $1 + \frac{1}{2} + \frac{1}{4} = 1.75_{10}$
Decimal No. = $2^4 \times 1.75 = 16 \times 1.75 = 28_{10}$

Example 2

Floating Point No. = 1 01111010 01100000000000000000000B
Sign — Exponent — Mantissa

TB000011

Coded Exponent = 0 1 1 1 1 0 1 0 B
Actual Exponent = 0 1 1 1 1 0 1 0 B – 0 1 1 1 1 1 1 1 B = 1 1 1 1 0 1 1 1
B = -5_{10}
Mantissa = 1.0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B
= $1 + \frac{1}{4} + \frac{1}{8} = 1.375_{10}$
Decimal No. = $-2^{-5} \times 1.375 = -.04296875_{10}$

2. Unpacking of the Floating-Point Numbers
The Am9512 unpacks the floating-point number into three parts before any of the arithmetic operation. The number is divided into three parts as described in Section 1. The sign and expondent are copied from the original number as 1-bit and 8-bit numbers, respectively. The mantissa is stored as a 24-bit number. The least significant 23 bits are copied from the original number and the MSB is set to 1. The binary point is asumed to the right of the MSB.

The abbreviations listed below are used in the following sections of algorithm description:

SIGN – Sign of Result
EXP – Exponent of Result
MAN – Mantissa of Result
SIGN (TOS) – Sign of Top of Stack
EXP (TOS) – Exponent of Top of Stack
MAN (TOS) – Mantissa of Top of Stack
SIGN (NOS) – Sign of Next on Stack
EXP (NOS) – Exponent of Next on Stack
MAN (NOS) – Mantissa of Next on Stack

3. Floating-Point Add/Subtract
The floating-point add and subtract essentially use the same algorithm. The only difference is that floating-point subtract changes the sign of the floating-point number at top of stack and then performs the floating-point add.

The following is a step-by-step description of a floating-point add algorithm (Figure 1):

a. Unpack TOS and NOS.

b. The exponent of TOS is compared to the exponent of NOS.

c. If the exponents are equal, go to step f.

d. Right shift the mantissa of the number with the smaller exponent.

e. Increment the smaller exponent and go to step b.

f. Set sign of result to sign of larger number.

g. Set exponent of result to exponent of larger number.

h. If sign of the two numbers are not equal, go to m.

i. Add Mantissas.

j. Right shift resultant mantissa by 1 and increment exponent of result by 1.

k. If MSB of exponent changes from 1 to 0 as a result of the increment, set overflow status.

l. Round if necessary and exit.

m. Subtract smaller mantissa from larger mantissa.

n. Left shift mantissa and decrement exponent of result.

o. If MSB of exponent changes from 0 to 1 as a result of the decrement, set underflow status and exit.

p. If the MSB of the resultant mantissa = 0, go to n.

q. Round if necessary and exit.

4. Floating-Point Multiply
Floating-point multiply basically involves the addition of the exponents and multiplication of the mantissas. The following is a step-by-step description of a floating multiplication algorithm (Figure 2):

a. Check if TOS or NOS = 0.

b. If either TOS or NOS = 0, set result to 0 and exit.

c. Unpack TOS and NOS.

d. Convert EXP (TOS) and EXP (NOS) to unbiased form.
EXP (TOS) = EXP (TOS) – 127_{10}
EXP (NOS) = EXP (NOS) – 127_{10}

e. Add exponents.
EXP = EXP (TOS) + EXP (NOS)

f. If MSB of EXP (TOS) = MSB of EXP (NOS) = 0 and MSB of EXP = 1, then set overflow status and exit.

g. If MSB of EXP (TOS) = MSB of EXP (NOS) = 1 and MSB of EXP = 0, then set underflow status and exit.

h. Convert Exponent back to biased form.
EXP = EXP + 127_{10}

i. If sign of TOS = sign of NOS, set sign of result to 0, else set sign of result to 1.

j. Multiply mantissa.

k. If MSB of resultant = 1, right shift mantissa by 1 and increment exponent of resultant.

l. If MSB of exponent changes from 1 to 0 as a result of the increment, set overflow status.

m. Round if necessary and exit.

5. Floating-Point Divide
The floating-point divide basically involves the subtraction of exponents and the division of mantissas. The following is a step-by-step description of a division algorithm (Figure 3).

a. If TOS = 0, set divide exception error and exit.

b. If NOS = 0, set result to 0 and exit.

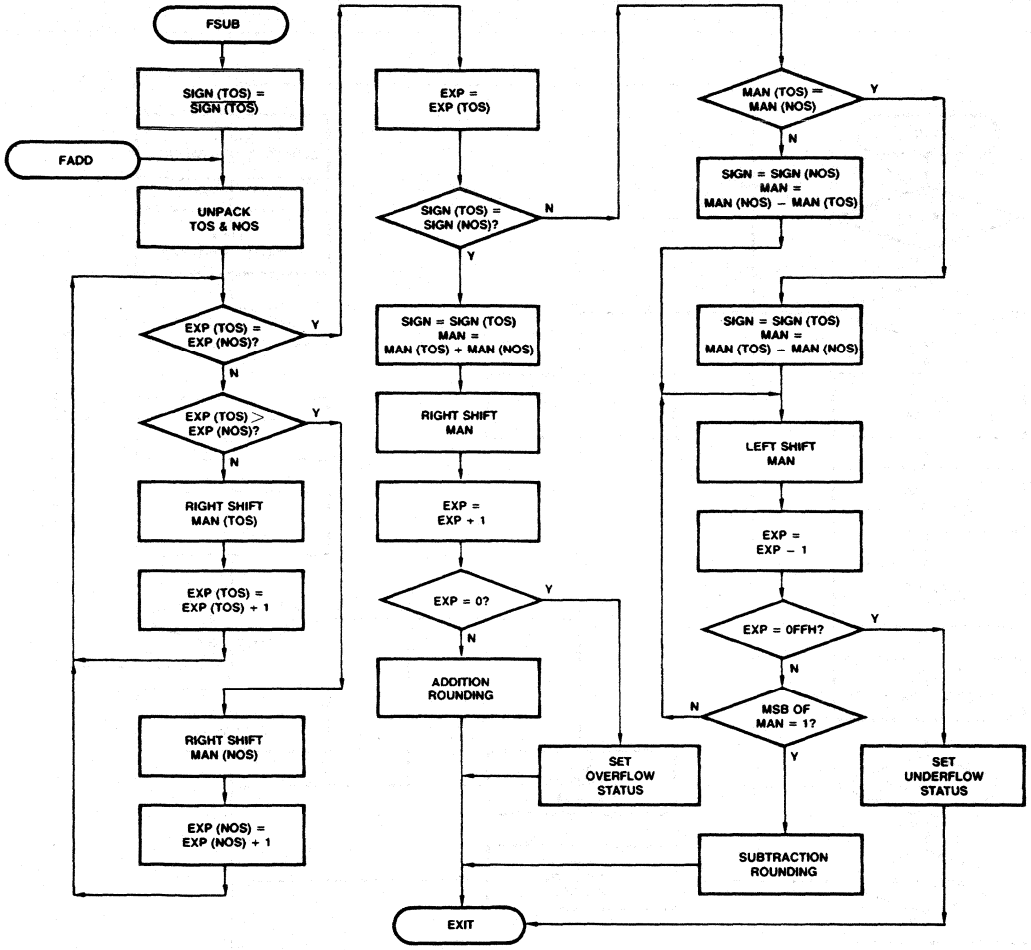
c. Unpack TOS and NOS.

d. Convert EXP (TOS) and EXP (NOS) to unbiased form.
EXP (TOS) = EXP(TOS) – 127_{10}
EXP (NOS) = EXP (NOS) – 127_{10}

e. Subtract exponent of TOS from exponent of NOS.
EXP = EXP (NOS) – EXP (TOS)

f. If MSB of EXP (NOS) = 0, MSB of EXP (TOS) = 1 and MSB of EXP = 1, then set overflow status and exit.

g. If MSB of EXP (NOS) = 1, MSB of EXP (TOS) = 0, and MSB of EXP = 0, then set underflow status and exit.



PF001201

Figure 1. Conceptual Floating-Point Addition/Subtraction

h. Add bias to exponent of result.
EXP = EXP + 127_{10}

i. If sign of TOS = sign of NOS, set sign of result to 0, else set sign of result to 1.

j. Divide mantissa of NOS by mantissa of TOS.

k. If MSB = 0, left shift mantissa and decrement exponent of resultant, else go to n.

l. If MSB of exponent changes from 0 to 1 as a result of the decrement, set underflow status.

m. Go to k.

n. Round if necessary and exit.

The algorithms described above provide the user a means of verifying the validity of the result. They do not necessarily reflect the exact internal sequence of the Am9512.

6. Rounding

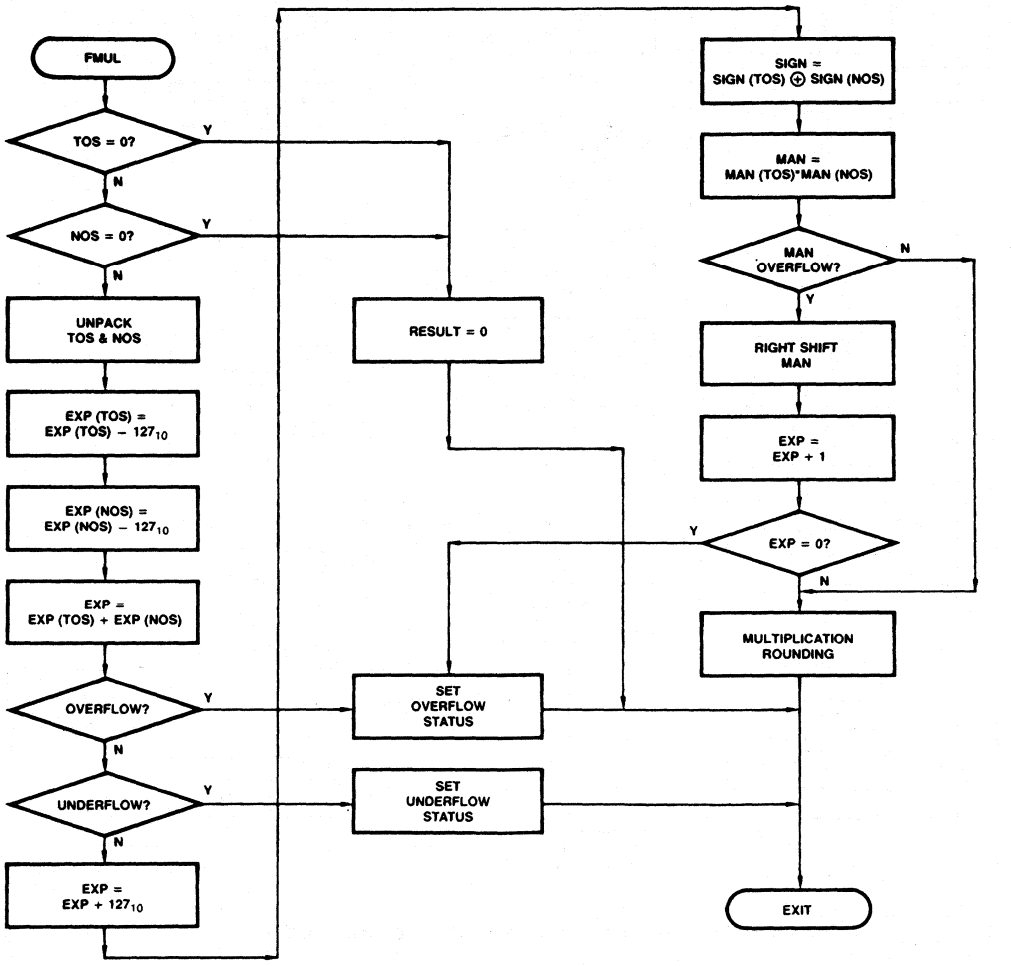
The Am9512 adopts a rounding algorithm that is consistent with the Intel® standard for floating-point arithmetic. The following description is an excerpt from the paper published in the proceedings of Compsac 77, November 1977, pp. 107-112 by Dr. John F. Palmer of Intel Corporation.

The method used for doing the rounding during floating-point arithmetic is known as "Round to Even"; i.e., if the resultant number is exactly halfway between two floating-point numbers, the number is rounded to the nearest floating-point number whose LSB of the mantissa is 0. To simplify the explanation, the algorithms will be illustrated with 4-bit arithmetic. The existence of an accumulator will be assumed as shown:

OF	B1	B2	B3	B4	G	R	ST

The bit labels denote:

OF – The overflow bit
B1-B4 – The 4 mantissa bits
G – The Guard bit
R – The Rounding bit
ST – The "Sticky" bit



PF001210

Figure 2. Conceptual Floating-Point Multiplication

The Sticky bit is set to one if any ones are shifted right of the rounding bit in the process of denormalization. If the Sticky bit becomes set, it remains set throughtout the operation. All shifting in the Accumulator involves the OF, G, R and ST bits. The ST bit is not affected by left shifts, but zeros are introduced into OF by right shifts.

Rounding during addition of magnitudes – add 1 to the G position, then if G = R = ST = 0, set B4 to 0 ("Rounding to Even").

Rounding during subtraction of magnitudes – if more than one left shift was performed, no rounding is needed; otherwise, round the same way as addition of magnitudes.

Rounding during multiplication – let the normalized double length product be:

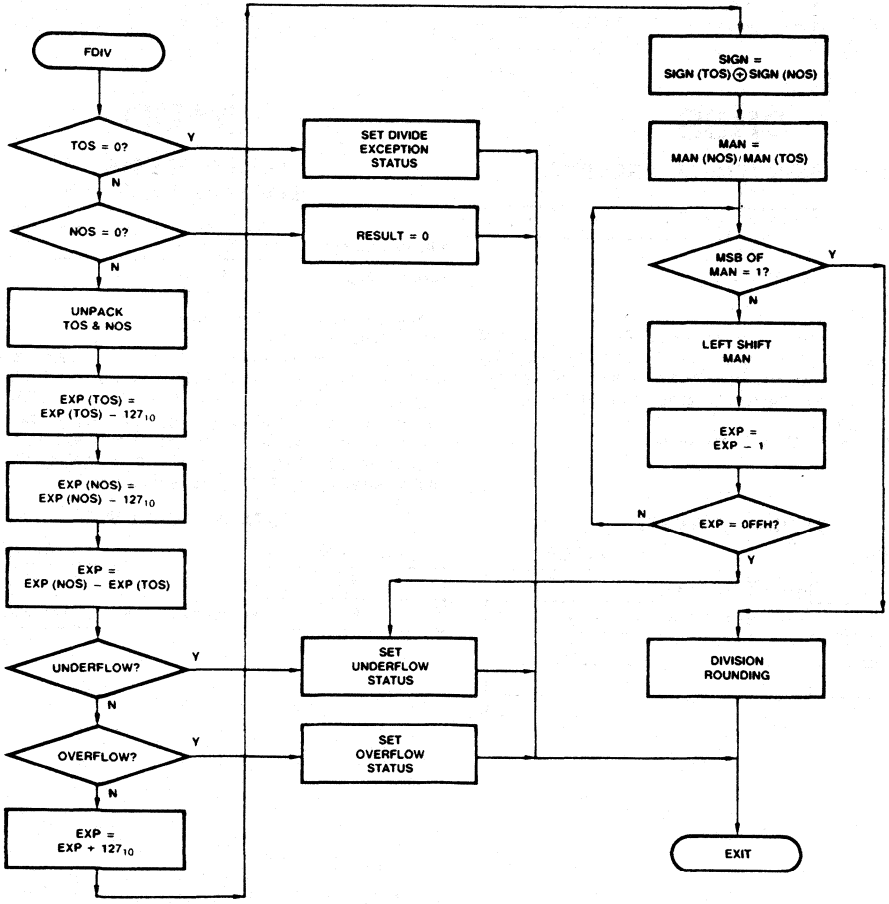
B1	B2	B3	B4	B5	B6	B7	B8

Then G = B5, R = B6, ST = B7 V B8. The rounding is then performed as in addition of magnitudes.

Rounding during division – let the first six bits of the normalized quotient be:

B1	B2	B3	B4	B5	B6

Then G = B5, R = B6, ST = 0 if and only if remainder = 0. The rounding is then performed as in addition of magnitudes.



PF001220

Figure 3. Conceptual Floating-Point Division

CHSD

CHANGE SIGN DOUBLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	1	0	1

Hex Coding: AD IF SRE = 1
2D IF SRE = 0

Execution Time: See Table 2

Description:
The sign of the double precision TOS operand A is complemented. The double precision result R is returned to TOS. If the double precision operand A is zero, then the sign is not affected. The status bits S and Z indicate the sign of the result and if the result is zero. The status bits U, V and D are always cleared to zero.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS

TB000067

CLR

CLEAR STATUS

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	0	0

Hex Coding: 80 IF SRE = 1
00 IF SRE = 0

Execution Time: 4 clock cycles

Description:
The status bits S, Z, D, U, V are cleared to zero. The stack is not affected. This essentially is a no operation command as far as operands are concerned.

Status Affected: S, Z, D, U, V always zero.

CHSS

CHANGE SIGN SINGLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	1	0	1

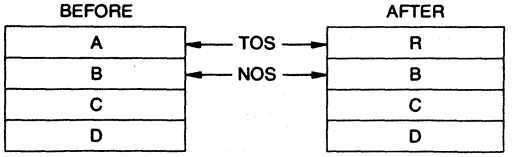
Hex Coding: 85 IF SRE = 1
05 IF SRE = 0

Execution Time: See Table 2

Description:
The sign of the single precision operand A at TOS is complemented. The single precision result R is returned to TOS. If the exponent field of A is zero, all bits of R will be zeros. The status bits S and Z indicate the sign of the result and if the result is zero. The status bits U, V and D are cleared to zero.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS



TB000068

DADD

DOUBLE PRECISION FLOATING-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	0	0	1

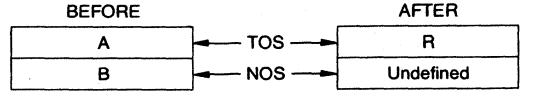
Hex Coding: A9 IF SRE = 1
29 IF SRE = 0

Execution Time: See Table 2

Description:
The double precision operand A from TOS is added to the double precision operand B from NOS. The result is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow, and exponent overflow, respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS



TB000069

DSUB

DOUBLE PRECISION FLOATING-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	0	1	0

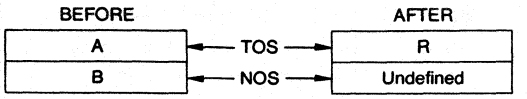
Hex Coding: AA IF SRE = 1
2A IF SRE = 0

Execution Time: See Table 2

Description:
The double precision operand A at TOS is subtracted from the double precision operand B at NOS. The result is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow, respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS



TB000069

DDIV

DOUBLE PRECISION FLOATING-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	1	0	0

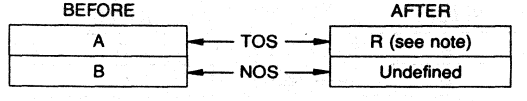
Hex Coding: AC IF SRE = 1
2C IF SRE = 0

Execution Time: See Table 2

Description:
The double precision operand B from NOS is divided by the double precision operand A from TOS. The result (quotient) is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, attempt to divide by zero, exponent underflow and exponent overflow, respectively.

Status Affected: S, Z, D, U, V.

STACK CONTENT



TB000072

Note: If A is zero, the R = B (Divide exception).

DMUL

DOUBLE PRECISION FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	1	0	1	0	1	1

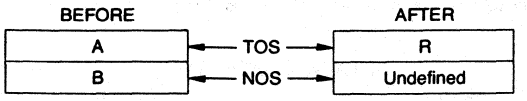
Hex Coding: AB IF SRE = 1
2B IF SRE = 0

Execution Time: See Table 2

Description:
The double precision operand A from TOS is multiplied by the double precision operand B from NOS. The result is rounded to obtain the final double precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow, respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS



TB000069

SADD

SINGLE PRECISION FLOATING-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	0	1

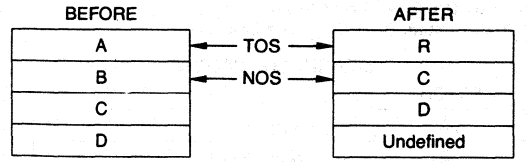
Hex Coding: 81 IF SRE = 1
01 IF SRE = 0

Execution Time: See Table 2

Description:
The single precision operand A from TOS is added to the single precision operand B from NOS. The result is rounded to obtain the final single precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow, respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENT



TB000074

SSUB

SINGLE PRECISION FLOATING-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	1	0

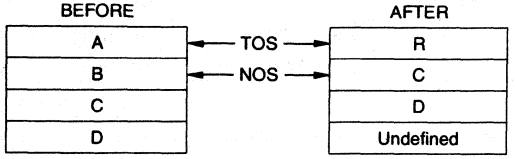
Hex Coding: 82 IF SRE = 1
02 IF SRE = 0

Execution Time: See Table 2

Description:
The single precision operand A at TOS is subtracted from the single precision operand B at NOS. The result is rounded to obtain the final single precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow, respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS



TB000073

SDIV

SINGLE PRECISION FLOATING-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	1	0	0

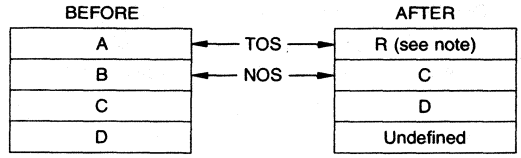
Hex Coding: 84 IF SRE = 1
04 IF SRE = 0

Execution Time: See Table 2

Description:
The single precision operand B from NOS is divided by the single precision operand A from TOS. The result (quotient) is rounded to obtain the final result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, attempt to divide by zero, exponent underflow and exponent overflow, respectively.

Status Affected: S, Z, D, U, V

STACK CONTENTS



TB000076

Note: If exponent field of A is zero, then R = B (Divide exception).

SMUL

SINGLE PRECISION FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	0	1	1

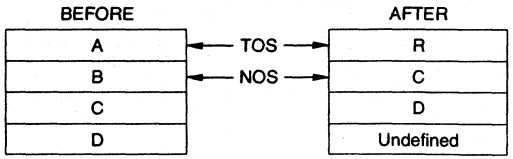
Hex Coding: 83 IF SRE = 1
03 IF SRE = 0

Execution Time: See Table 2

Description:
The single precision operand A from TOS is multiplied by the single precision operand B from NOS. The result is rounded to obtain the final single precision result R which is returned to TOS. The status bits S, Z, U and V are affected to report the sign of the result, if the result is zero, exponent underflow and exponent overflow, respectively. The status bit D will be cleared to zero.

Status Affected: S, Z, U, V. (D always zero.)

STACK CONTENTS



TB000075

POPS

POP STACK SINGLE PRECISION

	7	6	5	4	3	2	1	0
Binary Coding:	SRE	0	0	0	0	1	1	1

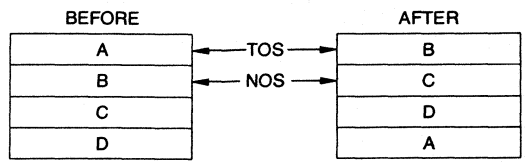
Hex Coding: 87 IF SRE = 1
07 IF SRE = 0

Execution Time: See Table 2

Description:
The single precision operand A is popped from the stack. The internal stack control mechanism is such that A will be written at the bottom of the stack. The status bits S and Z are affected to report the sign of the new operand at TOS and if it is zero, respectively. The status bits U, V and D will be cleared to zero. Note that only the exponent field of the new TOS is checked for zero; if it is zero, status bit Z will set to 1.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS



TB000078

PTOD

PUSH STACK DOUBLE PRECISION

Binary Coding:

7	6	5	4	3	2	1	0
SRE	0	1	0	1	1	1	0

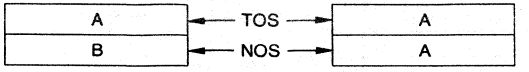
Hex Coding: AE IF SRE = 1
2E IF SRE = 0

Execution Time: See Table 2

Description:
The double precision operand A from the TOS is pushed back onto the stack. This is effectively a duplication of A into two consecutive stack locations. The status bits S and Z are affected to report the sign of the new TOS and if the new TOS is zero, respectively. The status bits U, V and D will be cleared to zero.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS



TB000077

POPD

POP STACK DOUBLE PRECISION

Binary Coding:

7	6	5	4	3	2	1	0
SRE	0	1	0	1	1	1	1

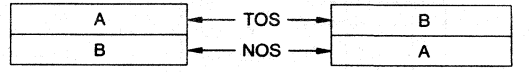
Hex Coding: AF IF SRE = 1
2F IF SRE = 0

Execution Time: See Table 2

Description:
The double precision operand A is popped from the stack. The internal stack control mechanism is such that A will be written at the bottom of the stack. This operation has the same effect as exchanging TOS and NOS. The status bits S and Z are affected to report the sign of the new operand at TOS and if it is zero, respectively. The status bits U, V and D will be cleared to zero.

Status Affected: S, Z. (U, V and D always zero.)

STACK CONTENTS



TB000080

PTOS

PUSH STACK SINGLE PRECISION

Binary Coding:

7	6	5	4	3	2	1	0
SRE	0	0	0	0	1	1	0

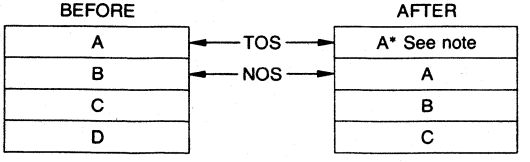
Hex Coding: 86 IF SRE = 1
06 IF SRE = 0

Execution Time: See Table 2

Description:
This instruction effectively pushes the single precision operand from TOS onto the stack. This amounts to duplicating the operand at two locations in the stack. However, if the operand at TOS prior to the PTOS command has only its exponent field as zero, the new content of the TOS will all be zeroes. The contents of NOS will be an exact copy of the old TOS. The status bits S and Z are affected to report the sign of the new TOS and if the content of TOS is zero, respectively. The status bits U, V and D will be cleared to zero.

Status Affected: S, Z. (U, V, D always zero.)

STACK CONTENTS



TB000079

Note: A* = A if Exponent field of A is not zero.
A* = 0 if Exponent field of A is zero.

XCHS

EXCHANGE TOS AND NOS SINGLE-PRECISION

Binary Coding:

7	6	5	4	3	2	1	0
SRE	0	0	0	1	0	0	0

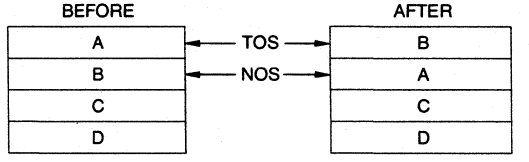
Hex Coding: 88 IF SRE = 1
08 IF SRE = 0

Execution Time: See Table 2

Description:
The single precision operand A from TOS and the single precision operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All other operands are unchanged.

Status Affected: S, Z. (U, V, and D always zero.)

STACK CONTENTS



TB000081

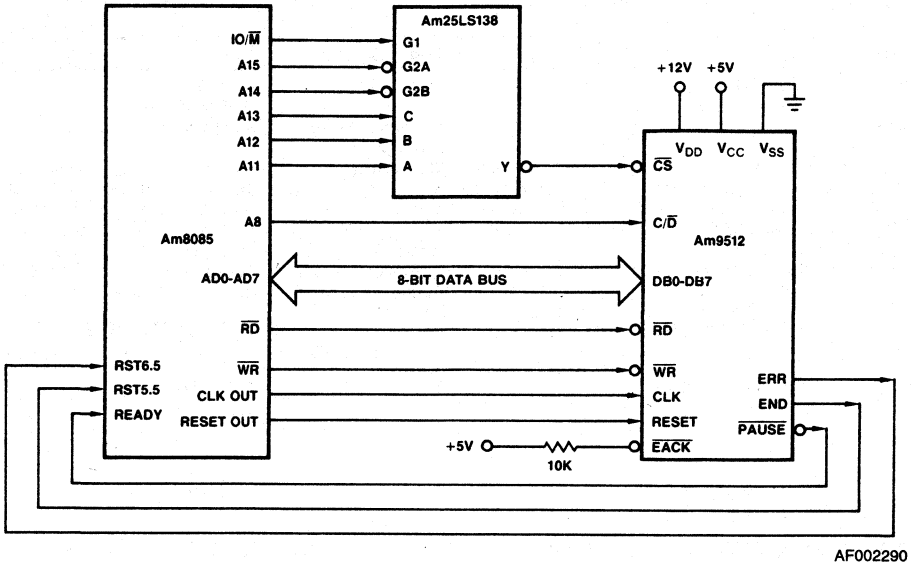


Figure 4. Am9512 to Am8085 Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
VDD with Respect to VSS ..... −0.5 to +15.0V
VCC with Respect to VSS ..... −0.5 to +7.0V
All Signal Voltages
with Respect to VSS ..... −0.5 to +7.0V
Power Dissipitation (Package Limitation) 2.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

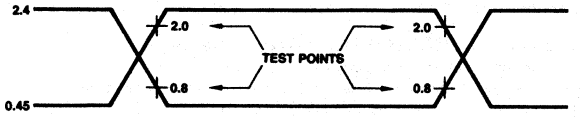
Grade	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5.0V ±5%	0V
Industrial	−40°C to 85°C	5.0V ±10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Typ	Max	Units
VOH	Output HIGH Voltage	IOH = −200μA	3.7			Volts
VOL	Output LOW Voltage	IOL = 3.2mA			0.4	Volts
VIH	Input HIGH Voltage		2.0		VCC	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC			±10	μA
IOZ	Data Bus Leakage	VO = 0.4V			10	μA
		VO = VCC			10	
ICC	VCC Supply Current	T_A = +25°C		50	90	mA
		T_A = 0°C			95	
IDD	VDD Supply Current	T_A = +25°C		50	90	mA
		T_A = 0°C			95	
CO	Output Capacitance	f_C = 1.0MHz, Inputs = 0V		8	10	pF
CI	Input Capacitance			5	8	pF
CIO	I/O Capacitance			10	12	pF

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF004000

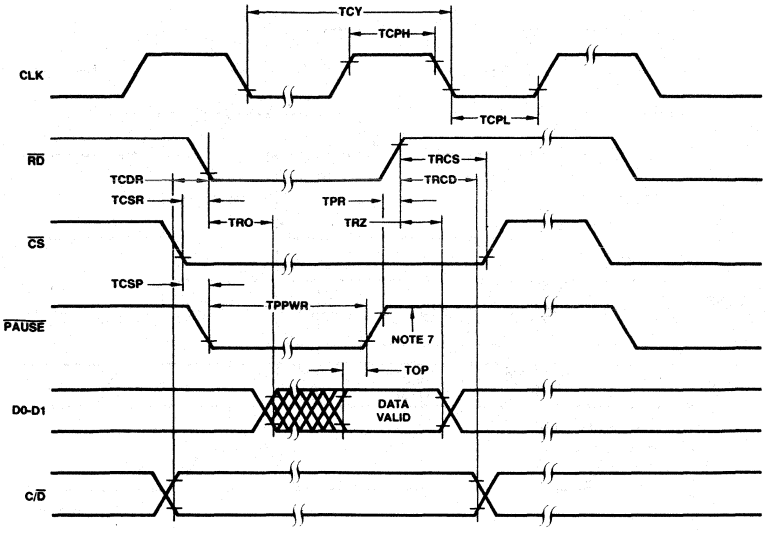
SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Note 1)

Parameters	Description		Am9512 Min	Am9512 Max	Am9512-1 Min	Am9512-1 Max	Units
TAPW	$\overline{EACK}$ LOW Pulse Width		100		75		ns
TCDR	C/$\overline{D}$ to $\overline{RD}$ LOW Set-up Time		0		0		ns
TCDW	C/$\overline{D}$ to $\overline{WR}$ LOW Set-up Time		0		0		ns
TCPH	Clock Pulse HIGH Width		200	500	140	500	ns
TCPL	Clock Pulse LOW Width		240		160		ns
TCSP	$\overline{CS}$ LOW to $\overline{PAUSE}$ LOW Delay (Note 5)		150		100		ns
TCSR	$\overline{CS}$ to $\overline{RD}$ LOW Set-up Time		0		0		ns
TCSW	$\overline{CS}$ LOW to $\overline{WR}$ LOW Set-up Time		0		0		ns
TCY	Clock Period		480	5000	320	2000	ns
TDW	Data Valid to $\overline{WR}$ HIGH Delay		150		100		ns
TEAE	$\overline{EACK}$ LOW to END LOW Delay			200		175	ns
TEHPHR	END HIGH to $\overline{PAUSE}$ HIGH Data Read when Busy			5.5TCY+300		5.5TCY+200	ns
TEHPHW	END HIGH to $\overline{PAUSE}$ HIGH Write when Busy			200		175	ns
TEPW	END HIGH Pulse Width		400		300		ns
TEX	Execution Time		See Table 2				ns
TOP	Data Bus Output Valid to $\overline{PAUSE}$ HIGH Delay		0		0		ns
TPPWR	$\overline{PAUSE}$ LOW Pulse Width Read	Data	3.5TCY+50	5.5TCY+300	3.5TCY+50	5.5TCY+200	ns
		Status	1.5TCY+50	3.5TCY+300	1.5TCY+50	3.5TCY+200	
TPPWRB	END HIGH to $\overline{PAUSE}$ HIGH Read when Busy	Data	See Table 2				ns
		Status	1.5TCY+50	3.5TCY+300	1.5TCY+50	3.5TCY+200	
TPPWW	$\overline{PAUSE}$ LOW Pulse Width Write when Not Busy			TCSW+50		TCSW+50	ns
TPPWWB	$\overline{PAUSE}$ LOW Pulse Width Write when Busy		See Table 2				ns
TPR	$\overline{PAUSE}$ HIGH to Read HIGH Hold Time		0		0		ns
TPW	$\overline{PAUSE}$ HIGH to Write HIGH Hold Time		0		0		ns
TRCD	$\overline{RD}$ HIGH to C/$\overline{D}$ Hold Time		0		0		ns
TRCS	$\overline{RD}$ HIGH to $\overline{CS}$ HIGH Hold Time		0		0		ns
TRO	$\overline{RD}$ LOW to Data Bus On Delay		50		50		ns
TRZ	$\overline{RD}$ HIGH to Data Bus Off Delay		50	200	50	150	ns
TSAPW	SVACK LOW Pulse Width		100		75		ns
TSAR	SVACK LOW to SVREQ LOW Delay			300		200	ns
TWCD	$\overline{WR}$ HIGH to C/D Hold Time		60		30		ns
TWCS	$\overline{WR}$ HIGH to CS HIGH Hold Time		60		30		ns
TWD	$\overline{WR}$ HIGH to Data Bus Hold Time		20		20		ns

Notes: 1. Typical values are for $T_A = 25°C$, nominal supply voltages and nominal processing parameters.
2. Switching parameters are listed in alphabetical order.
3. Test conditions assume transition times of 20ns or less, output loading of one TTL gate plus 100pF and timing reference levels of 0.8V and 2.0V.
4. END HIGH pulse width is specified for $\overline{EACK}$ tied to VSS. Otherwise TEAE applies.
5. $\overline{PAUSE}$ is pulled LOW for both command and data operations.
6. TEX is the execution time of the current command (see the Command Execution Times table).
7. $\overline{PAUSE}$ will go LOW at this point if $\overline{CS}$ is LOW and $\overline{RD}$ and $\overline{WR}$ are HIGH.

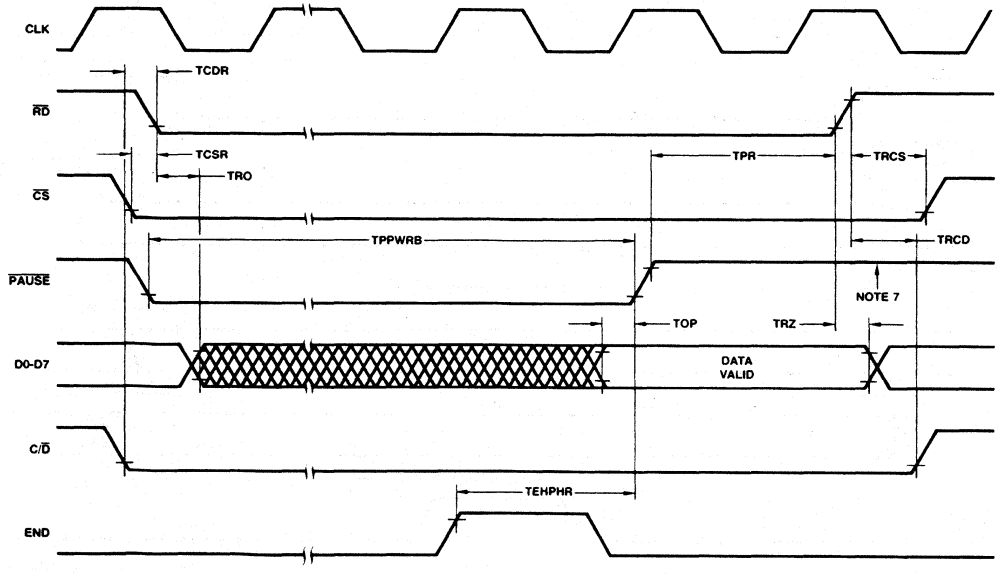
TIMING DIAGRAMS

READ OPERATION



WF004010

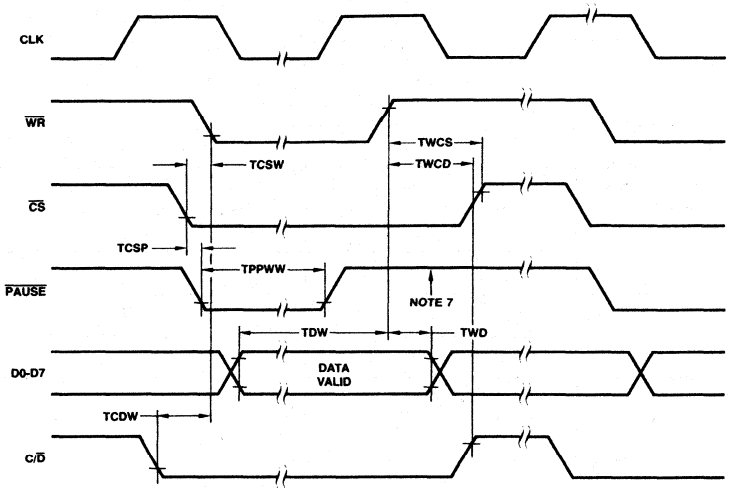
OPERAND READ WHEN Am9512 IS BUSY



WF004020

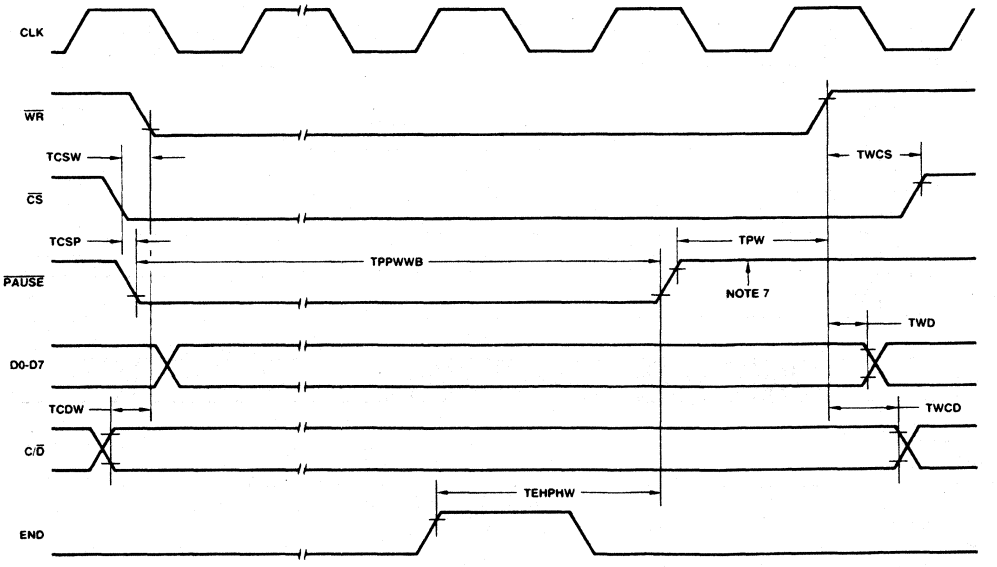
TIMING DIAGRAMS (Cont.)

OPERAND ENTRY



WF004030

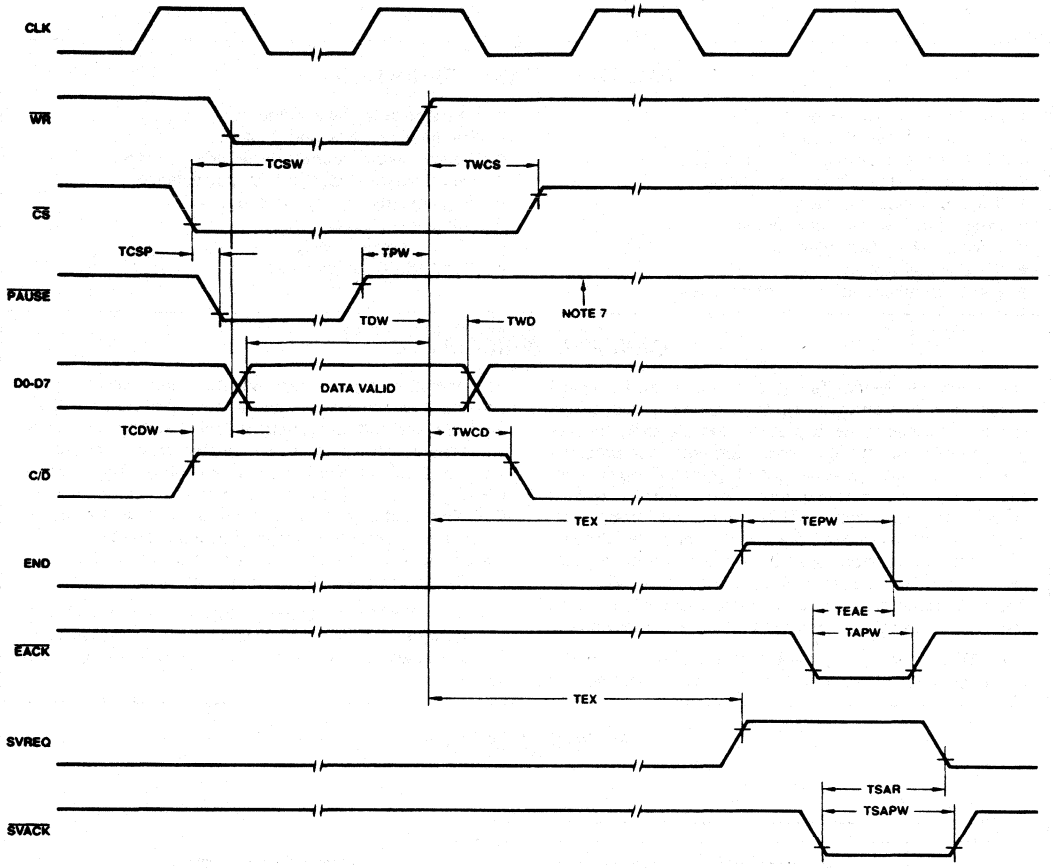
COMMAND OR DATA WRITE WHEN Am9512 IS BUSY



WF004040

TIMING DIAGRAMS (Cont.)

COMMAND INITIATION



WF004050

Am9513A/AmZ8073A

System Timing Controller

DISTINCTIVE CHARACTERISTICS

- Five independent 16-bit counters
- High speed counting rates
- Up/down and binary/BCD counting
- Internal oscillator frequency source
- Tapped frequency scaler
- Programmable frequency output
- 8-bit or 16-bit bus interface
- Time-of-day option
- Alarm comparators on counters 1 and 2
- Complex duty cycle outputs
- One-shot or continuous outputs
- Programmable count/gate source selection
- Programmable input and output polarities
- Programmable gating functions
- Retriggering capability
- +5 volt power supply
- Standard 40-pin package

GENERAL DESCRIPTION

The Am9513A System Timing Controller is an LSI circuit designed to service many types of counting, sequencing and timing applications. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital one-shots, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis, etc. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide pulses or levels and can be active-high or active-low. The counters can be programmed to count up or down in either binary or BCD. The host processor may read an accumulated count at any time without disturbing the counting process. Any of the counters may be internally concatenated to form any effective counter length up to 80 bits.

The AmZ8073A* is functionally equivalent to the Am9513A with timing enhancements which allow it to be fully speed compatible with the AmZ8001 and AmZ8002 microprocessors.

BLOCK DIAGRAM

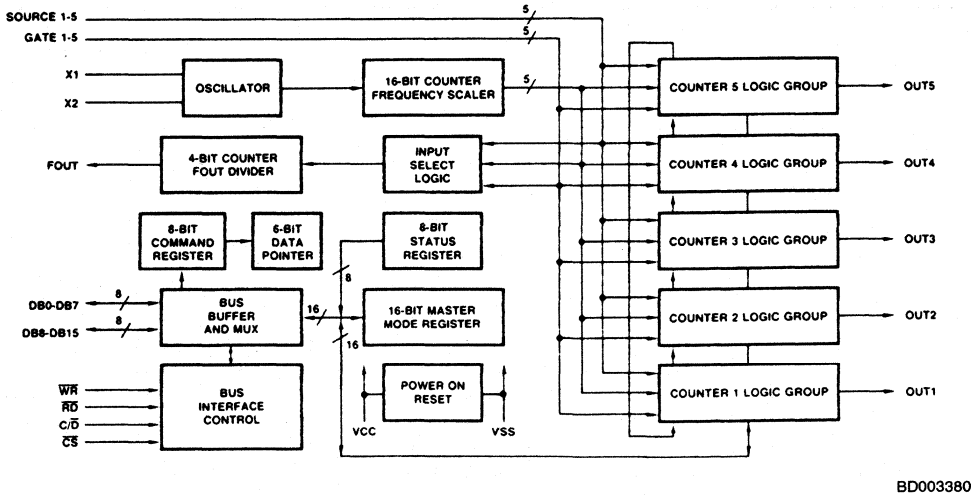


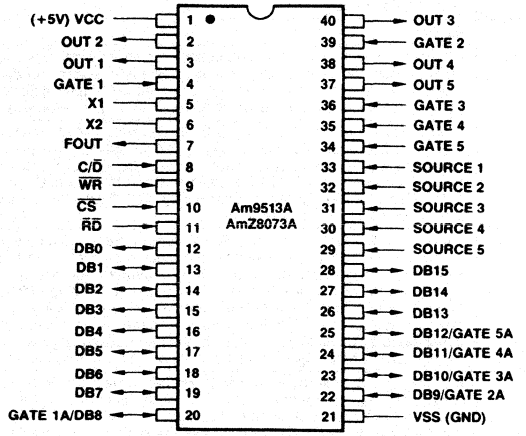
Figure 1-1.

*Z8000 is a trademark of Zilog, Inc.

01731D

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

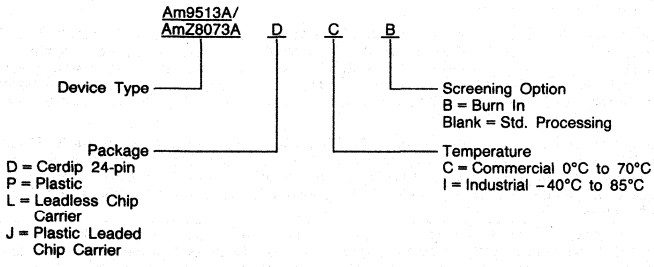


CD005210

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9513A/ AmZ8073A	DC, DCB, DI, DIB, PC, PCB, PI, PIB, LC, LCB, LI, LIB, /BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	V_{CC}		+5V Power Supply.
21	V_{SS}		Ground.
5, 6	X1, X2	O, I	(Crystal). X1 and X2 are the connections for an external crystal used to determine the frequency of the internal oscillator. The crystal should be a parallel-resonant, fundamental-mode type. An RC or LC or other reactive network may be used instead of a crystal. For driving from an external frequency source, X1 should be left open and X2 should be connected to a TTL source and a pull-up resistor.
7	FOUT	O	(Frequency Out). The FOUT output is derived from a 4-bit counter that may be programmed to divide its input by any integer value from 1 through 16 inclusive. The input to the counter is selected from any of 15 sources, including the internal scaled oscillator frequencies. FOUT may be gated on and off under software control and when off will exhibit a low impedance to ground. Control over the various FOUT options resides in the Master Mode register. After power-up, FOUT provides a frequency that is 1/16 that of the oscillator. The input source on power-up is F1.
4, 39, 36-34	GATE1 - GATE5	I	(Gate). The Gate inputs may be used to control the operations of individual counters by determining when counting may proceed. The same Gate input may control up to three counters. Gate pins may also be selected as count sources for any of the counters and for the FOUT divider. The active polarity for a selected Gate input is programmed at each counter. Gating function options allow level-sensitive gating or edge-initiated gating. Other gating modes are available including one that allows the Gate input to select between two counter output frequencies. All gating functions may also be disabled. The active Gate input is conditioned by an auxiliary input when the unit is operating with an external 8-bit data bus. See Data Bus description. Schmitt-trigger circuitry on the GATE inputs allows slow transition times to be used.
33-29	SRC1 - SRC5	I	(Source).The Source inputs provide external signals that may be counted by any of the counters. Any Source line may be routed to any or all of the counters and the FOUT divider. The active polarity for a selected SRC input is programmed at each counter. Any duty cycle waveform will be accepted as long as the minimum pulse width is at least half the period of the maximum specified counting frequency for the part. Schmitt-trigger circuitry on the SRC inputs allows slow transition times to be used.
3, 2, 40, 38, 37	OUT1 - OUT5	O	(Counter). Each 3-state OUT signal is directly associated with a corresponding individual counter. Depending on the counter configuration, the OUT signal may be a pulse, a square wave, or a complex duty cycle waveform. OUT pulse polarities are individually programmable. The output circuitry detects the counter state that would have been all bits zero in the absence of a reinitialization. That information is used to generate the selected waveform type. An optional output mode for Counters 1 and 2 overrides the normal output mode and provides a true OUT signal when the counter contents match the contents of an Alarm register.
12-19, 20, 22-28	DB0 - DB7, DB8 - DB15	I/O	(Data Bus). The 16 bidirectional Data Bus lines are used for information exchanges with the host processor. HIGH on a Data Bus line corresponds to one and LOW corresponds to zero. These lines act as inputs when $\overline{WR}$ and $\overline{CS}$ are active and as outputs when $\overline{RD}$ and $\overline{CS}$ are active. When $\overline{CS}$ is inactive, these pins are placed in a high-impedance state. After power-up or reset, the data bus will be configured for 8-bit width and will use only DB0 through DB7. DB0 is the least significant and DB7 is the most significant bit position. The data bus may be reconfigured for 16-bit width by changing a control bit in the Master Mode register. This is accomplished by writing an 8-bit command into the low-order DB lines while holding the DB13 - DB15 lines at a logic high level. Thereafter, all 16 lines can be used, with DB0 as the least significant and DB15 as the most significant bit position. When operating in the 8-bit data bus environment, DB8 - DB15 will never be driven active by the Am9513A. DB8 through DB12 may optionally be used as additional Gate inputs (see Figure 1-3). If unused, they should be held HIGH. When pulled LOW, a GATENA signal will disable the action of the corresponding counter N gating. DB13 - DB15 should be held HIGH in 8-bit bus mode whenever $\overline{CS}$ and $\overline{WR}$ are simultaneously active.
10	$\overline{CS}$	I	(Chip Select). The active-low Chip Select input enables Read and Write operations on the data bus. When Chip Select is HIGH, the Read and Write inputs are ignored. The first Chip Select signal after power-up is used to clear the power-on reset circuitry. If Chip Select is tied to ground permanently, the power-on reset circuitry may not function. In such a configuration, the software reset command must be issued following power-up to reset the Am9513A.
11	$\overline{RD}$	I	(Read). The active-low Read signal is conditioned by Chip Select and indicates that internal information is to be transferred to the data bus. The source will be determined by the port being addressed and, for Data Port reads, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
9	$\overline{WR}$	I	(Write). The active-low Write signal is conditioned by Chip Select and indicates that data bus information is to be transferred to an internal location. The destination will be determined by the port being addressed and, for Data Port writes, by the contents of the Data Pointer register. $\overline{WR}$ and $\overline{RD}$ should be mutually exclusive.
8	C/$\overline{D}$	I	(Control/Data). The Control/Data signal selects source and destination locations for Read and Write operations on the data bus. Control Write operations load the Command register and the Data Pointer. Control Read operations output the Status register. Data Read and Data Write transfers communicate with all other internal registers. Indirect addressing at the data port is controlled internally by the Data Pointer register.

Signal	Abbreviation	Type	Pins
+ 5 Volts	VCC	Power	1
Ground	VSS	Power	1
Crystal	X1, X2	O, I	2
Read	$\overline{RD}$	Input	1
Write	$\overline{WR}$	Input	1
Chip Select	$\overline{CS}$	Input	1
Control/Data	C/$\overline{D}$	Input	1
Source N	SRC	Input	5
Gate N	GATE	Input	5
Data Bus	DB	I/O	16
Frequency Out	FOUT	Output	1
Out N	OUT	Output	5

Figure 1-2. Interface Signal Summary

Figure 1-2 summarizes the interface signals and their abbreviations for the STC.

Package Pin	Data Bus Width (MM14)	
	16 Bits	8 Bits
12	DB0	DB0
13	DB1	DB1
14	DB2	DB2
15	DB3	DB3
16	DB4	DB4
17	DB5	DB5
18	DB6	DB6
19	DB7	DB7
20	DB8	GATE 1A
22	DB9	GATE 2A
23	DB10	GATE 3A
24	DB11	GATE 4A
25	DB12	GATE 5A
26	DB13	(VIH)
27	DB14	(VIH)
28	DB15	(VIH)

Figure 1-3. Data Bus Assignments

Interface Considerations

All of the input and output signals for the Am9513A are specified with logic levels compatible with those of standard TTL circuits. In addition to providing TTL compatible voltage levels, other output conditions are specified to help configure non-standard interface circuitry. The logic level specifications take into account all worst-case combinations of the three variables that affect the logic level thresholds: ambient temperature, supply voltage and processing parameters. A change in any of these toward nominal values will improve the actual operating margins and will increase noise immunity.

Unprotected open gate inputs of high quality MOS transistors exhibit very high resistances on the order of perhaps 10^{14} ohms. It is easy, therefore, in some circumstances, for charge to enter the gate node of such an input faster than it can be discharged and consequently, for the gate voltage to rise high enough to break down the oxides and destroy the transistor. All inputs to the Am9513A include protection networks to help prevent damaging accumulations of static charge. The protection circuitry is designed to slow the transitions of incoming current surges and to provide low-impedance discharge paths for voltages beyond the normal operating levels. Note, however, that input energy levels can nonetheless be too high to be successfully absorbed. Conventional design, storage, and handling precautions should be observed so that the protection networks themselves are not overstressed.

Within the limits of normal operation, the input protection circuitry is inactive and may be modeled as a lumped series RC as shown in Figure 1-4(a). The functionally active input connection during normal operation is the gate of a MOS transistor. No active sources or drains are connected to the inputs so that neither transient nor steady-state currents are impressed on the driving signals other than the charging or discharging of the input capacitance and the accumulated leakage associated with the protection network and the input circuit.

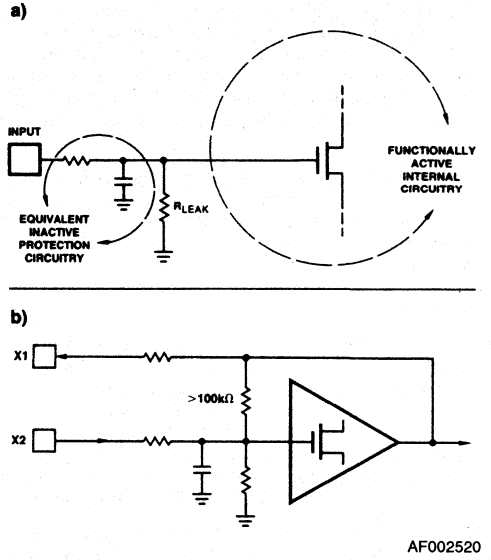


Figure 1-4. Input Circuitry

The only exception to the purely capacitive input case is the X2 crystal input. As shown in Figure 1-4(b) an internal resistor connects X1 and X2 in addition to the protection network. The resistor is a modestly high value of more than 100kohms.

Fanout from the driving circuitry into the Am9513A inputs will generally be limited by transition time considerations rather than DC current limitations when the loading is dominated by conventional MOS circuits. In an operating environment, all inputs should be terminated so they do not float and therefore will not accumulate stray static charges. Unused inputs should be tied directly to Ground or VCC, as appropriate. An input in use will have some type of logic output driving it, and termination during operation will not be a problem. Where inputs are driven from logic external to the card containing this chip, however, on-board termination should be provided to protect the chip when the board is unplugged (the input would otherwise float). A pull-up resistor or a simple inverter or gate will suffice.

DETAILED DESCRIPTION

The Am9513A System Timing Controller (STC) is a support device for processor oriented systems that is designed to enhance the available capability with respect to counting and timing operations. It provides the capability for programmable frequency synthesis, high resolution programmable duty cycle waveforms, retriggerable digital timing functions, time-of-day clocking, coincidence alarms, complex pulse generation, high resolution baud rate generation, frequency shift keying, stop-watching timing, event count accumulation, waveform analysis and many more. A variety of programmable operating modes and control features allow the Am9513A to be personalized for particular applications as well as dynamically reconfigured under program control.

The STC includes five general-purpose 16-bit counters. A variety of internal frequency sources and external pins may be selected as inputs for individual counters with software selectable active-high or active-low input polarity. Both hardware and software gating of each counter is available. Three-state outputs for each counter provide either pulses or levels. The counters can be programmed to count up or down in either binary or BCD. The accumulated count may be read without disturbing the counting process. Any of the counters may be internally concatenated to form an effective counter length of up to 80 bits.

The Am9513A block diagrams indicate the interface signals and the basic flow of information. Internal control lines and the internal data bus have been omitted. The control and data registers are all connected to a common internal 16-bit bus. The external bus may be 8- or 16-bits wide; in the 8-bit mode, the internal 16-bit information is multiplexed to the low order data bus pins DB0 through DB7.

An internal oscillator provides a convenient source of frequencies for use as counter inputs. The oscillator's frequency is controlled at the X1 and X2 interface pins by an external reactive network such as a crystal. The oscillator output is divided by the Frequency Scaler to provide several sub-frequencies. One of the scaled frequencies (or one of ten input signals) may be selected as an input to the FOUT divider and then comes out of the chip at the FOUT interface pin.

The STC is addressed by the external system as two locations: a Control port and a Data port. The Control port provides direct access to the Status and Command registers, as well as allowing the user to update the Data Pointer register. The Data port is used to communicate with all other addressable internal locations. The Data Pointer register controls the Data port addressing.

Among the registers accessible through the Data port are the Master Mode register and five Counter Mode registers, one for each counter. The Master Mode register controls the programmable options that are not controlled by the Counter Mode registers.

Each of the five general-purpose counters is 16-bits long and is independently controlled by its Counter Mode register. Through this register, a user can software select one of 16 sources as the counter input, a variety of gating and repetition modes, up or down counting in binary or BCD and active-high or active-low input and output polarities.

Associated with each counter are a Load register and a Hold register, both accessible through the Data port. The Load register is used to automatically reload the counter to any predefined value, thus controlling the effective count period. The Hold register is used to save count values without disturbing the count process, permitting the host processor to read intermediate counts. In addition, the Hold register may be used as a second Load register to generate a number of complex output waveforms.

All five counters have the same basic control logic and control registers. Counters 1 and 2 have additional Alarm registers and comparators associated with them, plus the extra logic necessary for operating in a 24-hour time-of-day mode. For real-time operation, the time-of-day logic will accept 50Hz, 60Hz or 100Hz input frequencies.

Each general counter has a single dedicated output pin. It may be turned off when the output is not of interest or may be configured in a variety of ways to drive interrupt controllers, Darlington buffers, bus drivers, etc. The counter inputs, on the other hand, are specifically not dedicated to any given interface line. Considerable versatility is available for configuring both the input and the gating of individual counters. This not only permits dynamic reassignment of inputs under software control, but also allows multiple counters to use a single input and a single gate pin to control more than one counter. Indeed, a single pin can be the gate for one counter and, at the same time, the count source for another.

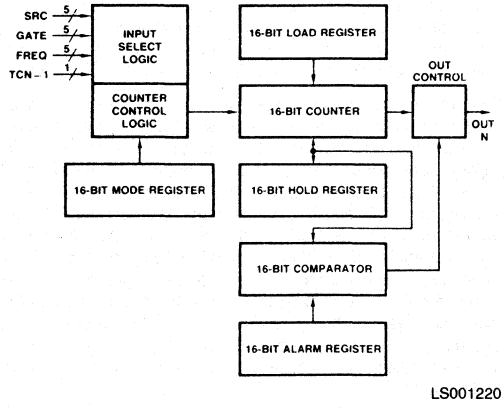


Figure 1-5. Counter Logic Groups 1 and 2

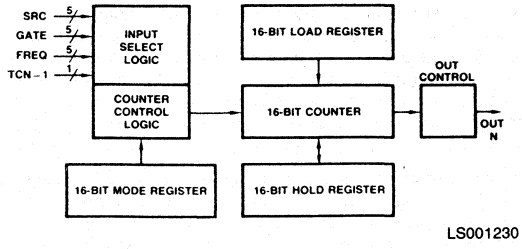


Figure 1-6. Counter Logic Groups 3, 4 and 5

A powerful command structure simplifies user interaction with the counters. A counter must be armed by one of the ARM commands before counting can commence. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. The ARM and DISARM commands permit software gating of the count process in some modes.

The LOAD command causes the counter to be reloaded with the value in either the associated Load register or the associated Hold register. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

The DISARM command disables further counting independent of any hardware gating. A disarmed counter may be reloaded using the LOAD command, may be incremented or decremented using the STEP command and may be read using the SAVE command. A count process may be resumed using an ARM command.

The SAVE command transfers the contents of a counter to its associated Hold register. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Two combinations of the basic commands exist to either LOAD AND ARM or to DISARM AND SAVE any combination of counters. Additional commands are provided to: step an individual counter by one count; set and clear an output toggle; issue a software reset; clear and set special bits in the Master Mode register; and load the Data Pointer register.

Note: Separate LOAD and ARM commands should be used for asynchronous operations.

Power Supply

The Am9513A requires only a single 5V power supply. Maximum supply currents are specified in the electrical specification at the high end of the voltage tolerance and the low end of the temperature range. In addition, the current specifications take into account the worstcase distribution of processing parameters that may be encountered during the manufacturing life of the product. Typical supply current values, on the other hand, are specified at a nominal +5.0 volts, a nominal ambient temperature of 25°C, and nominal processing parameters. Supply current always decreases with increasing ambient temperature: thermal run-away is not a problem.

Supply current will vary somewhat from part to part, but a given unit at a given operating temperature will exhibit a nearly constant power drain. There is no functional operating region that will cause more than a few percent change in the supply current. Decoupling of VCC, then, is straightforward and will generally be used to isolate the Am9513A from VCC noise originating externally

CONTROL PORT REGISTERS

The STC is addressed by the external system as only two locations: a Control port and a Data port. Transfers at the Control port (C/$\overline{D}$ = HIGH) allow direct access to the Command register when writing and the Status register when reading. All other available internal locations are accessed for both reading and writing via the Data port (C/$\overline{D}$ = LOW). Data port transfers are executed to and from the location currently addressed by the Data Pointer register. Options available in the Master Mode register and the Data Pointer control structure allow several types of transfer sequencing to be used. See Figure 1-7.

Transfers to and from the Control port are always 8-bits wide. Each access to the Control port will transfer data between the Command register (writes) or Status register (reads) and Data Bus pins DB0-DB7, regardless of whether the Am9513A is in 8- or 16-bit bus mode. When the Am9513A is in 8-bus mode, Data Bus pins DB13-DB15 should be held at a logic high whenever $\overline{CS}$ and WR are both active.

Command Register

The Command register provides direct control over each of the five general counters and controls access through the Data port by allowing the user to update the Data Pointer register. The "Command Description" section of this data sheet explains the detailed operation of each command. A summary of all commands appears in Figure 1-20. Six of the command types are used for direct software control of the counting process. Each of these six commands contains a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of the five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter.

Data Pointer Register

The 6-bit Data Pointer register is loaded by issuing the appropriate command through the Control port to the Command register. As shown in Figure 1-7, the contents of the Data Pointer register are used to control the Data port multiplexer, selecting which internal register is to be accessible through the Data port.

The Data Pointer consists of a 3-bit Group Pointer, a 2-bit Element Pointer and a 1-bit Byte Pointer, depicted in Figure 1-8. The Byte Pointer bit indicates which byte of a 16-bit register is to be transferred on the next access through the Data port. Whenever the Data Pointer is loaded, the Byte Pointer bit is set to one, indicating a least-significant byte is expected. The Byte Pointer toggles following each 8-bit data transfer with an 8-bit data bus (MM13 = 0), or it always remains set with the 16-bit data bus option (MM13 = 1). The Element and Group pointers are used to select which internal register is to be accessible through the Data port. Although the contents of the Element and Group Pointer in the Data Pointer register cannot be read by the host processor, the Byte Pointer is available as a bit in the Status register.

Random access to any available internal data location can be accomplished by simply loading the Data Pointer using the command shown in Figure 1-9 and then initiating a data read or data write. This procedure can be used at any time, regardless of the setting of the Data Pointer Control bit (MM14). When the 8-bit data bus configuration is being used (MM13 = 0), two bytes of data would normally be transferred following the issuing of the "Load Data Pointer" command.

To permit the host processor to rapidly access the various internal registers, automatic sequencing of the Data Pointer is provided.

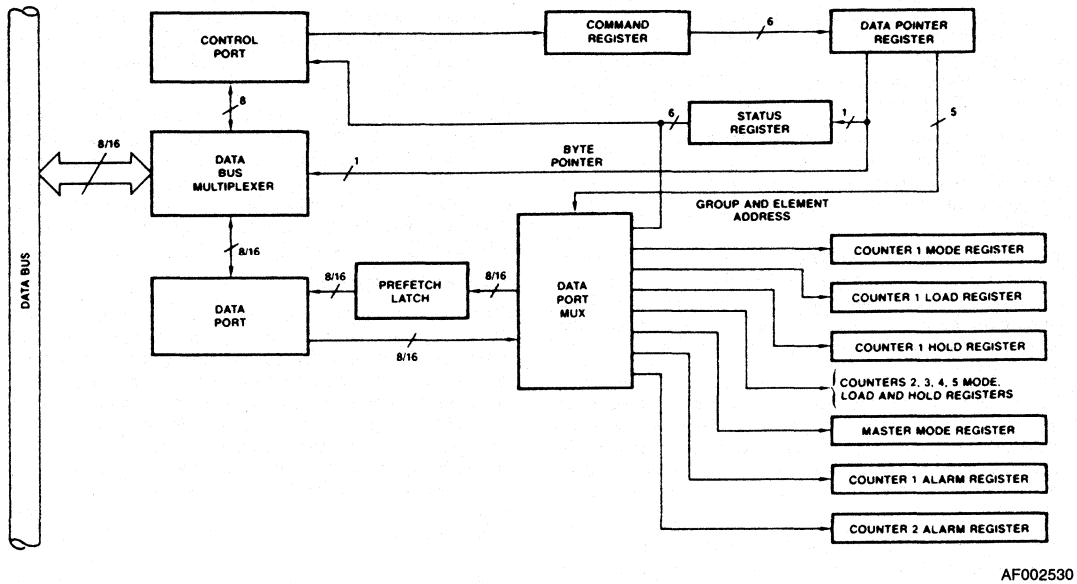


Figure 1-7. Am9513A Register Access

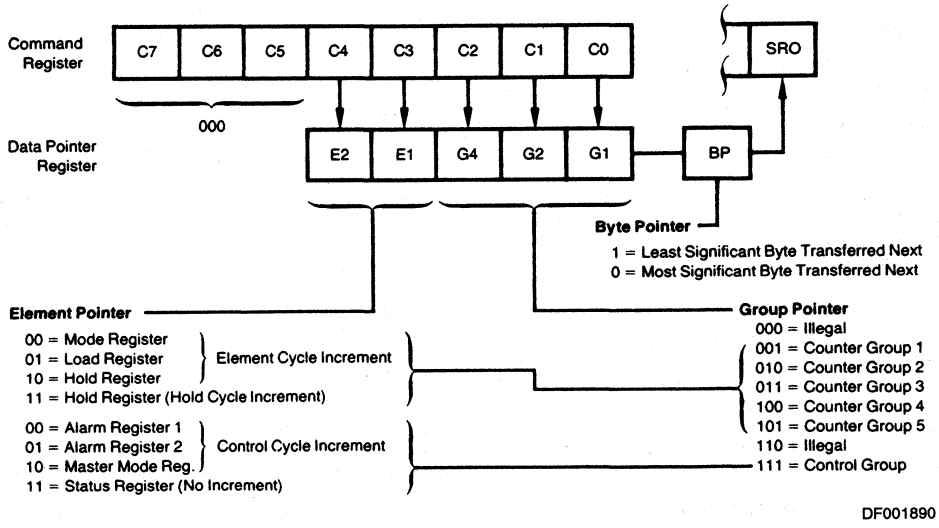


Figure 1-8. Data Pointer Register

	Element Cycle			Hold Cycle
	Mode Register	Load Register	Hold Register	Hold Register
Counter 1	FF01	FF09	FF11	FF19
Counter 2	FF02	FF0A	FF12	FF1A
Counter 3	FF03	FF0B	FF13	FF1B
Counter 4	FF04	FF0C	FF14	FF1C
Counter 5	FF05	FF0D	FF15	FF1D

Master Mode Register = FF17
Alarm 1 Register = FF07
Alarm 2 Register = FF0F
Status Register = FF1F

Notes:
1. All codes are in hex.
2. When used with an 8-bit bus, only the two low order hex digits should be written to the command port; the 'FF' prefix should be used only for a 16-bit data bus interface.

Figure 1-9. Load Data Pointer Commands

Sequencing is enabled by clearing Master Mode bit 14 (MM14) to zero. As shown in Figure 1-10 several types of sequencing are available depending on the data bus width being used and the initial Data Pointer value entered by command.

When E1 = 0 or E2 = 0 and G4, G2, G1 points to a Counter Group, the Data Pointer will proceed through the Element cycle. The Element field will automatically sequence through the three values 00, 01 and 10 starting with the value entered. When the transition from 10 to 00 occurs, the Group field will also be incremented by one. Note that the Element field in this case does not sequence to a value of 11. The Group field circulates only within the five Counter Group codes.

If E2, E1 = 11 and a Counter Group are selected, then only the Group field is sequenced. This is the Hold cycle. It allows the Hold registers to be sequentially accessed while bypassing the Mode and Load registers. The third type of sequencing is the Control cycle. If G4, G2, G1 = 111 and E2, E1 ≠ 11, the Element Pointer will be incremented through the values 00, 01 and 10, with no change to the Group Pointer.

When G4, G2, G1 = 111 and E2, E1 = 11, no incrementing takes place and only the Status register will be available through the Data port. Note that the Status register can also always be read directly through the Control port.

For all these auto-sequencing modes, if an 8-bit data bus is used, the Byte pointer will toggle after every data transfer to allow the least and most significant bytes to be transferred before the Element or Group fields are incremented.

Prefetch Circuit

To minimize the read access time to internal Am9513A registers, a prefetch circuit is used for all read operations through the Data port. Following each read or write operation through the Data port, the Data Pointer register is updated to point to the next register to be accessed. Immediately following this update, the new register data is transferred to a special prefetch latch at the interface pad logic. When the user performs a subsequent read of the Data port, the data bus drivers are enabled, outputting the prefetched data on the bus. Since the internal data register is accessed prior to the start of the read operation, its access time is transparent to the user. To keep the prefetched data consistent with the Data Pointer, prefetches are also performed after each write to the Data port and after execution of the "Load Data Pointer" command. The following rules should be kept in mind regarding Data port Transfers.

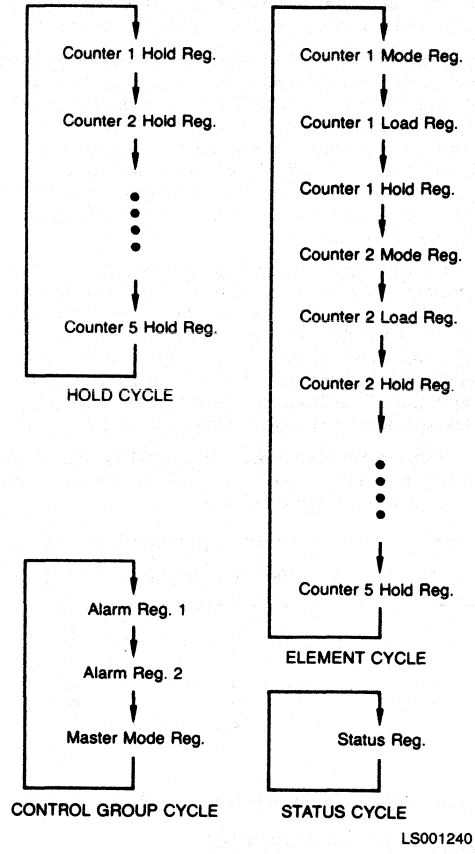


Figure 1-10. Data Pointer Sequencing

1. The Data Pointer register should always be reloaded before reading from the Data port if a command, other than "Load Data Pointer," was issued to the Am9513A following the last Data port read or write. The Data Pointer does not have to be loaded again if the first Data port transaction after a command entry is a write, since the Data port write will automatically cause a new prefetch to occur.

2. Operating modes N, O, Q, R and X allow the user to save the counter contents in the Hold register by applying an active-going gate edge. If the Data Pointer register had been pointing to the Hold register in question, the prefetched value will not correspond to the new value saved in the Hold Register. To avoid reading an incorrect value, a new "Load Data Pointer" command should be issued before attempting to read the saved data. A Data port write (to another register) will also initiate a prefetch; subsequent reads will access the recently saved Hold register data. Many systems will use the "saving" gate edge to interrupt the host CPU. In systems such as this, the interrupt service routine should issue a "Load Data Pointer" command prior to reading the saved data.

Status Register

The 8-bit read-only Status register indicates the state of the Byte Pointer bit in the Data Pointer register and the state of the OUT signal for each of the general counters. See Figures 1-11 and 1-18. The OUT signals reported are those internal to the

chip after the polarity-select logic and just before the 3-state interface buffer circuitry. Bits SR6 and SR7 may be 0 or 1.

The Status register OUT bit reflects an active-high or active-low TC output, or a TC Toggled output, as programmed in the Output Control Field of the Counter Mode register. That is, it reflects the exact state of the OUT pin. When the low-impedance to Ground Output option (CM2-CM0 = 000) is selected, the Status register will reflect an active-high TC Output. When a high-impedance Output option (CM2-CM0 = 100) is selected, the Status register will reflect an active-low TC output.

For Counters 1 and 2, the OUT pin will reflect the comparator output if the comparators are enabled. The Status register bit and OUT pin are active-high if CM2 = 0 and active-low if CM2 = 1. When the high-impedance option is selected and the comparator is enabled, the status register bit will reflect an active-high comparator output. When the low-impedance to Ground option is selected and the comparator is enabled, the status register bit will reflect an active-low comparator output.

The Status register is normally accessed by reading the Control port (see Figure 1-7) but may also be read via the Data port as part of the Control Group.

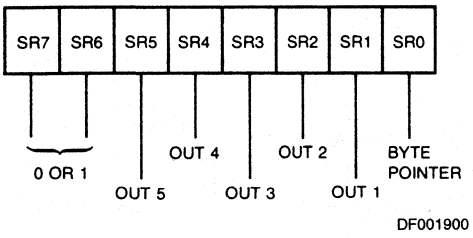


Figure 1-11. Status Register Bit Assignments

DATA PORT REGISTERS

Counter Logic Groups

As shown in Figures 1-5 and 1-6, each of the five Counter Logic Groups consists of a 16-bit general counter with associated control and output logic, a 16-bit Load register, a 16-bit Hold register and a 16-bit Mode register. In addition, Counter Groups 1 and 2 also include 16-bit Comparators and 16-bit Alarm registers. The comparator/alarm functions are controlled by the Master Mode register. The operation of the Counter Mode registers is the same for all five counters. The host CPU has both read and write access to all registers in the Counter Logic Groups through the Data port. The counter itself is never directly accessed.

Load Register

The 16-bit read/write Load register is used to control the effective length of the general counter. Any 16-bit value may be written into the Load register. That value can then be transferred into the counter each time the Terminal Count (TC) occurs. "Terminal Count" is defined as that period of time when the counter contents would have been zero if an external value had not been transferred into the counter. Thus, the terminal count frequency can be the input frequency divided by the value in the Load register. In all operating modes, either the Load or Hold register will be transferred into the counter when TC occurs. In cases where values are being accumulated in the counter, the Load register action can become transparent by filling the Load register with all zeros.

Hold Register

The 16-bit read/write Hold register is dual-purpose. It can be used in the same way as the Load register, thus offering an alternate source for module definition for the counter. The Hold register may also be used to store accumulated counter values for later transfer to the host processor. This allows the count to be sampled while the counting process proceeds without interruption. Transfer of the counter contents into the Hold register is accomplished by the hardware interface in some operating modes or by software commands at any time.

Counter Mode Register

The 16-bit read/write Counter Mode register controls the gating, counting, output and source select functions within each Counter Logic Group. The "Counter Mode Control Options" section of this document describes the detailed control options available. Figure 1-17 shows the bit assignments for the Counter Mode registers.

Alarm Registers and Comparators

Added functions are available in the Counter Logic Groups for Counters 1 and 2 (see Figure 1-5). Each contains a 16-bit Alarm register and a 16-bit Comparator. When the value in the counter reaches the value in the Alarm register, the Comparator output will go true. The Master Mode register contains control bits to individually enable/disable the comparators. When enabled, the comparator output appears on the OUT pin of the associated counter in place of the normal counter output. The output will remain true as long as the comparison is true, that is, until the next input causes the count to change. The polarity of the Comparator output will be active-high if the Output Control field of the Counter Mode register is 001 or 010 and active-low if the Output Control field is 101.

MASTER MODE CONTROL OPTIONS

The 16-bit Master Mode (MM) register is used to control those internal activities that are not controlled by the individual Counter Mode registers. This includes frequency control, Time-of-Day operation, comparator controls, data bus width and data pointer sequencing. Figure 1-12 shows the bit assignments for the Master Mode register. This section describes the use of each control field.

Master Mode register bits MM12, MM13 and MM14 can be individually set and reset using commands issued to the Command register. In addition, they can all be changed by writing directly to the Master Mode register.

After power-on reset or a Master Reset command, the Master Mode register is cleared to an all zero condition. This results in the following configuration:

Time-of-Day disabled
Both Comparators disabled
FOUT Source is frequency F1
FOUT Divider set for divide-by-16
FOUT gated on
Data Bus 8 bits wide
Data Pointer Sequencing enabled
Frequency Scaler divides in binary

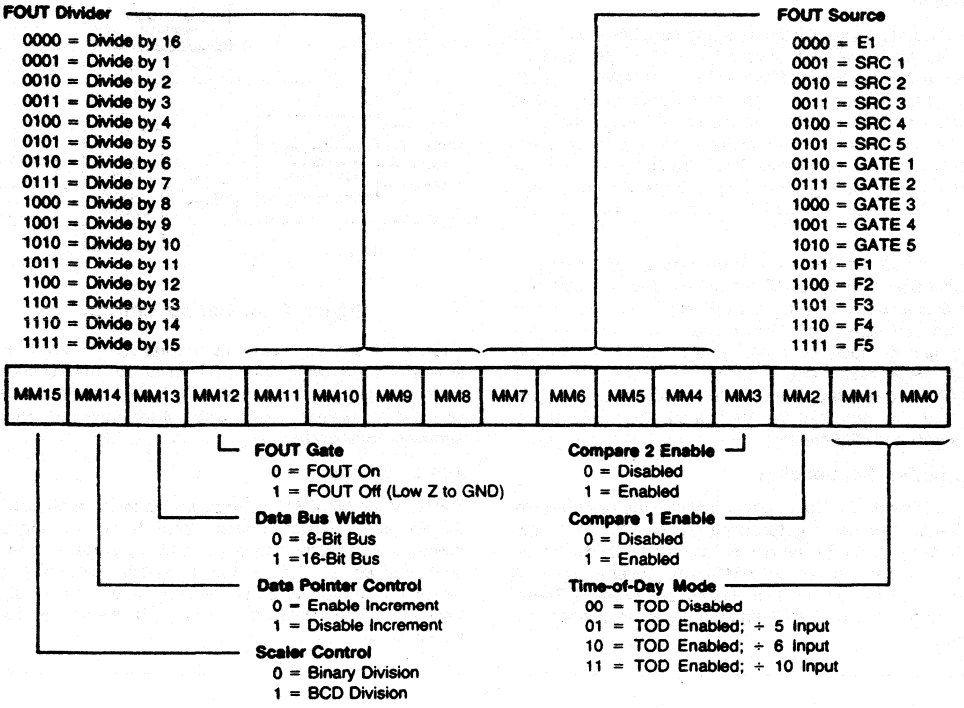


Figure 1-12. Master Mode Register Bit Assignments

Time-of-Day

Bits MM0 and MM1 of the Master Mode register specify the Time-of-Day (TOD) options. When MM0 = 0 and MM1 = 0, the special logic used to implement TOD is disabled, and Counters 1 and 2 will operate in exactly the same way as Counters 3, 4 and 5. When MM0 = 1 or MM1 = 1, additional counter decoding and control logic is enabled on Counters 1 and 2, which causes their decades to turn over at the counts that generate appropriate 24-hour TOD accumulations. For additional information, see the Time-of-Day chapter in the 9513A System timing controller technical manual.

Comparator Enable

Bits MM2 and MM3 control the Comparators associated with Counters 1 and 2. When a Comparator is enabled, its output is substituted for the normal counter output on the associated OUT1 or OUT2 pin. The comparator output will be active-high if the output control field of the Counter Mode register is 001 or 010 and active-low for a code of 101. Once the compare output is true, it will remain so until the count changes and the comparison therefore goes false.

The two Comparators can always be used individually in any operating mode. One special case occurs when the Time-of-Day option is revoked and both Comparators are enabled. The operation of Comparator 2 will then be conditioned by Comparator 1 so that a full 32-bit compare must be true in order to generate a true signal on OUT2. OUT1 will continue, as usual, to reflect the state of the 16-bit comparison between Alarm 1 and Counter 1.

FOUT Source

Master Mode bits MM4 through MM7 specify the source input for the FOUT divider. Fifteen inputs are available for selection, and they include the five Source pins, the five Gate pins and the five internal frequencies derived from the oscillator. The 16th combination of the four control bits (all zeros) is used to assure that an active frequency is available at the input to the FOUT divider following reset.

FOUT Divider

Bits MM8 through MM11 specify the dividing ratio for the FOUT Divider. The FOUT source (selected by bits MM4 through MM7) is divided by an integer value between 1 and 16, inclusive, and is then passed to the FOUT output buffer. After power-on or reset, the FOUT divider is set to divide-by-16.

FOUT Gate

Master Mode bit MM12 provides a software gating capability for the FOUT signal. When MM12 = 1, FOUT is off and in a low-impedance state to ground. MM12 may be set or cleared in conjunction with the loading of the other bits in the Master Mode register; alternatively, there are commands that allow MM12 to be individually set or cleared directly without changing any other Master Mode bits. After power-up or reset, FOUT is gated on.

When changing the FOUT divider ratio or FOUT source, transient pulses as short as half the period of the FOUT source may appear on the FOUT pin. Turning the FOUT gate on or off can also generate a transient. This should be considered when using FOUT as a system clock source.

Bus Width

Bit MM13 controls the multiplexer at the data bus interface in order to configure the part for an 8-bit or 16-bit external bus. The internal bus is always 16-bits wide. When MM13 = 1, 16-bit data is transferred directly between the internal bus and all 16 of the external bus lines. In this configuration, the Byte Pointer bit in the Data Pointer register remains set at all times. When MM13 = 0, 16-bit internal data is transferred a byte at a time to and from the eight low-order external data bus lines. The Byte Pointer bit toggles with each byte transfer in this mode.

When the Am9513A is set to operate with an 8-bit data bus width, pins DB8 through DB15 are not used for the data bus and are available for other functions. Pins DB13 through DB15 should be tied high. Pins DB8 through DB12 are used as auxiliary gating inputs and are labeled GATE1A through GATE5A respectively. The auxiliary gate pin, GATENA, is logically ANDed with the gate input to Counter N, as shown in Figure 1-13. The output of the AND gate is then used as the gating signal for Counter N.

Data Pointer Sequencing

Bit MM14 controls the Data Pointer logic to enable or disable the automatic sequencing functions. When MM14 = 1, the contents of the Data Pointer can be changed only directly by entering a command. When MM14 = 0, several types of automatic sequencing of the Data Pointer are available. These are described in the Data Pointer register section of this document.

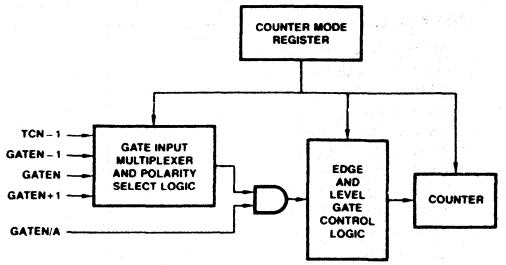
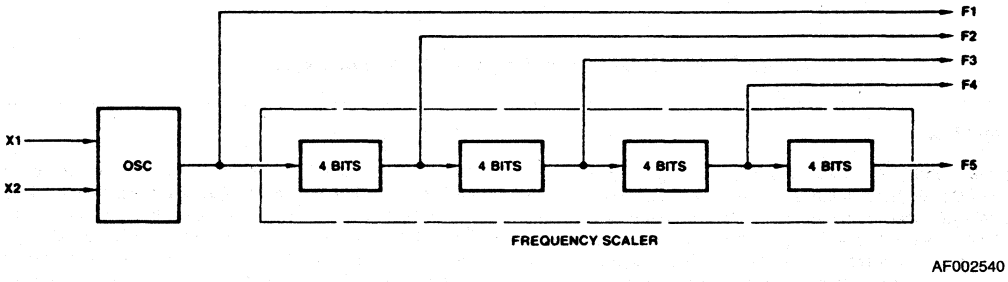


Figure 1-13. Gating Control

Thus, the host processor, by controlling MM14, may repetitively read/write a single internal location, or may sequentially read/write groups of locations. Bit MM14 can be loaded by writing to the Master Mode register or can be set or cleared by software command.

Scaler Ratios

Master Mode bit MM15 controls the counting configuration of the Frequency Scaler counter. When MM15 = 0, the Scaler divides the oscillator frequency in binary steps so that each subfrequency is 1/16 of the preceding frequency. When MM15 = 1, the Scaler divides in BCD steps so that adjacent frequencies are related by ratios of 10 instead of 16 (see Figure 1-14).



Frequency	BCD Scaling MM15 = 1	Binary Scaling MM15 = 0
F1	OSC	OSC
F2	F1 ÷ 10	F1 ÷ 16
F3	F1 ÷ 100	F1 ÷ 256
F4	F1 ÷ 1,000	F1 ÷ 4,096
F5	F1 ÷ 10,000	F1 ÷ 65,536

Figure 1-14. Frequency Scaler Ratios

Counter Mode	A	B	C	D	E	F	G	H	I	J	K	L
Special Gate (CM7)	0	0	0	0	0	0	0	0	0	0	0	0
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15-CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm	X	X	X									
Count to TC twice, then disarm							X	X	X			
Count to TC repeatedly without disarming				X	X	X				X	X	X
Gate input does not gate counter input	X			X			X			X		
Count only during active gate level		X			X			X			X	
Start count on active gate edge and stop count on next TC			X			X						
Start count on active gate edge and stop count on second TC									X			X
No hardware retriggering	X	X	X	X	X	X	X	X	X	X	X	X
Reload counter from Load register on TC	X	X	X	X	X	X						
Reload counter on each TC, alternating reload source between Load and Hold registers							X	X	X	X	X	X
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.												
On active gate edge transfer counter into Hold register and then reload counter from Load register												

Counter Mode	M	N	O	P	Q	R	S	T	U	V	W	X
Special Gate (CM7)	1	1	1	1	1	1	1	1	1	1	1	1
Reload Source (CM6)	0	0	0	0	0	0	1	1	1	1	1	1
Repetition (CM5)	0	0	0	1	1	1	0	0	0	1	1	1
Gate Control (CM15-CM13)	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE	000	LEVEL	EDGE
Count to TC once, then disarm		X	X									
Count to TC twice, then disarm							X					
Count to TC repeatedly without disarming					X	X				X		X
Gate input does not gate counter input							X			X		
Count only during active gate level		X			X							
Start count on active gate edge and stop count on next TC			X			X						X
Start count on active gate edge and stop count on second TC												
No hardware retriggering							X			X		X
Reload counter from Load register on TC		X	X		X	X						X
Reload counter on each TC, alternating reload source between Load and Hold registers.												
Transfer Load register into counter on each TC that gate is LOW, transfer Hold register into counter on each TC that gate is HIGH.							X			X		
On active gate edge transfer counter into Hold register and then reload counter from Load register		X	X		X	X						
On active gate edge transfer counter into Hold register, but counting continues												X

Notes: 1. Counter modes M, P, T, U and W are reserved and should not be used.
2. Mode X is available for Am9513A only.

Figure 1-15 Counter Mode Operating Summary

COUNTER MODE DESCRIPTIONS

Counter Mode register bits CM15-CM13 and CM7-CM5 select the operating mode for each counter (see Figure 1-15). To simplify references to a particular mode, each mode is assigned a letter from A through X. Representative waveforms for the counter modes are illustrated in Figures 1-16a through 1-16v. (Because the letter suffix in the figure number is keyed to the mode, Figures 1-16m, 1-16p, 1-16t, 1-16u and 1-16w do not exist.) The figures assume down counting on rising source edges. Those modes which automatically disarm the counter (CM5 = 0) are shown with the $\overline{WR}$ plus entering the required ARM command; for modes which count repetitively (CM5 = 1), the ARM command is omitted. The retriggering modes (N, O, Q and R) are shown with one retrigger operation. Both a TC output waveform and a TC Toggled output waveform are shown for each mode. The symbols L and H are used to represent count values equal to the Load and Hold register contents, respectively. The symbols K and N represent arbitrary count values. For each mode, the required bit pattern in the Counter Mode register is shown; "don't care" bits are marked "X." These figures are designed to clarify the mode descriptions; the Am9513A Electrical Specification should be used as the authoritative reference for timing relationships between signals.

To keep the following mode descriptions concise and to the point, the phrase "source edges" is used to refer to active-going source edges only, not to inactive-going edges. Similarly, the phrase "gate edges" refers only to active-going gate edges. Also, again to avoid verbosity and euphuism, the descriptions of some modes state that a counter is stopped or disarmed "on a TC, inhibiting further counting." As is fully explained in the TC section of this document, for these modes the counter is actually stopped or disarmed following the active-going source edge which drives the counter out of TC. In other words, since a counter in the TC state always counts, irrespective of its gating or arming status, the stopping or disarming of the count sequence is delayed until TC is terminated.

MODE A

Software-Triggered Strobe with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode A, shown in Figure 1-16a, is one of the simplest operating modes. The counter will be available for counting source edges when it is issued an ARM command. On each TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume when a new ARM command is issued.

MODE B

Software-Triggered Strobe with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode B, shown in Figure 1-16b, is identical to Mode A except that source edges are counted only when the assigned Gate is active. The counter must be armed before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. On each TC the counter will reload from the Load register and automatically disarm itself, inhibiting further counting unitl a new ARM command is issued.

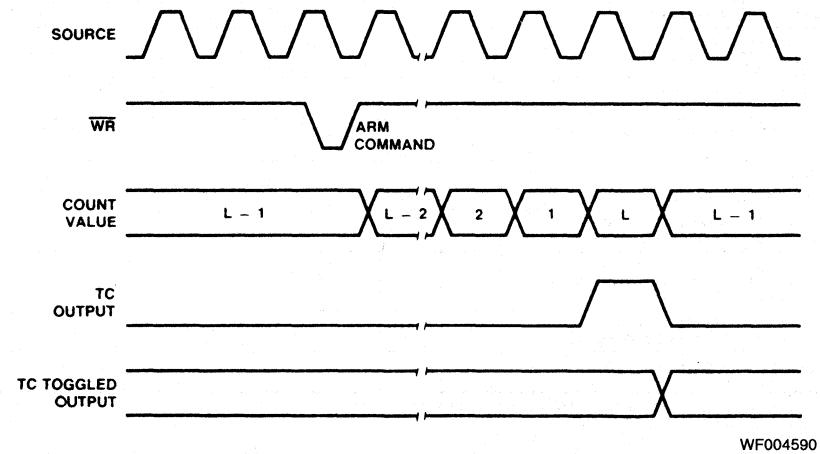


Figure 1-16a. Mode A Waveforms

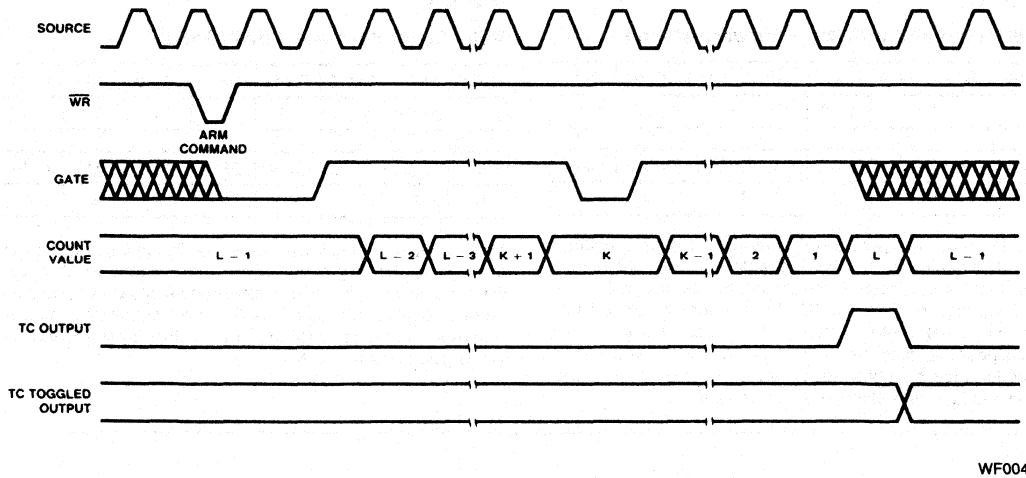


Figure 1-16b. Mode B Waveforms

MODE C

Hardware-Triggered Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	0	X	X	X	X	X

Mode C, shown in Figure 1-16c, is identical to Mode A, except that counting will not begin until a Gate edge is applied to the armed counter. The counter must be armed before application of the triggered Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting on the first source edge after the triggering Gate edge and will continue counting until TC. At TC, the counter will reload from the Load register and automatically disarm itself. Counting will then remain inhibited until a new ARM command and a new Gate edge are applied in that order. Note that after application of a triggered Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode B, where the Gate can be modulated throughout the count cycle to stop and start the counter.

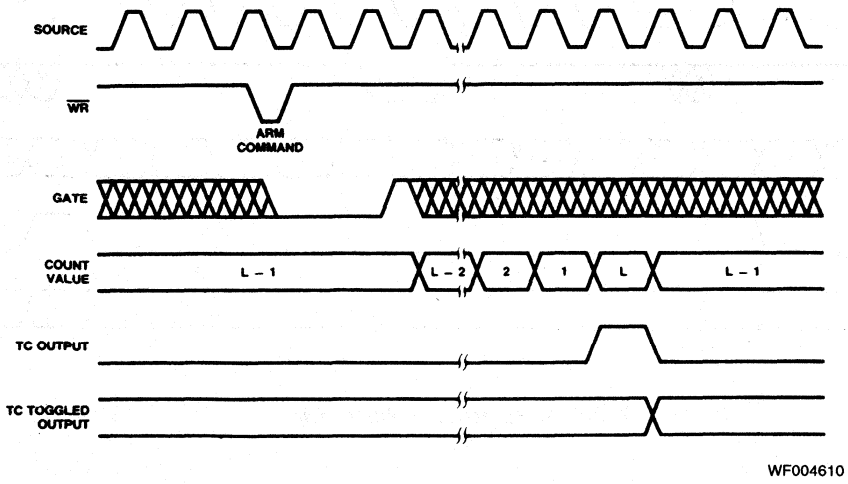


Figure 1-16c. Mode C Waveforms

MODE D

Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode D, shown in Figure 1-16d, is typically used in frequency generation applications. In this mode, the Gate input does not affect counter operation. Once armed, the counter will count to TC repetitively. On each TC, the counter will reload itself from the Load register; hence, the Load register value determines the time between TCs. A square wave rate generator may be obtained by specifying the TC Toggled output mode in the Counter Mode register.

MODE E

Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode E, shown in Figure 1-16e, is identical to Mode D, except the counter will only count those source edges which occur while the Gate input is active. This feature allows the counting process to be enabled and disabled under hardware control. A square wave rate generator may be obtained by specifying the TC Toggled output mode.

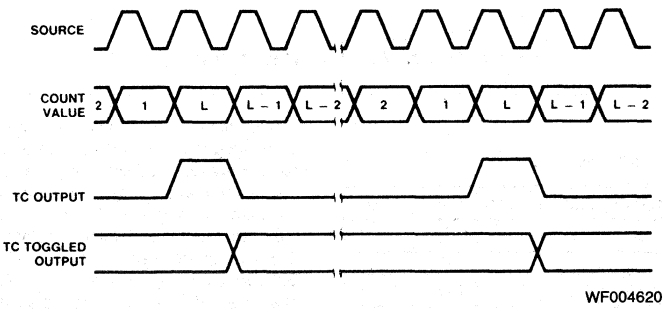


Figure 1-16d. Mode D Waveforms

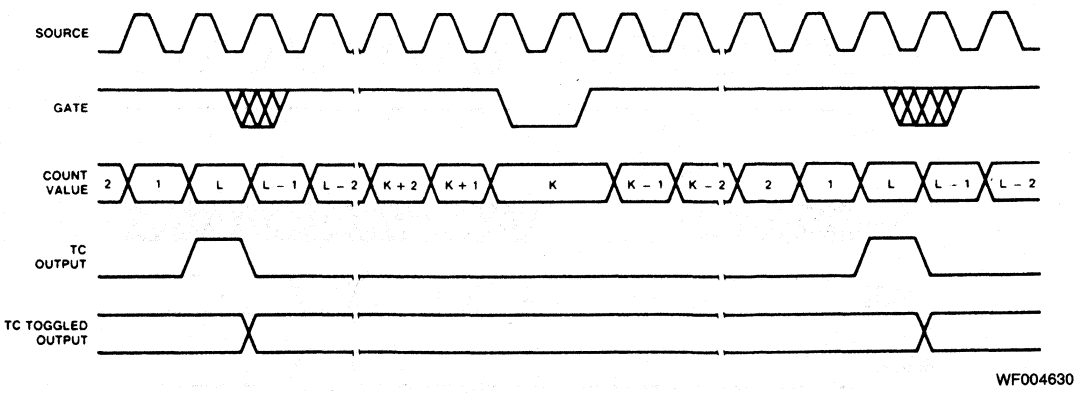


Figure 1-16e. Mode E Waveforms

MODE F

Non-Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	0	1	X	X	X	X	X

Mode F, shown in Figure 1-16f, provides a non-retriggerable one-shot timing function. The counter must be armed before it will function. Application of a Gate edge to the armed counter will enable counting. When the counter reaches TC, it will reload itself from the Load register. The counter will then stop counting, awaiting a new Gate edge. Note that unlike Mode C, a new ARM command is not needed after TC, only a new Gate edge. After application of a triggering Gate edge, the Gate input is disregarded until TC.

MODE G

Software-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

In Mode G, the Gate does not affect the counter's operation. Once armed, the counter will count to TC twice and then automatically disarm itself. For most applications, the counter will initially be loaded from the Load register either by a LOAD command or by the last TC of an earlier timing cycle. Upon counting to the first TC, the counter will reload itself from the Hold register. Counting will proceed until the second TC, when the counter will reload itself from the Load register and automatically disarm itself, inhibiting further counting. Counting can be resumed by issuing a new ARM command. A software-triggered delayed pulse one-shot may be generated by specifying the TC Toggled output mode in the Counter Mode register. The initial counter contents control the delay from the ARM command until the output pulse starts. The Hold register contents control the pulse duration. Mode G is shown in Figure 1-16g.

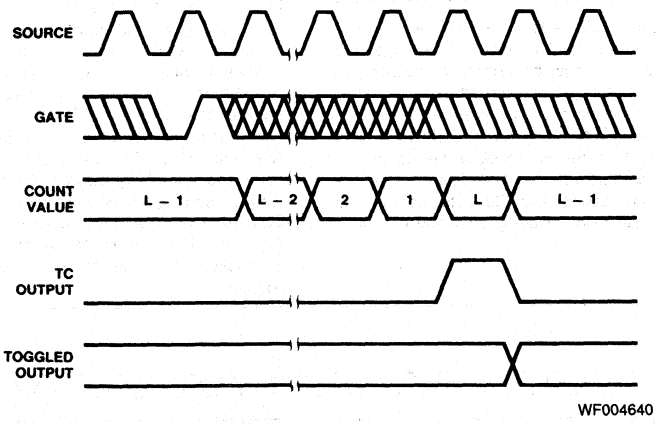


Figure 1-16f. Mode F Waveforms

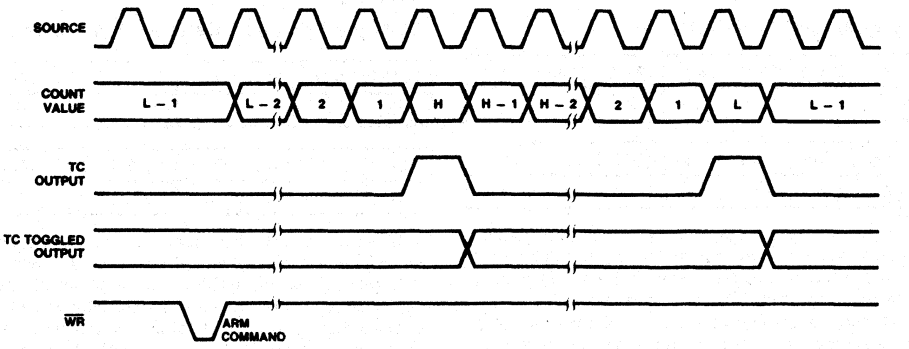


Figure 1-16g. Mode G Waveforms

MODE H

Software-Triggered Delayed Pulse One-Shot with Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode H, shown in Figure 1-16h, is identical to Mode G except that the Gate input is used to qualify which source edges are to be counted. The counter must be armed for counting to occur. Once armed, the counter will count all source edges that occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode G, the counter will be reloaded from the Hold register on the first TC and reloaded from the Load register and disarmed on the second TC. This mode allows the Gate to control the extension of both the initial output delay time and the pulse width.

MODE I

Hardware-Triggered Delayed Pulse Strobe

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	0	X	X	X	X	X

Mode I, shown in Figure 1-16i, is identical to Mode G, except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. An armed counter will start counting on the first source edge after the triggering Gate edge. Counting will then proceed in the same manner as in Mode G. After the second TC, the counter will disarm itself. An ARM command and Gate edge must be issued in this order to restart counting. Note that after application of a triggering Gate edge, the Gate input will be disregarded until the second TC. This differs from Mode H, where the Gate can be modulated throughout the count cycle to stop and start the counter.

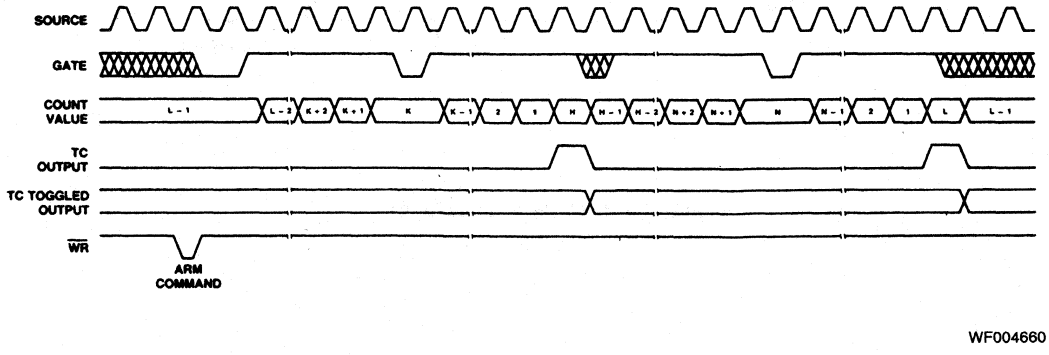


Figure 1-16h. Mode H Waveforms

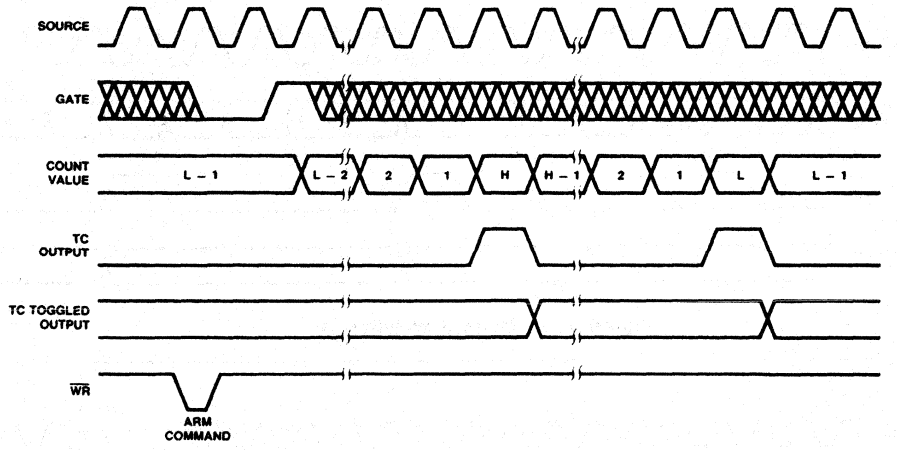


Figure 1-16i. Mode I Waveforms

MODE J

Variable Duty Cycle Rate Generator with No Hardware Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode J, shown in Figure 1-16i, will find the greatest usage in frequency generation applications with variable duty cycle requirements. Once armed, the counter will count continuously until it is issued a DISARM command. On the first TC, the counter will be reloaded from the Hold register. Counting will then proceed until the second TC at which time the counter will be reloaded from the Load register. Counting will continue, with the reload source alternating on each TC, until a DISARM command is issued to the counter. (The third TC reloads from the Hold register, the fourth TC reloads from the Load register, etc.) A variable duty cycle output can be generated by specifying the TC Toggled output in the Counter Mode register. The Load and Hold values then directly control the output duty cycle, with high resolution available when relatively high count values are used.

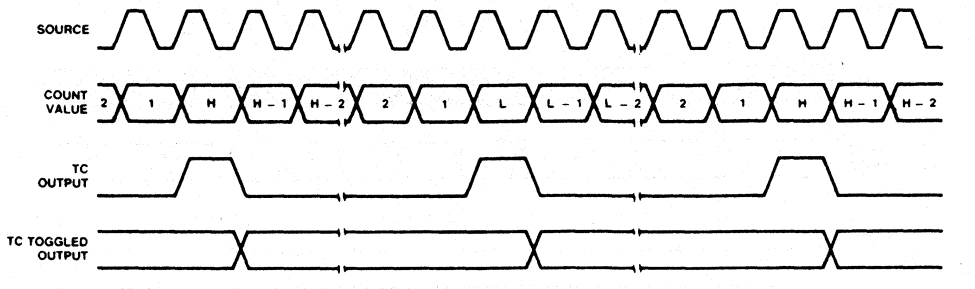
MODE K

Variable Duty Cycle Rate Generator with Level Gating

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

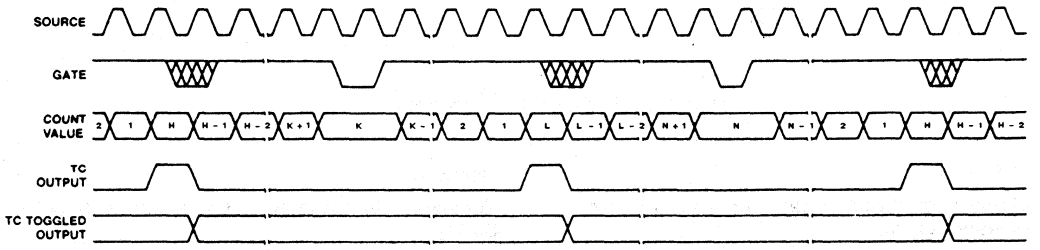
CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode K, shown in Figure 1-16k, is identical to Mode J, except that source edges are only counted when the Gate is active. The counter must be armed for counting to occur. Once armed, the counter will count all source edges which occur while Gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. As with Mode J, the reload source used will alternate on each TC, starting with the Hold register on the first TC after any ARM command. When the TC Toggled output is used, this mode allows the Gate to modulate the duty cycle of the output waveform. It can affect both the HIGH and LOW portions of the output waveform.



WF004680

Figure 1-16j. Mode J Waveforms



WF004690

Figure 1-16k. Mode K Waveforms

MODE L

Hardware-Triggered Delayed Pulse One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
0	1	1	X	X	X	X	X

Mode L, shown in Figure 1-16l, is similar to Mode J except that counting will not begin until a Gate edge is applied to an armed counter. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. The counter will start counting source edges after the triggering Gate edge, and counting will proceed until the second TC. Note that after application of a triggering Gate edge, the Gate input will be disregarded for the remainder of the count cycle. This differs from Mode K, where the gate can be modulated throughout the count cycle to stop and start the counter. On the first TC after application of the triggering Gate edge, the counter will be reloaded from the Hold register. On the second TC, the counter will be reloaded from the Load register, and counting will stop until a new gate edge is issued to the counter. Note that unlike Mode K, new Gate edges are required after every second TC to continue counting.

MODE N

Software-Triggered Strobe with Level Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode N, shown in Figure 1-16n, provides a software-triggered strobe with level gating that is also hardware retriggerable. The counter must be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the gate is active and disregard those source edges which occur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of the ARM command and the application of an active Gate, the counter will count to TC. Upon reaching TC, the counter will reload from the Load register and automatically disarm itself, inhibiting further counting. Counting will resume upon the issuance of a new ARM command. All active-going Gate edges issued to an armed counter will cause a retrigger operation. Upon application of the Gate edge, the counter contents will be saved in the Hold register. On the first qualified source edge after application of the retriggering gate edge, the contents of the Load register will be transferred into the counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

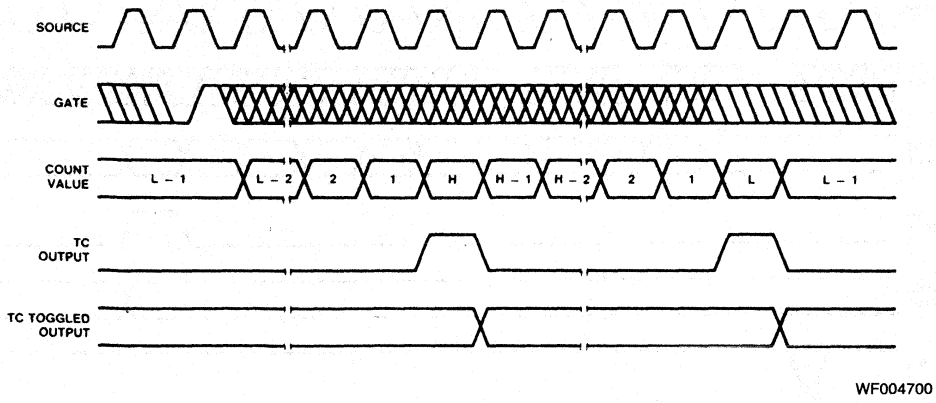


Figure 1-16l. Mode L Waveforms

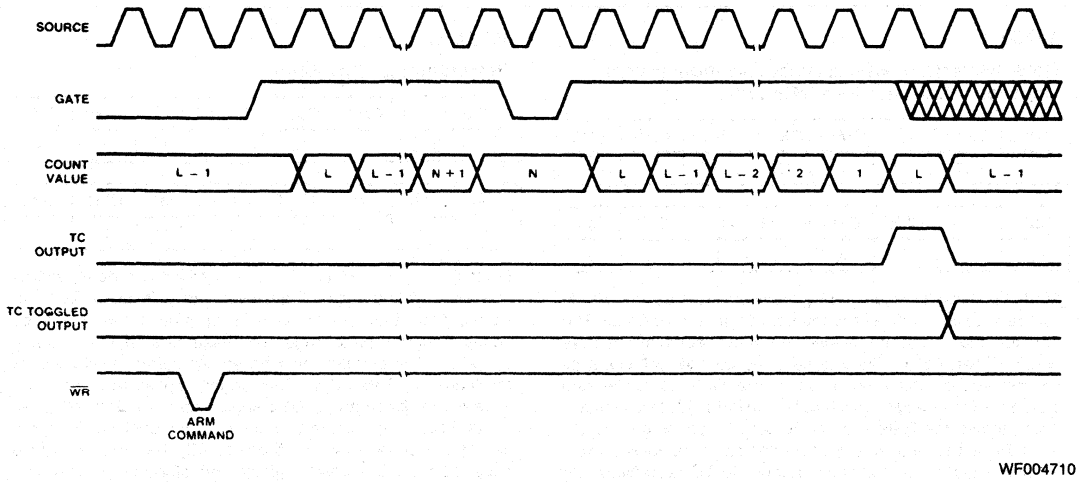


Figure 1-16n. Mode N Waveforms

MODE O

Software-Triggered Strobe with Edge Gating and Hardware Retriggering

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	0	X	X	X	X	X

Mode O, shown in Figure 1-16o, is similar Mode N, except that counting will not begin until an active-going Gate edge is applied to an armed counter and the Gate level is not used to modulate counting. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. Irrespective of the Gate level, the counter will count all source edges after the triggering Gate edge until the first TC. On the first TC,the counter will be reloaded from the Load register and disarmed. A new ARM command and a new Gate edge must be applied in that order to initiate a new counting cycle. Unlike Modes C, F, I and L, which disregard the Gate input once counting starts, in Mode O the count process will be retriggered on all active-going Gate edges, including the first Gate edge used to start the counter. On each retriggering Gate edge, the counter contents will be transferred into the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second-source edge after a retrigger.

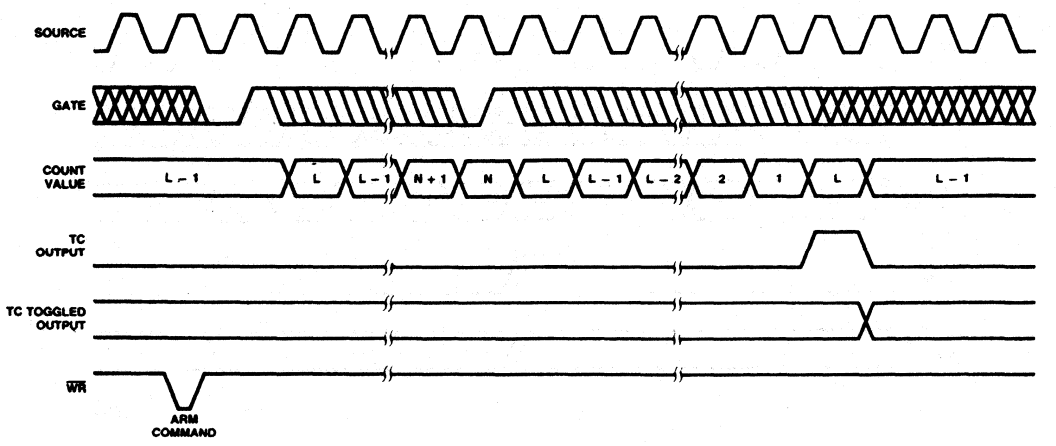


Figure 1-16o. Mode O Waveforms

MODE Q

Rate Generator with Synchronization (Event Counter with Auto-Read/Reset)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
LEVEL			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode Q, shown in Figure 1-16q, provides a rate generator with synchronization or an event counter with auto-read/reset. The counter must first be issued an ARM command before counting can occur. Once armed, the counter will count all source edges which occur while the Gate is active and disregard those edges which cccur while the Gate is inactive. This permits the Gate to turn the count process on and off. After the issuance of an ARM command and the application of an active Gate, the counter will count to TC repetitively. On each TC,the counter will reload itself from the Load register. The counter may be retriggered at any time by presenting an active-going Gate edge to the Gate input. The retriggering Gate edge will transfer the contents of the counter into the Hold register. The first qualified source edge after the retriggering Gate edge will transfer the contents of the Load register into the Counter. Counting will resume on the second qualified source edge after the retriggering Gate edge. Qualified source edges are active-going edges which occur while the Gate is active.

MODE R

Retriggerable One-Shot

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
EDGE			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	0	1	X	X	X	X	X

Mode R, shown in Figure 1-16r, is similar to Mode Q, except that edge gating rather than level gating is used. In other words, rather than use the Gate level to qualify which source edges to count, Gate edges are used to start the counting operation. The counter must be armed before application of the triggering Gate edge; Gate edges applied to a disarmed counter are disregarded. After application of a Gate edge, an armed counter will count all source edges until TC, irrespective of the Gate level. On the first TC, the counter will be reloaded from the Load register and stopped. Subsequent counting will not occur until a new Gate edge is applied. All Gate edges applied to the counter, including the first used to trigger counting, initiate a retrigger operation. Upon application of a Gate edge, the counter contents are saved in the Hold register. On the first source edge after the retriggering Gate edge, the Load register contents will be transferred into the counter. Counting will resume on the second source edge after the retriggering Gate edge.

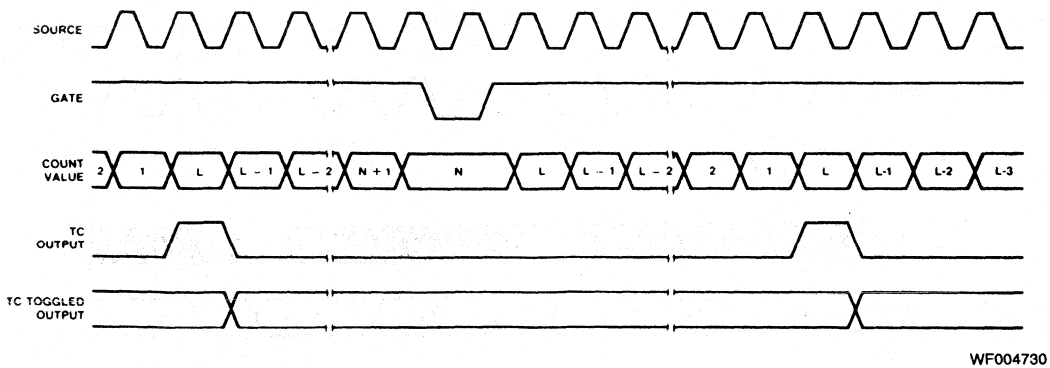


Figure 1-16q. Mode Q Waveforms

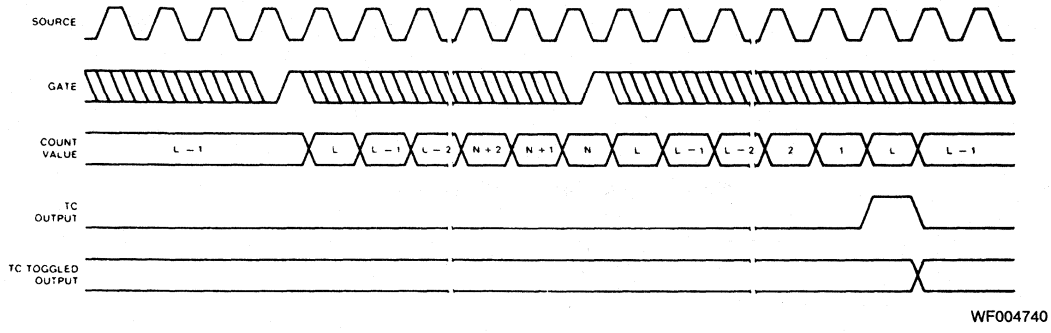


Figure 1-16r. Mode R Waveforms

MODE S

RELOAD SOURCE

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	0	X	X	X	X	X

In this mode, the reload source for LOAD commands (irrespective of whether the counter is armed or disarmed) and for TC-initiated reloads is determined by the Gate input. The Gate input in Mode S is used only to select the reload source, not to start or modulate counting. When the Gate is Low, the Load register is used; when the Gate is High, the Hold register is used. Note the Low-Load, High-Hold mnemonic convention. Once armed, the counter will count to TC twice and then disarm itself. On each TC, the counter will be reloaded from the reload source selected by the Gate. Following the second TC, an ARM command is required to start a new counting cycle. Mode S is shown in Figure 1-16s.

MODE V

Frequency-Shift Keying

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
0	0	0	X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode V, shown in Figure 1-16v, provides frequency-shift keying modulation capability. Gate operation in this mode is identical to that in Mode S. If the Gate is Low, a LOAD command or a TC-induced reload will reload the counter from the Load register. If the Gate is HIGH, LOADs and reloads will occur from the Hold register. The polarity of the Gate only selects the reload source; it does not start or modulate counting. Once armed, the counter will count repetitively to TC. On each TC, the counter will reload itself from the register determined by the polarity of the Gate. Counting will continue in this manner until a DISARM command is issued to the counter. Frequency shift keying may be obtained by specifying a TC Toggled output mode in the Counter Mode register. The switching of frequencies is achieved by modulating the Gate.

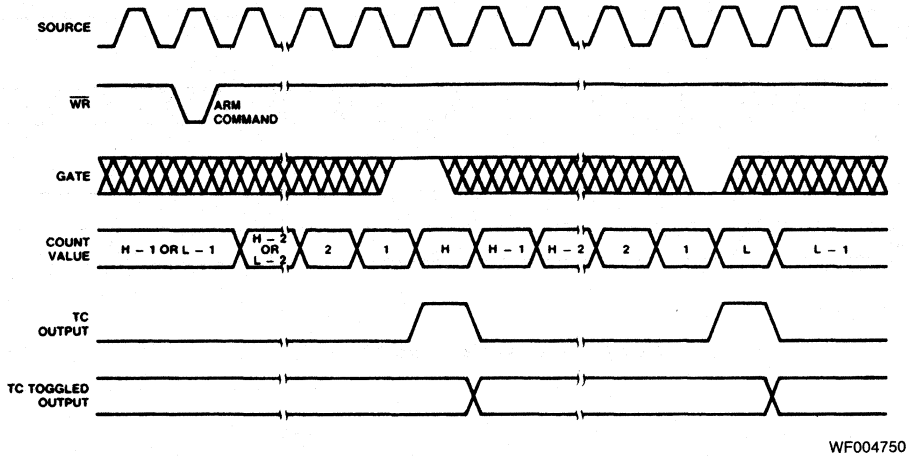


Figure 1-16s. Mode S Waveforms

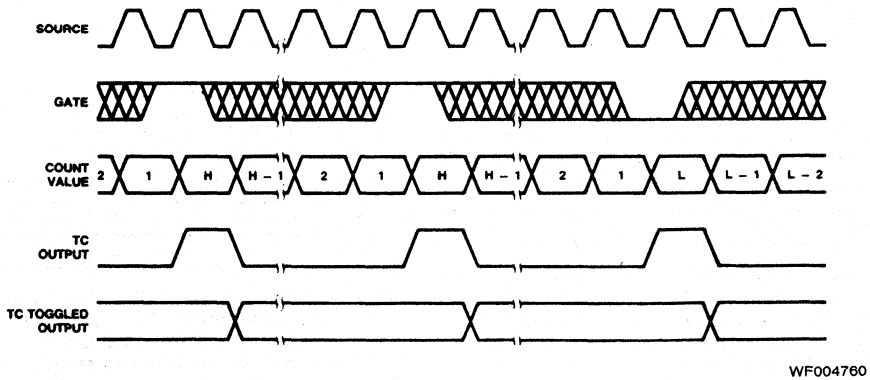


Figure 1-16v. Mode V Waveforms

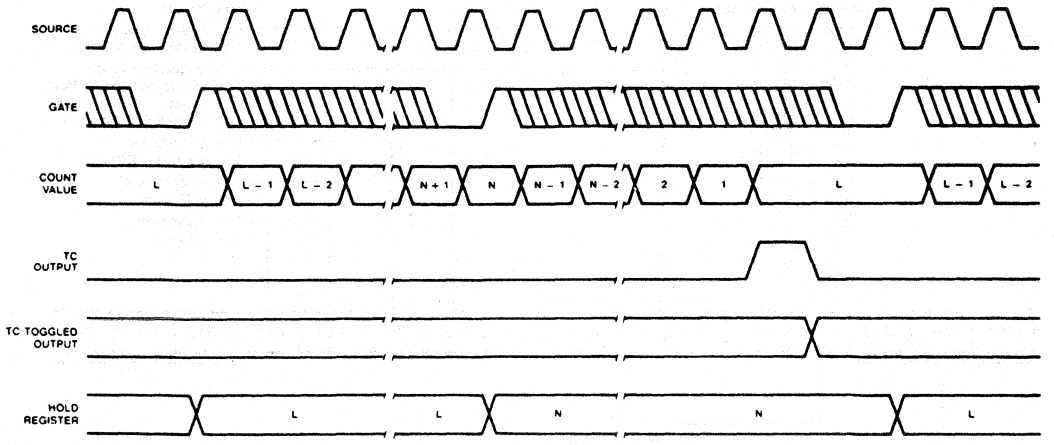


Figure 1-16x. Mode X Waveforms

MODE X

Hardware Save (available in Am9513A only)

CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8
Edge			X	X	X	X	X

CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
1	1	1	X	X	X	X	X

Mode X, as shown in Figure 1-16x, provides a hardware sampling of the counter contents without interrupting the count. A LOAD AND ARM command or a LOAD command followed by an ARM command is required to initialize the counter. Once armed, a Gate edge starts the counting operation; Gate edges applied to a disarmed counter are disregarded. After application of the Triggering Gate edge, the counter will count all qualified source edges until the first TC, irrespective of the gate level. All gate edges applied during the counting sequence will store the current count in the Hold register, but they will not interrupt the counting sequence. On each TC, the counter will be reloaded from the Load register and stopped. Subsequent counting requires a new triggering Gate edge; counting resumes on the first source edge following the triggering Gate edge.

Note: Mode X is only available in the Am9513'A' devices.

COUNTER MODE CONTROL OPTIONS

Each Counter Logic Group includes a 16-bit Counter Mode (CM) register used to control all of the individual options available with its associated general counter. These options include output configuration, count control, count source and gating control. Figure 1-17 shows the bit assignments for the Counter Mode registers. This section describes the control options in detail. Note that generally each counter is independently configured and does not depend on information outside its Counter Logic Group. The Counter Mode register should be loaded only when the counter is Disarmed. Attempts to load the Counter Mode register when the counter is armed may result in erratic counter operation.

After power-on reset or a Master Reset command, the Counter Mode registers are initialized to a preset condition. The value entered is 0B00 hex and results in the following control configuration:

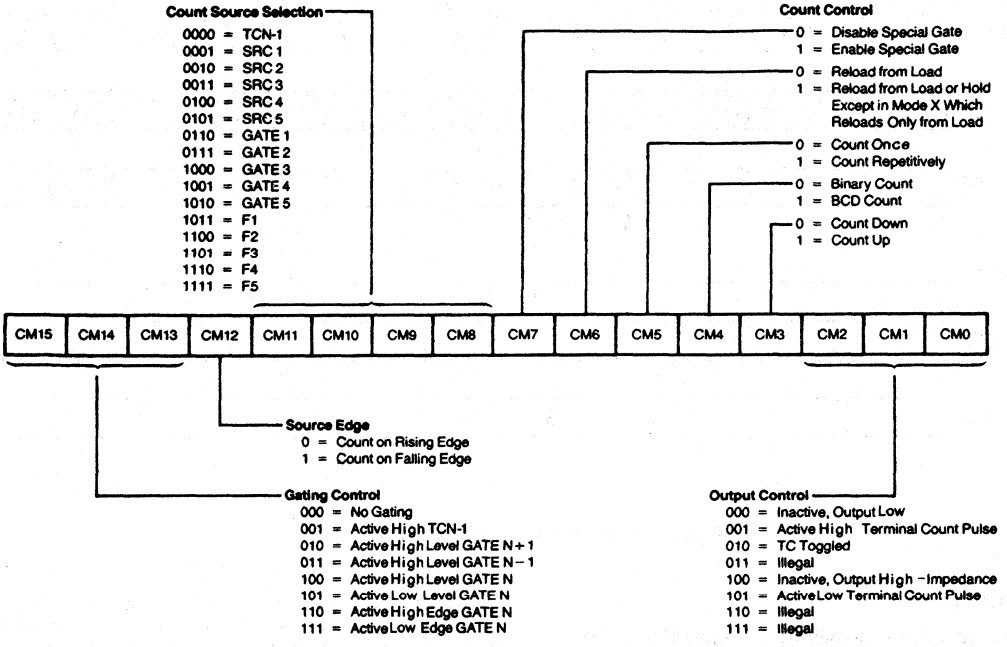
Output low-impedance to ground
Count down
Count binary
Count once
Load register selected
No retriggering
F1 input source selected
Positive-true input polarity
No gating

Output Control

Counter mode bits CM0 through CM2 specify the output control configuration. Figure 1-18 shows a schematic representation of the output control logic. The OUT pin may be off (a high-impedance state), or it may be inactive with a low-impedance to ground. The three remaining valid combinations represent the active-high, active-low or TC Toggle output waveforms.

One output form available is called Terminal Count (TC) and represents the period in time that the counter reaches an equivalent value of zero. TC will occur on the next count when the counter is at 0001 for down counting, at 9999 (BCD) for BCD up counting or at FFFF (hex) for binary up counting. Figure 1-19 shows a Terminal Count pulse and an example context that generated it. The TC width is determined by the period of the counting source. Regardless of any gating input or whether the counter is Armed or Disarmed, the terminal count will go active for only one clock cycle. Figure 1-19 assumes active-high source polarity, counter armed, counter decrementing and an external reload value of K.

The counter will always be loaded from an external location when TC occurs; the user can choose the source location and the value. If a non-zero value is picked, the counter will never really attain a zero state, and TC will indicate the counter state that would have been zero had no parallel transfer occurred.



Note: See Figure 1-16 for restrictions on Count Control and Gating Control bit combinations.

Figure 1-17. Counter Mode Register Bit Assignments

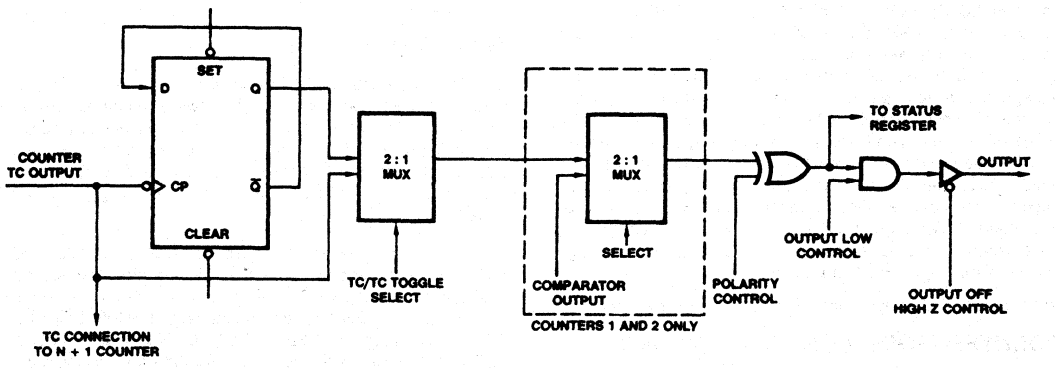


Figure 1-18. Output Control Logic

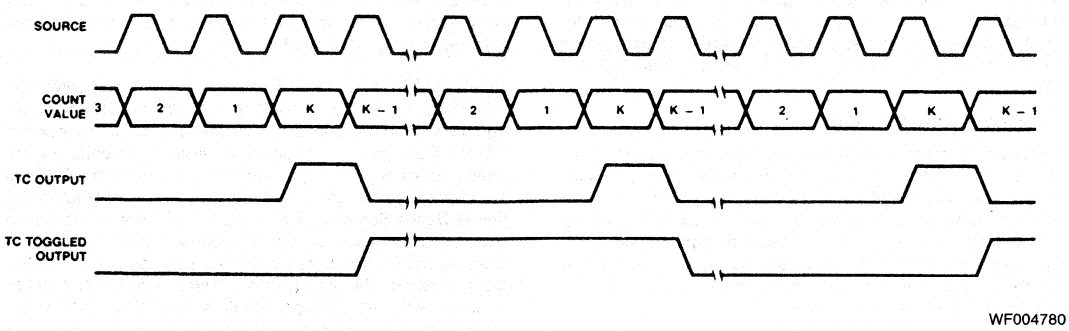


Figure 1-19. Counter Output Waveforms

The other output form, TC Toggled, uses the trailing edge of TC to toggle a flip-flop to generate an output level instead of a pulse. The toggle output is 1/2 the frequency of TC. The TC Toggled output will frequently be used to generate variable duty-cycle square waves in Operating Modes G through K.

In Mode L the TC Toggled output can be used to generate a one-shot function, with the delay to the start of the output pulse and the width of the output pulse separately programmable. With selection of the minimum delay to the start of the pulse, the output will toggle on the second source pulse following application of the triggering Gate edge.

Note that the TC Toggled output form contains no implication about whether the output is active-high or active-low. Unlike the TC output, which generates a transient pulse which can clearly be active-high or active-low, the TC Toggled output waveform only flips the state of the output on each TC. The sole criterion of whether the TC Toggled output is active-high or active-low is the level of the output at the start of the count cycle. This can be controlled by the Set and Clear Output commands. (See Figure 1-20.)

TC (Terminal Count)

On each Terminal Count (TC), the counter will reload itself from the Load or Hold register. TC is defined as that period of time when the counter contents would have been zero had no reload occurred. Some special conditions apply to counter operation immediately before and during TC.

1. In the clock cycle before TC, an internal signal is generated that commits the counter to go to TC on the next count, and retriggering by a hardware Gate edge (Modes N, O, Q and R) or a software LOAD or LOAD AND ARM command will not extend the time to TC. Note that the "next count" driving the counter to TC can be caused by the application of a count source edge (in level gating modes, the edge must occur while the gate is active, or it will be disregarded), by the application of a LOAD or LOAD AND ARM command (see 2 below) or by the application of a STEP command.
2. If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will immediately go to TC. If these commands are issued during TC, the TC state will immediately terminate.
3. When TC is active, the counter will always count the next source edge issued to it, even if it is disarmed or gated off during TC. This means that TC will never be active for longer than one count period and it may, in fact, be shorter if a STEP command or a LOAD or LOAD AND ARM command is applied during TC (see item 2 above). This also means that a counter that is disarmed or stopped on TC is actually disarmed/stopped immediately following TC.

This may cause count sequences different from what a user might expect. Since the counter is always reloaded at the start of TC and since it always counts at the end of TC, the counter contents following TC will differ by one from the reloaded value, irrespective of the operating mode used.

If the reloaded value was 0001 for down counting, 9999 (BCD) for BCD up counting or FFFF (hex) for binary up counting, the count at the end of TC will drive the counter into TC again regardless of whether the counter is gated off or disarmed. As long as these values are reloaded, the TC output will stay active. If a TC Toggled output is selected, it will toggle on each count. Execution of a LOAD, LOAD AND ARM or STEP command with these counter contents will act the same as application of a source pulse, causing TC to remain active and a TC Toggled output to toggle.

Count Control

Counter Mode bits CM3 through CM7 specify the various options available for direct control of the counting process. CM3 and CM4 operate independently of the others and control up/down and BCD/binary counting. They may be combined freely with other control bits to form many types of counting configurations. The other three bits and the Gating Control field interact in complex ways. Bit CM5 controls the repetition of the count process. When CM5 = 1, counting will proceed in the specified mode until the counter is disarmed. When CM5 = 0, the count process will proceed only until one full cycle of operation occurs. This may occur after one or two TC events. The counter is then disarmed automatically. The single or double TC requirement will depend on the state of other control bits. Note that even if the counter is automatically disarmed upon a TC, it always counts the count source edge which generates the trailing TC edge.

When TC occurs, the counter is always reloaded with a value from either the Load register of the Hold register. Bit CM6 specifies the source options for reloading the counter. When CM6 = 0, the contents of the Load register will be transferred into the counter at every occurrence of TC. When CM6 = 1, the counter reload location will be either the Load or Hold Register. The reload location in this case may be controlled externally by using a Gate pin (Modes S and V) or may alternate on each TC (Modes G through L). With alternating sources and with the TC Toggled output selected, the duty cycle of the output waveform is controlled by the relative Load and Hold values and very fine resolution of duty cycles ratios may be achieved.

Bit CM7 controls the special gating functions that allow retriggering and the selection of Load or Hold sources for counter reloading. The use and definition of CM7 will depend on the status of the Gating Control field and bits CM5 and CM6.

Hardware Retriggering

Whenever hardware retriggering is enabled (Modes N, O, Q, and R), all active-going Gate edges initiate retrigger operations. On application of the Gate edge, the counter contents will be transferred to the Hold register. On the first qualified source edge after application of the retriggering Gate edge, the Load register contents will be transferred into the counter. (Qualified source edges are edges which occur while the counter is gated on and Armed.)

This means that, if level gating is used, the edge occurring on active-going gate transitions will initiate a retrigger. Similarly, when edge gating is enabled, an edge used to start the counter will also initiate a retrigger. The first count source edge applied after the Gate edge will not increment/decrement the counter but retrigger it.

If a LOAD, LOAD AND ARM, or a STEP Command occurs between the retriggering Gate edge and the first qualified source edge, it will be interpreted as a source edge and transfer the Load register contents into the counter. Thereafter, the counter will count all qualified source edges.

When some form of Gating is specified, CM7 controls hardware retriggering. In this case, when CM7 = 0, hardware retriggering does not occur; when CM7 = 1, the counter is retriggered any time an active-going Gate edge occurs. Retriggering causes the counter value to be saved in the Hold register and the Load register contents to be transferred into the counter.

When No Gating is specified, the definition of CM7 changes. In this case, when CM7 = 0, the Gate input has no effect on the counting; when CM7 = 1, the Gate input specifies the source (selecting either the Load or Hold register) used to reload the counter when TC occurs. Figure 1-15 shows the various available control combinations for these interrelated bits.

Count Source Selection

Counter Mode bits CM8 through CM12 specify the source used as input to the counter and the active edge that is counted. Bit CM12 controls the polarity for all the sources; logic zero counts rising edges and logic one counts falling edges. Bits CM8 through CM11 select 1 of 16 counting sources to route to the counter input. Five of the available inputs are internal frequencies derived from the internal oscillator (see Figure 1-14 for frequency assignments). Ten of the available inputs are interface pins; five are labeled SRC and five are labeled GATE.

The 16th available input is the TC output from the adjacent lower-numbered counter. (The Counter 5 TC wraps around to the Counter 1 input.) This option allows internal concatenating that permits very long counts to be accumulated. Since all five counters may be concatenated, it is possible to configure a counter that is 80-bits long on one Am9513A chip. When TCN-1 is the source, the count ripples between the connected counters. External connections can also be made, and can use the toggle bit for even longer counts. This is easily accomplished by selecting a TC Toggled output mode and wiring OUTN to one of the SRC inputs.

Gating Control

Counter Mode bits CM15, CM14, CM13 specify the hardware gating options. When "no gating" is selected (000),the counter will proceed unconditionally as long as it is armed. For any other gating mode, the count process is conditioned by the specified gating configuration.

For a code of 100 in this field, counting can proceed only when the pin labeled GATEN associated with Counter N is at a logic high level. When it goes LOW, counting is simply suspended until the Gate goes HIGH again. A code of 101 performs the same function with an opposite active polarity. Codes 010 and 011 offer the same function as 100, but specify alternate input pins as Gating Sources. This allows any of three interface pins to be used as gates for a given counter. On Counter 4, for example, pin 34, pin 35 or pin 36 may be used to perform the gating function. This also allows a single Gate pin to simultaneously control up to three counters. Counters 1 and 5 are considered adjacent when using TCN - 1 (001), Gate N + 1 (010) and Gate N – 1 (011) controls.

For codes of 110 or 111 in this field, counting proceeds after the specified active Gate edge until one or two TC events occur. Within this interval, the Gate input is ignored, except for the retriggering option. When repetition is selected, a cycle will be repeated as soon as another Gate edge occurs. With repetition selected, any Gate edge applied after TC goes active will start a new count cycle. Edge gating is useful when implementing a digital single-shot since the gate can serve as a convenient firing trigger.

A 001 code in this field selects the TC (not TOGGLE) output from the adjacent lower-numbered counter as the gate. This is useful for synchronous counting when adjacent counters are concatenated.

COMMAND DESCRIPTIONS

The command set for the Am9513A allows the host processor to customize and manage the operating modes and features for particular applications, to initialize and update both the internal data and control information, and to manipulate operating bits during operation. Commands are entered directly into the 8-bit Command register by writing into the Control port (see Figure 1-7).

All available commands are described in the following text. Figure 1-20 summarizes the command codes and includes a brief description of each function. Figure 1-21 shows all the unused code combinations; unused codes should not be entered into the Command register since undefined activities may occur.

Six of the command types are used for direct software control of the counting process and they each contain a 5-bit S field. In a linear-select fashion, each bit in the S field corresponds to one of five general counters (S1 = Counter 1, S2 = Counter 2, etc.). When an S bit is a one, the specified operation is performed on the counter so designated; when an S bit is a zero, no operation occurs for the corresponding counter. This type of command format has three basic advantages. It saves host software by allowing any combination of counters to be acted on by a single command. It allows simultaneous action on multiple counters where synchronization of commands is important. It allows counter-specific service routines to control individual counters without needing to be aware of the operating context of other counters.

Three of the commands use a 3-bit binary code (N4, N2, N1) to identify the affected counter (a 001 programs counter 1, etc.). Unlike the previously mentioned commands, these commands allow you to program only one counter at a time.

Command Code								Command Description
C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	E2	E1	G4	G2	G1	Load Data Pointer register with contents of E and G fields. (G ≠ 000, G ≠ 110)
0	0	1	S5	S4	S3	S2	S1	Arm counting for all selected counters
0	1	0	S5	S4	S3	S2	S1	Load contents of specified source into all selected counters
0	1	1	S5	S4	S3	S2	S1	Load and Arm all selected counters*
1	0	0	S5	S4	S3	S2	S1	Disarm and Save all selected counters
1	0	1	S5	S4	S3	S2	S1	Save all selected counters in Hold register
1	1	0	S5	S4	S3	S2	S1	Disarm all selected counters
1	1	1	0	1	N4	N2	N1	Set Toggle out (HIGH) for counter N (001 ≤ N ≤ 101)
1	1	1	0	0	N4	N2	N1	Clear Toggle out (LOW) for counter N (001 ≤ N ≤ 101)
1	1	1	1	0	N4	N2	N1	Step counter N (001 ≤ N ≤ 101)
1	1	1	0	1	0	0	0	Set MM14 (Disable Data Pointer Sequencing)
1	1	1	0	1	1	1	0	Set MM12 (Gate off FOUT)
1	1	1	0	1	1	1	1	Set MM13 (Enter 16-bit bus mode)
1	1	1	0	0	0	0	0	Clear MM14 (Enable Data Pointer Sequencing)
1	1	1	0	0	1	1	0	Clear MM12 (Gate on FOUT)
1	1	1	0	0	1	1	1	Clear MM13 (Enter 8-bit bus mode)
1	1	1	1	1	0	0	0	Enable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	0	0	1	Disable Prefetch for Write operations (Am9513'A' only)
1	1	1	1	1	1	1	1	Master reset

*Not to be used for asynchronous operations.

Figure 1-20. Am9513A Command Summary

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	1
0	0	0	X	X	1	1	0
0	0	0	X	X	0	0	0
*1	1	1	1	1	X	X	X

*Unused except when XXX = 111, 001 or 000.

Figure 1-21. Am9513A Unused Command Codes

Arm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be enabled for counting. A counter must be armed before counting can commerce. Once armed, the counting process may be further enabled or disabled using the hardware gating facilities. This command can only arm or do nothing for a given counter; a zero in the S field does not disarm the counter.

ARM and DISARM commands can be used to gate counter operation on and off under software control. DISARM commands entered while a counter is in the TC state will not take effect until the counter leaves TC. This ensures that the counter never latches up in a TC state. (The counter may leave the TC state because of application of a count source edge, execution of a LOAD or LOAD AND ARM command, or execution of a STEP command.)

In modes which alternate reload sources (Modes G-L), the ARMing operation is used as a reset for the logic which determines which reload source to use on the upcoming TC. Following each ARM or LOAD AND ARM command, a counter in one of these modes will reload from the Hold register on the first TC and alternate reload sources thereafter (reload from the Load register on the second TC, the Hold register on the third, etc.).

Load Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be loaded with previously entered values. The source of information for each counter will be either the associated Load register or the associated Hold register, as determined by the operating configuration in the Mode register. The Load/Hold contents are not changed. This command will cause a transfer independent of any current operating configuration for the counter. It will often be used as a software retrigger or as counter initialization prior to active hardware gating.

If a LOAD or LOAD AND ARM command is executed during the cycle preceding TC, the counter will go immediately to TC. This occurs because the LOAD operation is performed by generating a pseudo-count pulse internal to the Am9513A, and the Am9513A is expecting to go into TC on the next count pulse. The reload source used to reload the counter will be the same as that which would have been used if the TC were generated by a source edge rather than by the LOAD operation.

Execution of a LOAD or LOAD AND ARM command while the counter is in TC will cause the TC to end. For Armed counters in all modes except S or V, the LOAD source used will be that to be used for the upcoming TC. (The LOADing operation will not alter the selection of reload source for the upcoming TC.) For Disarmed counters in modes except S or V, the reload sources used will be the LOAD register. For modes S or V, the reload source will be selected by the GATE input, regardless of whether the counter is Armed or Disarmed.

Special considerations apply when modes with alternating reload sources are used (Modes G-L). If a LOAD command drives the counter to TC in these modes, the reload source for the next TC will be from the opposite reload location. In other words, the LOAD-generated TC will cause the reload sources to alternate just as a TC generated by a source edge would. Note that if a second LOAD command is issued during the LOAD-generated TC (or during any other TC, for that matter), the second LOAD command will terminate the TC and cause a reload from the source designated for use with the next TC. The second LOAD will not alter the reload source for the next TC since the second LOAD does not generate a TC; reload sources alternate on TCs only, not on LOAD commands.

Load and Arm Counters*

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified in the S field, will be first loaded and then armed. This command is equivalent to issuing a LOAD command and then an ARM command.

A LOAD AND ARM command which drives a counter to TC generates the same sequence of operations as execution of a LOAD command and then an ARM command. In modes which disarm on TC (Modes A-C and N-O, and Modes G-I and S if the current TC is the second in the cycle), the ARM part of the LOAD AND ARM command will re-enable counting for another cycle. In modes which alternate reload sources (Modes G-L), the ARMing operation will cause the next TC to reload from the HOLD register, irrespective of which reload source the current TC used.

*This command should not be used during asynchronous operations.

Disarm Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disabled from counting. A disarmed counter will cease all counting independent of other conditions. The only exception to this is that a counter in the TC state will always count once, in order to leave TC, before DISARMing. This count may be generated by a source edge, by a LOAD or LOAD AND ARM command (the LOAD AND ARM command will negate the DISARM command) or by a STEP command. A disarmed counter may be updated using the LOAD command and may be read using the SAVE command. A count process may be resumed using an ARM command. See the ARM command description for further details.

Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by S field, will have their contents transferred into their associated Hold register. The transfer takes place without interfering with any counting that may be underway. This command will overwrite any previous Hold register contents. The SAVE command is designed to allow an accumulated count to be preserved so that it can be read by the host CPU at some later time.

Disarm and Save Counters

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	S5	S4	S3	S2	S1

Description: Any combination of counters, as specified by the S field, will be disarmed, and the contents of the counter will be transferred into the associated Hold registers. This command is identical to issuing a DISARM command followed by a SAVE command.

Set TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: The initial output level for TC Toggle mode is set (HIGH) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 1-18), but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Clear TC Toggle Output

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: The initial output level for TC Toggle mode is Cleared (LOW) for counter N selected by N4, N2, N1 = 001 (Counter 1) thru 101 (Counter 5) respectively. This command conditions the TC Toggle flip-flop (see Figure 1-18, but does not appear at the counter output unless TC Toggle mode (CM2, CM1, CM0 = 010) is selected.

Step Counter

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	0	N4	N2	N1

$(001 \leqslant N \leqslant 101)$

Description: Counter N is incremented or decremented by one, depending on its operating configuration. If the Counter Mode register associated with the selected counter has its CM3 bit cleared to zero, this command will cause the counter to decrement by one. If CM3 is set to a logic high, this command will increment the counter by one. The STEP command will take effect even on a disarmed counter.

Load Data Pointer Register

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	E2	E1	G4	G2	G1

(G4, G2, G1 $\neq$ 000, $\neq$ 110)

Description: Bits in the E and G fields will be transferred into the corresponding Element and Group fields of the Data Pointer register as shown in Figure 1-8. The Byte Pointer bit in the Data Pointer register is set. Transfers into the Data Pointer only occur for G field values of 001, 010, 011, 100, 101 and 111. Values of 000 and 110 for G should not be used. See the "Setting the Data Pointer Register" section of this document for additional details.

Disable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	0	0	0

Description: This command sets Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Disabling the sequencing allows repetitive host processor access to a given internal location without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register.

Enable Data Pointer Sequencing

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	0	0	0

Description: This command clears Master Mode bit 14 without affecting other bits in the Master Mode register. MM14 controls the automatic sequencing of the Data Pointer register. Enabling the sequencing allows sequential host processor access to several internal locations without repetitive updating of the Data Pointer. MM14 may also be controlled by loading a full word into the Master Mode register. See the "Data Pointer Register" section of this document for additional information on Data Pointer sequencing

Enable 16-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	1

Description: This command sets Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is set, no multiplexing takes place and all 16 external data bus lines are used to transfer information into and out of the STC. MM13 may also be controlled by loading the full Master Mode register in parallel.

Enable 8-Bit Data Bus

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	1

Description: This command clears Master Mode bit 13 without affecting other bits in the Master Mode register. MM13 controls the multiplexer in the data bus buffer. When MM13 is cleared, the multiplexer is enabled and 16-bit internal information is transferred eight bits at a time to the eight low-order external data bus lines. MM13 may also be controlled by loading the full Master Mode register in parallel.

Gate Off FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	1	1	1	0

Description: This command sets Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output state of the FOUT signal. When gated off, the FOUT line will exhibit a low-impedance to ground. MM12 may also be controlled by loading the full Master Mode register in parallel.

Gate On FOUT

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	0	0	1	1	0

Description: This command clears Master Mode bit 12 without affecting other bits in the Master Mode register. MM12 controls the output status of the FOUT signal. When MM12 is cleared, FOUT will become active and will drive out the selected and divided FOUT signal. MM12 may also be controlled by loading the full Master Mode register in parallel. When FOUT is gated on or off, a transient pulse may be generated on the FOUT signal.

Disable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	1

Description: This command disables the prefetch circuitry during Write operations (if does not affect Read operations). This reduces the write recovery time and allows the user to use block move instructions for initialization of the Am9513A registers. Once prefetch is disabled for writing, an Enable Prefetch for Write or a Reset command is necessary to re-enable the prefetch circuitry for writing. Note: This command is only available in Am9513'A' devices; it is an illegal command in the "non-A Am9513" device.

Enable Prefetch for Write Operations

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	0	0	0

Description: This command re-enables the prefetch circuitry for Write operations. It is used only to terminate the Disable Prefetch Command. Note: This command is only available in Am9513'A' devices; it is an illegal command in the "non-A Am9513" device.

Master Reset

Coding:

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	1	1	1	1	1

Description: The Master Reset command duplicates the action of the power-on reset circuitry. It disarms all counters, enters 0000 in the Master Mode, Load and Hold registers and enters 0B00 (hex) in the Counter Mode registers.

Following either a power-up or software reset, the LOAD command should be applied to all the counters to clear any that may be in a TC state. The Data Pointer register should also be set to a legal value, since reset does not initialize it. A complete reset operation follows.

1. Using the procedure given in the "Command Initiation" section of this document, enter the FF (hex) command to perform a software reset.
2. Using the "Command Initiation" procedure, enter the LOAD command for all counters, opcode 5F (hex).
3. Using the procedure given in the "Setting the Data Pointer Register" section of this document, set the Data Pointer to a valid code. The legal Data Pointer codes are given in Figure 1-9.

The Master Mode, Counter Mode, Load and Hold registers can now be initialized to the desired values.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5V to +7.0V
All Signal Voltages
with Respect to VSS ..... −0.5V to +7.0V
Power Dissipitation (Package Limitation) 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

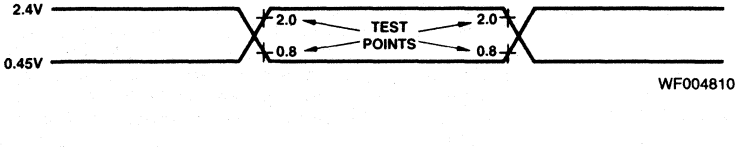
Grade	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5V ±5%	0V
Industrial	−40°C to 85°C	5V ±10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

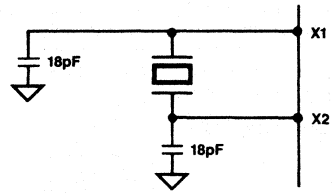
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description		Test Conditions	Min	Typ	Max	Units
VIL	Input Low Voltage	All Inputs Except X2		VSS − 0.5		0.8	Volts
		X2 Input		VSS − 0.5		0.8	
VIH	Input High Voltage	All Input Except X2		2.2V		VCC	Volts
		X2 Input		3.8		VCC	
VITH	Input Hysteresis (SRC and GATE Inputs Only)			0.2	0.3		Volts
VOL	Output Low Voltage		IOL = 3.2mA			0.4	Volts
VOH	Output High Voltage		IOH = −200μA	2.4			Volts
IIX	Input Load Current (Except X2)		VSS ≤ VIN ≤ VCC			±10	μA
IIX	Input Load Current X2					±100	μA
IOZ	Output Leakage Current (Except X1)		VSS+ 0.4 ≤ VOUT ≤ VCC High-Impedance State			±25	μA
ICC	VCC Supply Current (Steady State)		T_A = −55°C			275	mA
			T_A = 0°C			255	
			T_A = +25°C		190	235	
CIN	Input Capacitance		f = 1MHz, T_A = +25°C. All pins not under test at 0V.			10	pF
COUT	Output Capacitance					15	
CIO	IN/OUT Capacitance					20	

SWITCHING TEST INPUT/OUTPUT WAVEFORMS



WF004810



TC002000

Crystal is fundamental mode parallel resonant 32pF load capacitance less than 100Ω ESR C_o less than 100pF.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Notes 2, 3, 4)

Parameters	Description		Figure	Am9513A Min	Am9513A Max	AmZ8073A Min	AmZ8073A Max	Units
TAVRL	C/$\overline{D}$ Valid to Read Low		23	25		25		ns
TAVWH	C/$\overline{D}$ Valid to Write High		23	170		170		ns
TCHCH	X2 High to X2 High (X2 Period)		24	145		145		ns
TCHCL	X2 High to X2 Low (X2 High Pulse Width)		24	70		70		ns
TCLCH	X2 Low to X2 High (X2 Low Pulse Width)		24	70		70		ns
TDVWH	Data In Valid to Write High		23	80		80		ns
TEHEH	Count Source High to Count Source High (Source Cycle Time) (Note 10)		24	145		145		ns
TEHEL TELEH	Count Source Pulse Duration (Note 10)		24	70		70		ns
TEHFV	Count Source High to FOUT Valid (Note 10)		24		500		500	ns
TEHGV	Count Source High to Gate Valid (Level Gating Hold Time) (Notes 10, 12, 13)		24	10		10		ns
TEHRL	Count Source High to Read Low (Set-up Time) (Notes 5, 10)		23	190		190		ns
TEHWH	Count Source High to Write High (Set-up Time) (Notes 6, 10)		23	-100		-100		ns
TEHYV	Count Source High to Out Valid (Note 10)	TC Output	24		300		300	ns
		Immediate or Delayed Toggle Output	24		300		300	
		Comparator Output	24		350		350	
TFN	FN High to FN + 1 Valid (Note 14)		24		75		75	ns
TGVEH	Gate Valid to Count Source High (Level Gating Set-up Time) (Notes 10, 12, 13)		24	100		100		ns
TGVGV	Gate Valid to Gate Valid (Gate Pulse Duration) (Notes 11, 13)		24	145		145		ns
TGVWH	Gate Valid to Write High (Notes 6, 13)		23	-100		-100		ns
TRHAX	Read High to C/$\overline{D}$ Don't Care		23	0		0		ns
TRHEH	Read High to Count Source High (Notes 7, 10)		23	0		0		ns
TRHQX	Read High to Data Out Invalid		23	10		10		ns
TRHQZ	Read High to Data Out at High-Impedance (Data Bus Release Time)		23		85		85	ns
TRHRL	Read High to Read Low (Read Recovery Time)		23	1000		1000		ns
TRHSH	Read High to $\overline{CS}$ High (Note 15)		23	0		0		ns
TRHWL	Read High to Write Low (Read Recovery Time)		23	1000		1000		ns
TRLQV	Read Low to Data Out Valid		23		110		110	ns
TRLQX	Read Low to Data Bus Driven (Data Bus Drive Time)		23	20		20		ns
TRLRH	Read Low to Read High (Read Pulse Duration) (Note 15)		23	160		160		ns
TSLRL	$\overline{CS}$ Low to Read Low (Note 15)		23	20		20		ns
TSLWH	$\overline{CS}$ Low to Write High (Note 15)		23	170		170		ns
TWHAX	Write High to C/$\overline{D}$ Don't Care		23	20		20		ns
TWHDX	Write High to Data In Don't Care		23	20		20		ns
TWHEH	Write High to Count Source High (Notes 8, 10, 17)		23	550		550		ns
TWHGV	Write High to Gate Valid (Notes 8, 13, 17)		23	475		475		ns
TWHRL	Write High to Read Low (Write Recovery Time)		23	1500*		1000		ns
TWHSH	Write High to $\overline{CS}$ High (Note 15)		23	20		20		ns
TWHWL	Write High to Write Low (Write Recovery Time)		23	1500*		1000		ns
TWHYV	Write High to Out Valid (Note 9, 17)		23		650		650	ns
TWLWH	Write Low to Write High (Write Pulse Duration) (Note 15)		23	150		150		ns

*In short data write mode TWHRL and TWHWL minimun = 1000ns.

Notes:

1. Typical values are for $T_A = 25°C$, nominal supply voltage and nominal processing parameters.
2. Test conditions assume transition times of 10ns or less, timing reference levels of 0.8V and 2.0V and output loading of one TTL gate plus 100pF, unless otherwise noted.
3. Abbreviations used for the switching parameter symbols are given as the letter T followed by four or five characters. The first and third characters represent the signal names on which the measurements start and end. Signal abbreviations used are:

A (Address) = C/$\overline{D}$
C (Clock) = X2
D (Data In) = DB0-DB15
E (Enabled counter source input) = SRC1-SRC5, GATE1-GATE5, F1-F5,TCN-1
F = FOUT
G (Counter gate input) = GATE1-GATE5, TCN-1
Q (Data Out) = DB0-DB15
R (Read) = $\overline{RD}$
S (Chip Select) = $\overline{CS}$
W (Write) = $\overline{WR}$
Y (Output) = OUT1-OUT5

The second and fourth letters designate the reference states of the signals named in the first and third letters respectively, using the following abbreviations.

H = HIGH
L = LOW
V = VALID
X = Unknown or Don't care
Z = High-Impedance

4. Switching parameters are listed in alphabetical order.

5. Any input transition that occurs before this minimum setup requirement will be reflected in the contents read from the status register.

6. Any input transition that occurs before this minimum setup requirement will act on the counter before the execution of the operation initiated by the write and the counter may be off by one count.

7. Any input transition that occurs after this minimum hold time is guaranteed to not influence the contents read from the status register on the current read operation.

8. Any input transition that occurs after this minimum hold time is guaranteed to be seen by the counter as occurring after the action initiated by the write operation and the counter may be off by one count.

9. This parameter applies to cases where the write operation causes a change in the output bit.

10. The enabled count source is one of F1-F5, TCN-1 SRC1-SRC5 or GATE1-GATE 5, as selected in the applicable Counter Mode register. The timing diagram assumes the counter counts on rising source edges. The timing specifications are the same for falling-edge counting.

11. This parameter applies to edge gating (CM15-CM13 = 110 or 111) and gating when both CM7 = 1 and CM15-CM13 ≠ 000. This parameter represents the minimum GATE pulse width needed to ensure that the pulse initiates counting or counter reloading.

12. This parameter applies to both edge and level gating (CM15-CM13 = 001 through 111) and gating when both CM7 = 1 and CM15-CM13 = 000. This parameter represents the minimum setup or hold times to ensure that the Gate input is seen at the intended level on the active source edge and the counter may be off by one count.

13. This parameter assumes that the GATENA input is unused (16-bit bus mode) or is tied high. In cases where the GATENA input is used, this timing specification must be met by both the GATE and GATENA inputs.

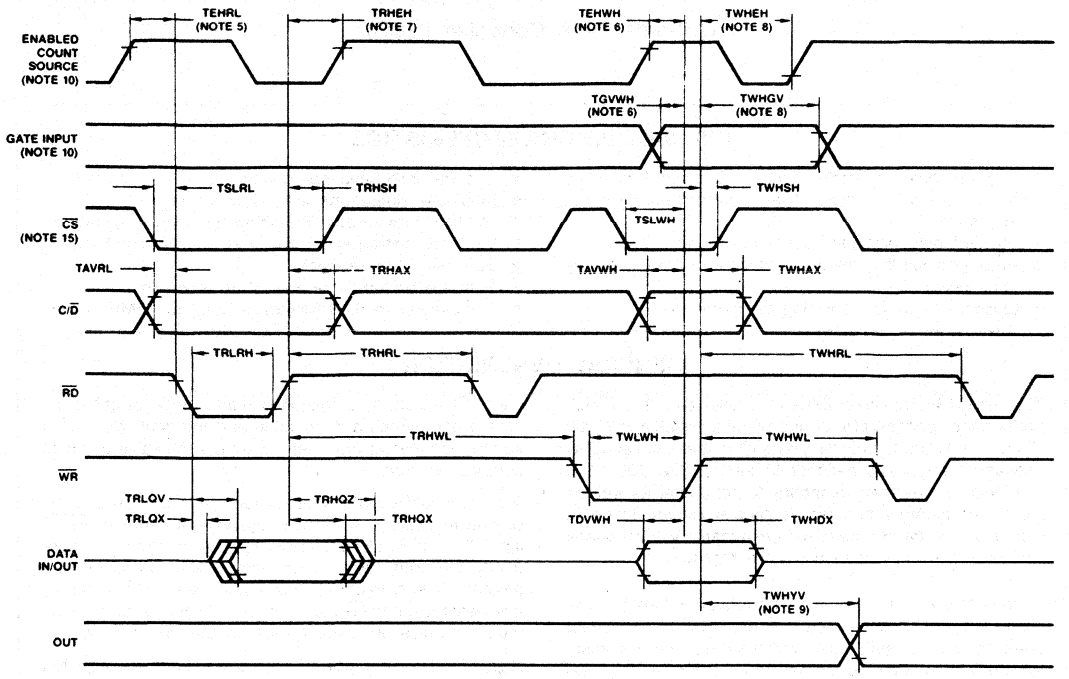
14. Signals F1-F5 cannot be directly monitored by the user. The phase difference between these signals will manifest itself by causing counters using two different F signals to count at different times on nominally simultaneous transitions in the F signals. F1 = X2.

15. This timing specification assumes that $\overline{CS}$ is active whenever $\overline{RD}$ or $\overline{WR}$ are active. $\overline{CS}$ may be held active indefinitely.

16. This parameter assumes X2 is driven from an external gate with a square wave.

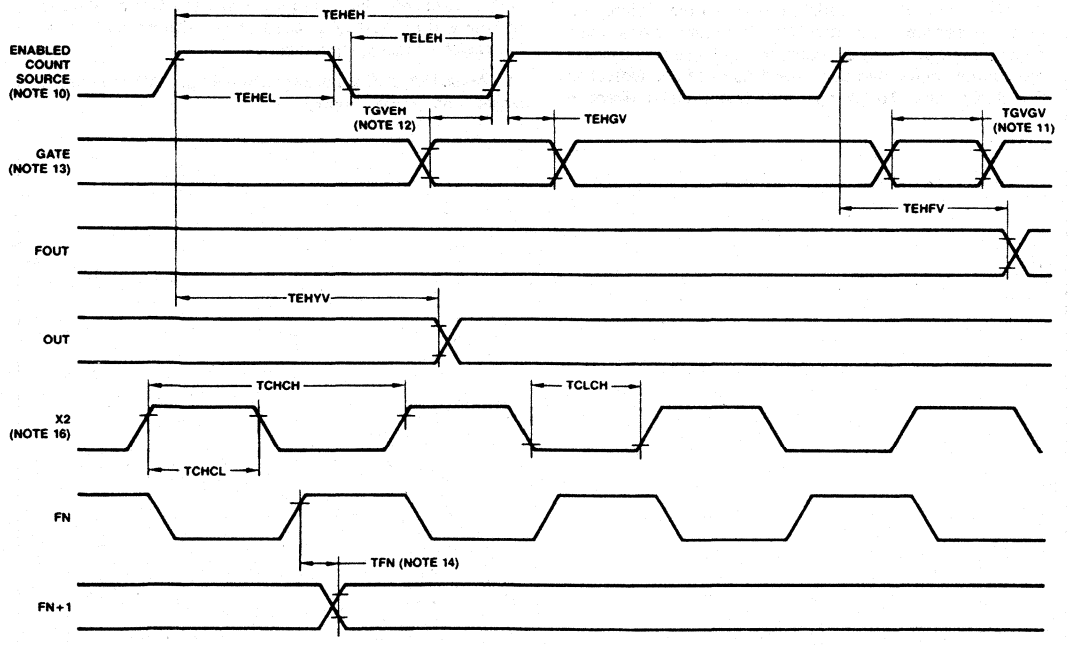
17. This parameter assumes that the write operation is to the command register.

BUS TRANSFER SWITCHING WAVEFORMS



WF004790

COUNTER SWITCHING WAVEFORMS



WF004800

01731D

Am9516A

Universal DMA Controller (UDC)

DISTINCTIVE CHARACTERISTICS

- Transfer Modes: Single, demand dedicated with bus hold, demand dedicated with bus release, demand interleave
- 16 Megabyte physical addressing range
- Automatic loading/reloading of control parameters by each channel
- Optional automatic chaining of operations
- Channel interleave operations
- Interleave operations with system bus
- Masked data pattern matching for search operations
- Vectored interrupts on selected transfer conditions
- Software DMA request
- Software or hardware wait state insertion
- Transfer up to 6.66 Mbytes/second at 10MHz clock

GENERAL DESCRIPTION

The Am9516A Universal DMA Controller (UDC) is a high performance peripheral interface circuit for 8086 and 68000 CPUs. In addition to providing data block transfer capability between memory and peripherals, each of the UDC's two channels can perform peripheral-to-peripheral as well as memory-to-memory transfer. A special Search Mode of Operation compares data read from a memory or peripheral source to the content of a pattern register.

For all DMA operations (search, transfer, and transfer-and-search), the UDC can operate with either byte or word data sizes. In some system configurations it may be necessary to transfer between word-organized memory and a byte-oriented peripheral. The UDC provides a byte packing/unpacking capability through its byte-word funneling transfer or transfer-and-search option. Some DMA applications may continuously transfer data between the same two memory areas. These applications may not require the flexibility inherent in reloading registers from memory tables. To service these repetitive DMA operations, base registers are provided on each channel which re-initialize the current source and destination Address and Operation Count registers. To change the data transfer direction under CPU control, provision is made for reassigning the source address as a destination and the destination as a source, eliminating the need for actual reloading of these address registers.

Frequently DMA devices must interface to slow peripherals or slow memory. In addition to providing a hardware $\overline{\text{WAIT}}$ input, the Am9516A UDC allows the user to select independently for both source and destination addresses and automatic insertion of 0, 1, 2 or 4 wait states. The user may even disable the $\overline{\text{WAIT}}$ input pin function altogether and use these software programmed wait states exclusively.

High throughput and powerful transfer options are of limited usefulness if a DMA requires frequent reloading by the host CPU. The Am9516A UDC minimizes CPU interactions by allowing each channel to load its control parameters from memory into the channel's control registers. The only action required of the CPU is to load the address of the control parameter table into the channel's Chain Address register and then issue a "Start Chain" Command to start the register loading operation.

The Am9516A UDC is packaged in a 48-pin DIP and uses a single +5V Power Supply.

03242C

Refer to page 7-1 for Essential Information on Military Devices

BLOCK DIAGRAM

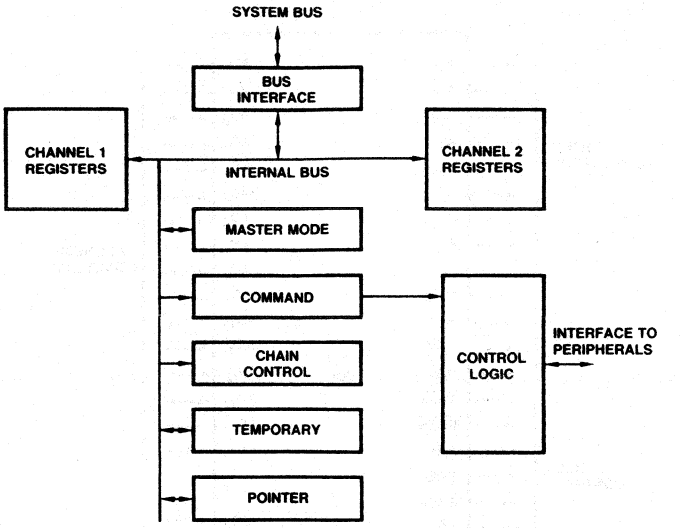
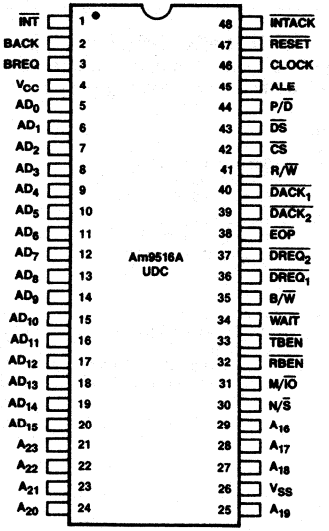


Figure 1.

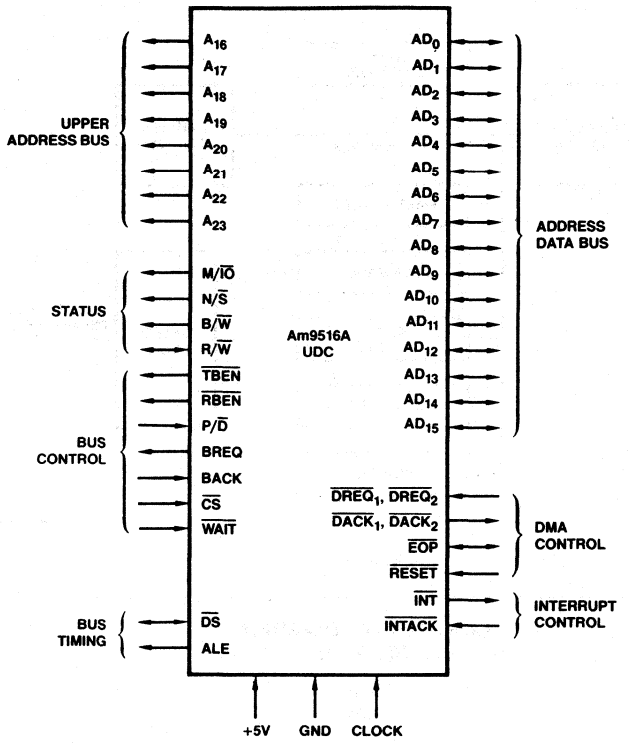
CONNECTION DIAGRAM
Top View

D-48
P-48



Note: Pin 1 is marked for orientation

LOGIC SYMBOL



LS001331

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am9516A –1 D C B

Device Type: Am9516A

Speed:
–4 = 4MHz
–6 = 6MHz
–8 = 8MHz
–1 = 10MHz

48-pin Package:
D = Cerdip side-brazed
P = Plastic
L = Leadless Chip Carrier
J = Plastic Leadless Chip Carrier

Temperature Range:
C = 0°C to +70°C
I = –40°C to +85°C

Screening Option:
B = Burn In
Blank = Std. Processing

Valid Combinations	
Am9516A-4 Am9516A-6 Am9516A-8 Am9516A-1	DC, DCB, DI, DIB PC, PCB, PI, PIB, LC, LCB, LI, LIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	V_{CC}		+5V Power Supply.
26	V_{SS}		Ground.
46	CLOCK	I	Clock.
46	CLOCK	I	(Clock). The Clock signal controls the internal operations and the rates of data transfers. It is usually derived from a master system clock or the associated CPU clock. The Clock input requires a high voltage input signal. Many UDC input signals can make transitions independent of the UDC clock; these signals can be asynchronous to the UDC clock. On other signals, such as $\overline{WAIT}$ inputs, transitions must meet setup and hold requirements relative to the UDC clock. See the Timing diagrams for details.
5-20	$AD_0 - AD_{15}$	I/O	(Address-Data Bus, Three-State). The Address Data Bus is a time-mulitiplexed, bidirectional, active-high, three-state bus used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD_0 is the least significant bit position and AD_{15} is the most significant. The presence of addresses is defined by the timing edge of ALE, and the asserted or requested presence of data is defined by the $\overline{DS}$ signal. The status output lines $M/\overline{IO}$ and $N/\overline{S}$ indicate the type of transaction, either memory or I/O. The $R/\overline{W}$ line indicates the direction of the transaction. When the UDC is in control of the system bus, it dominates the AD Bus; when the UDC is not in control of the system bus, the CPU or other external devices dominate the AD Bus. The presence of address of data on the $AD_0 - AD_{15}$ bus is defined only by ALE and $\overline{DS}$. When the UDC is not in control of the bus, there is no required relation between the presence of address or data and the UDC clock. This allows the UDC to be used with a system bus which does not have a bussed clock signal.
43	$\overline{DS}$	I/O	(Data Strobe, Three-State). Data Strobe is a bidirectional, active-low, three-state signal. A LOW on this signal indicates that the AD_0-AD_{15} bus is being used for data transfer. When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, $\overline{DS}$ is a timing input used by the UDC to move data to or from the $AD_0 - AD_{15}$ bus. Data is written into the UDC by the external ststem on the LOW-to-HIGH $\overline{DS}$ transition. Data is read from the UDC by the external system while $\overline{DS}$ is LOW. There are no timing requirements between $\overline{DS}$ as an input and the UDC clock; this allows use of the UDC with a system bus which does not have a bussed clock. During a DMA operation when the UDC is in control of the ststem, $\overline{DS}$ is an output generated by the UDC and used by the system to move data to or from the $AD_0 - AD_{15}$ bus. When the UDC has bus control, it writes to the external system by placing data on the $AD_0 - AD_{15}$ bus before the HIGH-to LOW $\overline{DS}$ transition and holding the data stable until after the LOW-to-HIGH $\overline{DS}$ transition; while reading from the external system, the LOW-to-HIGH transition of $\overline{DS}$ inputs data from the $AD_0 - AD_{15}$ bus into the UDC (see Timing diagram).
41	$R/\overline{W}$	I/O	(Read/Write, Three-State). Read/Write is a bidirectional, three-state signal. Read polarity is HIGH and write polarity is LOW. $R/\overline{W}$ indicates the data direction of the current bus transaction, and is stable starting when ALE is HIGH until the bus transaction ends (see Timing diagram). When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, $R/\overline{W}$ is a status input used by the UDC to determine if data is entering or leaving on the $AD_0 - AD_{15}$ bus during $\overline{DS}$ time. In such a case, Read (HIGH) indicates that the system is requesting data from the UDC, and Write (LOW) indicates that the system is presenting data to the UDC. There are no timing requirements between $R/\overline{W}$ as an input and the UDC clock; transitions on $R/\overline{W}$ as an input are only defined relative to $\overline{DS}$. When the UDC is in control of the system bus, $R/\overline{W}$ is an output generated by the UDC, with Read indicating that data is being requested from the addressed location or device. the addressed location or device and Write indicating that data is being presented to the addressed location or device. Flyby DMA operations are a special case where $R/\overline{W}$ is valid for the normally addressed memory or peripheral locations and must be interpreted in reverse by the "Flyby" peripheral that uses it.
33	$\overline{TBEN}$	O	(Transmit Buffer Enable, Open Drain). Transmit Buffer Enable is an active-low, open drain output. When UDC is a bus master, a LOW on this output indicates that the data is being transferred from the UDC to the data bus lines through the buffer. The purpose of this signal is to eliminate bus contention. When UDC is not in control of the system bus, these pins float to three-state OFF.
32	$\overline{RBEN}$	O	(Receive Buffer Enable, Open Drain). Receive Buffer Enable is an active-low, open drain output. When UDC is in control of system bus, a LOW on this output indicates that the data is being transferred from the data bus lines to the UDC through the buffer. The purpose of this signal is to eliminate bus contention. This pin floats to three-state OFF when the UDC is not in control of the system bus.
45	ALE	O	(Address Latch Enable). This active HIGH signal is provided by the UDC to latch the address signals $AD_0 - AD_{15}$ into the address latch. This pin is never floated.
44	$P/\overline{D}$	I	(Pointer/Data). Pointer/Data is an input signal to indicate the information is on the $AD_0 - AD_{15}$ bus only when the UDC is the bus slave. A HIGH on this signal indicates the information is on the AD bus is an address of the internal register to be accessed. The data on the AD bus is loaded into the Pointer register of UDC. A LOW on this signal indicates that a data transfer is taking place between the bus and the internal register designated by the Pointer register. Note that if a transaction is carried out with $R/\overline{W}$ HIGH and $P/\overline{D}$ HIGH, the contents of the Pointer register will be read.
31	$M/\overline{IO}$	O	(Memory/Input-Output, Three-State). This signal specifies the type of transaction. A HIGH on this pin indicates a memory transaction. A LOW on this pin indicates an I/O transaction. It floats to three-state OFF when UDC is not in control of the ststem bus.
30	$N/\overline{S}$	O	(Normal/System, Three-State). This output is a three-state signal activated only when the UDC is the bus master. Normal is indicated when $N/\overline{S}$ is HIGH, and System is indicated when $N/\overline{S}$ is LOW. This signal supplements the $M/\overline{IO}$ line and is used to indicate which memory or I/O space is being accessed.
35	$B/\overline{W}$	O	(Byte/Word, Three-State). This output indicates the size of data transferred on the $AD_0 - AD_{15}$ bus. HIGH indicates a byte (8-bit) and LOW indicates a word (16-bit) transfer. This output is activated when ALE is HIGH and remains valid for the duration of the whole transaction (see Timing diagram). All word-sized data are word-aligned and must be addressed by even addresses ($A_0 = 0$). When addressing byte read transactions, the least significant address bit determines which byte is needed; an even address specifies the most significant byte ($AD_8 - AD_{15}$), and an odd address specifies the least significant byte ($AD_0 - AD_7$). (Note that the higher address specifies the least significant byte!) This addressing mechanism applies to memory accesses as well as I/O accesses. When the UDC is a slave, it ignores the $B/\overline{W}$ signal and this pin floats to three-state OFF.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
42	$\overline{CS}$	I	(Chip Select). This pin is an active-low input. A CPU or other external device uses $\overline{CS}$ to activate the UDC for reading and writing of its internal registers. There are no timing requirements between the $\overline{CS}$ input and the UDC clock; the $\overline{CS}$ input timing requirements are only defined relative to $\overline{DS}$. This pin is ignored when UDC is in control of system bus.
34	$\overline{WAIT}$	I	(WAIT). This pin is an active-low input. Slow memories and peripheral devices may use $\overline{WAIT}$ to extend $\overline{DS}$ and $\overline{RBEN}$ or $\overline{TBEN}$ during operation. Unlike the $\overline{CS}$ input, transitions on the $\overline{WAIT}$ input must meet certain timing requirements relative to the UDC clock. See Timing Diagram 4 for details. The Wait function may be disabled using a control bit in the Master Mode register (MM2).
3	BREQ	O	(Bus Request). Bus Request is an active-HIGH signalsignal used by the UDC to obtain control of the bus from the CPU. BREQ lines from multiple devices are connected to a priority encoder.
2	BACK	I	(Bus Acknowledge). BACK is an active-HIGH, asynchronchronous input, indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since BACK is internally synchronized by the UDC before being used, transitions on BACK do not have to be synchronous with the UDC clock. The BACK input is usually connected to the HLDA line from the CPU or to the output of a priority decoder.
1	$\overline{INT}$	O	(Interrupt Request, Open Drain). Interrupt Request is an active-low output used to interrupt the CPU. It is driven LOW whenever the IP and CIE bits of the Status Register are set. It is cleared by UDC after receiving a clear IP command.
48	$\overline{INTACK}$	I	(Interrupt Acknowledge). Interrupt Acknowledge is an active-low input indicating that the request for interrupt has been granted. The UDC will place a vector onto the AD bus if the No Vector or Interrupt bit (MM3) is reset.
47	$\overline{RESET}$	I	(Reset). Reset is an active-low input to disable the UDC and clear its Master Mode register.
36, 37	$\overline{DREQ}_1$, $\overline{DREQ}_2$	I	(DMA Request). The DMA Request lines are two active-low inputs, one per channel. They may make transitions independent of the UDC clock and are used by external logic to initiate and control DMA operations performed by the UDC.
40, 39	$\overline{DACK}_1$, $\overline{DACK}_2$	O	(DMA Acknowledge). The DMA Acknowledge lines are active-low outputs, one per channel, which indicate that the channel is performing a DMA operation. $\overline{DACK}$ is pulsed, held active or held inactive during DMA operations as programmed in the Channel Mode register. For Flowthru operations, the peripheral is fully addressed using the conventional I/O addressing protocols and therefore may choose to ignore $\overline{DACK}$. $\overline{DACK}$ is always output as programmed in the Channel Mode register for a DMA operation, even when the operation is intiated by a CPU software request command or as a result of chaining. $\overline{DACK}$ is not output during the chaining operations.
38	$\overline{EOP}$	I/O	(End of Process). $\overline{EOP}$ is an active-low, open-drain, bidirectional signal. It must be pulled up with an external resistor of 1.8kohm or more. The UDC emits an output pulse on $\overline{EOP}$ when a TC or MC termination occurs, as defined later. An external source may terminate a DMA operation in progress by driving $\overline{EOP}$ low. $\overline{EOP}$ always applies to the active channel; if no channel is active, $\overline{EOP}$ is ignored.
29-27 25-21	$A_{16} - A_{23}$	O	(Upper Address Bus, Three-state). The $A_{16} - A_{23}$ address lines are three-state outputs activated only when the UDC is controlling the system bus. Combined with the lower 16 address bits appearing on AD_0 through AD_{15} respectively, this 24-bit linear address allows the UDC to access anywhere within 16 Megabytes of memory.

Note: All inputs to the UDC, except the clock are directly TTL compatible.

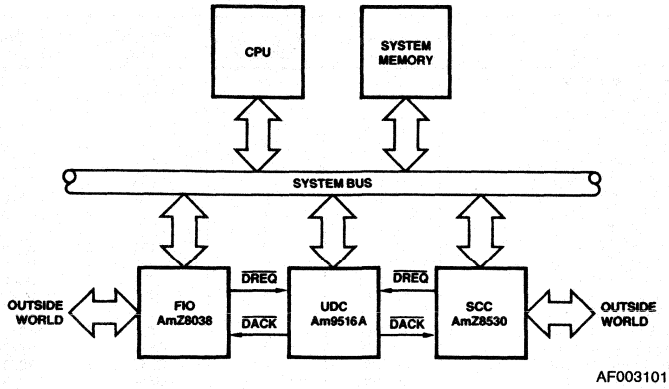


Figure 2. UDC Configurations

PRODUCT OVERVIEW

Register Description

The Am9516A UDC block diagram illustrates the internal registers. Figure 3 lists each register along with its size and read/write access restrictions. Registers which can be read by the CPU are either fast (F) or slow (S) readable. Fast registers can be read by a normal CPU I/O operation without additional wait states. Reading slow registers requires multiple wait states. Registers can be written to by the host CPU (W) and/or can be loaded by the DMA channel itself during chaining (C). All reads or writes must be word accesses since the UDC ignores the B/$\overline{W}$ line in slave mode. It is the responsibility of the user to supply the necessary external logic if slow readable registers are to be read.

The UDC registers can be categorized into chip-level registers, which control the overall operation and configuration of the UDC, and channel-level registers which are duplicated for each channel. The five chip-level registers are the Master Mode register, the Command register, the Chain Control register, the Pointer register, and the Temporary register. The Master Mode register selects the way the UDC chip interfaces to the system. The Command register is written to by the host CPU to initiate certain operations within the UDC chip, such as resetting the unit. The Chain Control register is used by a channel while it is reloading its channel-level registers from memory. The Pointer register is written to by the host CPU when the P/$\overline{D}$ input is HIGH. The data in Pointer register is the address of the internal register to be accessed. The Temporary register is used to hold data for Flowthru Transfer/Transfer-and-Searches.

The channel-level registers can be divided into two subcategories: general purpose registers, which would be found on most DMA chips, and special purpose registers, which provide additional features and functionality. The general purpose registers are the Base and Current Operation Count registers, the Base and Current Address registers A and B, and the Channel Mode register. The special purpose registers are the Pattern and Mask registers, the Status register, the Interrupt Vector register, the Interrupt Save register, and the Chain Address register.

The internal registers are read or written in two steps. First, the address of the register to be accessed is written to the Pointer register, when the P/$\overline{D}$ input is HIGH. Then, the data is read from or written into the desired register, which is indicated by the Pointer register, when P/$\overline{D}$ input is LOW. Note that a read with P/$\overline{D}$ HIGH causes the contents of the Pointer register to be read on AD_1 through AD_6.

Master Mode Register

The 4-bit Master Mode register, shown in Figure 4, controls the chip-level interfaces. It can be read from and written to by the host CPU without wait states through pins $AD_0 - AD_3$, but it is not loadable by chaining. On a reset, the Master Mode register is cleared to all zeroes. The function of each of the Master Mode bits is described in the following paragraphs.

The Chip Enable bit CE = 1 enables the UDC to request the bus. When enabled, the UDC can perform DMA Operations and reload registers. It can always issue interrupts and respond to interrupt acknowledges. When the Chip Enable bit is cleared, the UDC is inhibited from requesting control of the system bus and, therefore, inhibited from performing chaining or DMA operations.

The CPU Interleave bit enables interleaving between the CPU and the UDC.

The Wait Line Enable bit is used to enable sampling of the $\overline{WAIT}$ line during Memory and I/O transactions. Because the UDC provides the ability to insert software programmable wait states, many users may disable sampling of the $\overline{WAIT}$ pin to eliminate the logic driving this pin. The Wait Line Enable bit provides this flexibility. See the "Wait States" section of this document for details on wait state insertion.

The "No Vector on Interrupt" bit selects whether the UDC channel or a peripheral returns a vector during interrupt acknowledge cycles. When this bit is cleared, a channel receiving an interrupt acknowledge will drive the contents of its Interrupt Save register onto the $AD_0 - AD_{15}$ data bus while $\overline{INTACK}$ is LOW. If this bit is set, interrupts are serviced in an identical manner, but the $AD_0 - AD_{15}$ data bus remains in a high-impedance state throughout the acknowledge cycle.

Pointer Register

The Pointer register contains the address of the internal register to be accessed. It can be read from or written to by the CPU when the P/$\overline{D}$ line is HIGH.

Name	Size	Number	Access Type	Port Address CH-1/CH-2
Master Mode Register	4 bits	1	FW	38
Pointer Register	6 bits	1	FW	
Chain-Control Register	10 bits	1	C	
Temporary Register	16 bits	1	D	
Command Register	8 bits	1	W	2E/2C*
Current Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1A/18
Lower Address field	16 bits	2	CFW	0A/08
Current Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	12/10
Lower Address field	16 bits	2	CFW	02/00
Base Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1E/1C
Lower Address field	16 bits	2	CFW	0E/0C
Base Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	16/14
Lower Address field	16 bits	2	CFW	06/04
Current Operation Count	16 bits	2	CFW	32/30
Base Operation Count	16 bits	2	CFW	36/34
Pattern Register	16 bits	2	CSW	4A/48
Mask Register	16 bits	2	CSW	4E/4C
Status Register	16 bits	2	F	2E/2C
Interrupt Save Register	16 bits	2	F	2A/28
Interrupt Vector Register	8 bits	2	CSW	5A/58
Channel Mode Register – HIGH	5 bits	2	CS	56/54
Channel Mode Register – LOW	16 bits	2	CSW	52/50
Chain Address Register:				
Up-Addr/Tag field	10 bits	2	CFW	26/24
Lower Address field	16 bits	2	CFW	22/20

Access Codes: C = Chain Loadable
D = Accessible by UDC channel
F = Fast Readable
S = Slow Readable
W = Writable by CPU

Note: The address of the register to be accessed is stored in the Pointer register.

*Port addresses of the Command register can be used alternately for both channels except when issuing a "set or clear IP" command.

Figure 3. UDC Internal Register

Chain Control Register

When a channel starts a chaining operation, it fetches a Reload word from the memory location pointed to by the Chain Address register (Figure 11). This word is then stored in the Chain Control register. The Chain Control register cannot be written to or read from by the CPU. Once a channel starts a chain operation, the channel will not relinquish bus control until all registers specified in the Reload word are reloaded unless an $\overline{EOP}$ signal is issued to the chip. Issuing an $\overline{EOP}$ to a channel during chaining will prevent the chain operation from resuming and the contents of the Reload Word register can be discarded.

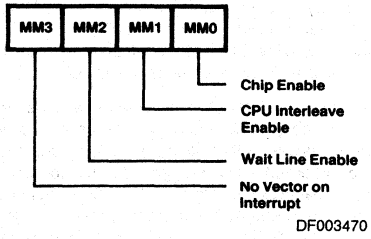


Figure 4. Master Mode Register

Temporary Register

The Temporary register is used to stage data during Flowthru transfers and to hold data being compared during a Search or a Transfer-and-Search. The temporary register cannot be written to or read from by the CPU. In byte-word funneling, data may be loaded into or from the Temporary register on a byte-by-byte basis, with bytes sometimes moving between the low byte of the data bus and the high byte of the Temporary register or vice-versa. See the "Transfer" section for details.

Command Register

The UDC Command register (Figure 20) is an 8-bit write-only register written to by the host CPU. The Command register is loaded from the data on $AD_7 - AD_0$; the data on $AD_{15} - AD_8$ is disregarded. A complete discussion of the commands is given in the "Command Descriptions" section.

Current and Base Address Registers A and B

The Current Address registers A and B (Current ARA and ARB) are used to point to the source and destination addresses for DMA operations. The contents of the Base ARA and ARB registers are loaded into the Current ARA and ARB registers at the end of a DMA operation if the user enables Base-to-Current reloading in the Completion Field of the Channel Mode register. This facilitates DMA operations without reloading of the Current registers. The ARA and ARB registers can be loaded during chaining, can be written to by the host CPU without wait states and can be read by the CPU.

Each of the Base and Current ARA and ARB registers consists of two words organized as a 6-bit Tag Field and an 8-bit Upper Address in one word and a 16-bit Lower Address in the other. See Figure 5. The Tag Field selects whether the address is to be incremented, decremented or left unchanged, and the status codes associated with the address. The Tag field also allows the user to insert 0, 1, 2 or 4 wait states into memory or I/O accesses addressed by the offset and segment fields.

The Address Reference Select Field in the Tag field selects whether the address pertains to memory space or I/O space. Note that the N/$\overline{S}$ output pin may be either HIGH (indicating Normal) or LOW (indicating System) for space. At the end of each iteration of a DMA Operation, the user may select to leave the address unchanged or to increment it or to decrement it. I/O addresses, if changed, are always incremented/decremented by 2. Memory addresses are changed by 1 if the address points to a byte operand (as programmed in the Channel Mode register's Operation field) and by 2 if the address points to a word operand. Note that, if an I/O or memory address is used to point to a word operand, the address must be even to avoid unpredictable results. An address used to point to a byte operand may be even or odd. Since memory byte operand addresses will increment/decrement by 1, they will toggle between even and odd values. Since I/O byte operand addresses will increment/decrement by 2, once programmed to an even or odd value, they will remain even or odd, allowing consecutive I/O operations to access the same half of the data bus. High bus is for even address and low bus for odd.

Current and Base Operation Count Registers

Both the Current and Base Operation Count registers may be loaded during chaining, and may be written to and read from by the host CPU.

The 16-bit Current Operation Count register is used to specify the number of words or bytes to be transferred, searched or transferred-and-searched. For word-to-word operations and byte-word funneling, the Current Operation Count register must be programmed with the number of words to be transferred or searched.

Each time a datum is transferred or searched, the Operation Count register is decremented by 1. Once all of the data is transferred or searched, the transfer or search operation will stop, the Current Operation Count register will contain all zeroes, and the TC bit in Status Register will be "1." If the transfer or search stops before the Current Operation Count register reaches 0, the contents of the register will indicate the number of bytes or words remaining to be transferred or searched. This allows a channel which had been stopped prematurely to be restarted where it left off without requiring reloading of the Current Operation Count register.

For the byte-to-byte operations, the Current Operation Count register should specify the number of bytes to be transferred or searched. The maximum number of bytes which can be specified is 64K bytes by setting the Current Operation Count register to 0000.

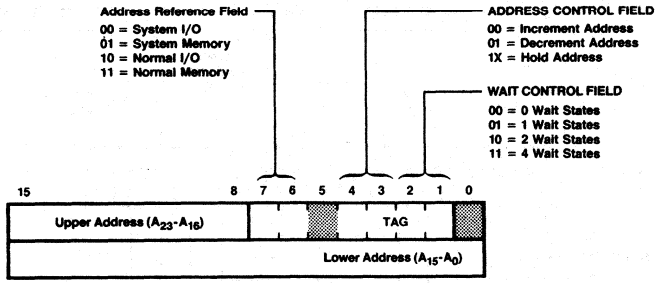


Figure 5. Address Registers A and B

Pattern and Mask Registers

The 16-bit Pattern and Mask registers are used in Search and Transfer-and-Search operations. Both the Pattern and Mask registers may be loaded by chaining, may be written to by the host CPU, and may be read from by the host CPU, provided wait states are inserted, since these registers are slow readable. The Pattern register contains the pattern that the read data is compared to. Setting a Mask register bit to "1" specifies that the bit always matches. See the "Search" and "Transfer-and-Search" sections for further details.

Status Register

The two 16-bit Status registers, depicted in Figure 6, are read-only registers which can be read by the CPU without wait states. Each of these registers reports on the status of its associated channel.

The Interrupt Status Field in the Status register contains the Channel Interrupt Enable (CIE) and Interrupt Pending (IP) bits. These bits are described in detail in the "Interrupt" section of this document.

The UDC status field contains the current channel status. The "channel initialized and waiting for request" status is not explicitly stated – it is reflected by Status register bits ST_{12} through ST_9 being all zero. The "Waiting for Bus" (WFB) status will cause bit ST_{10} to be set and indicates that the channel wants bus control to perform a DMA operation. The channel may or may not actually be asserting BREQ HIGH, depending on the programming of the Master Mode Chip Enable bit (MMO) when the channel decided it wanted the bus. See the "Bus Request/Grant" section for details. If a channel completes a DMA operation and neither Base-to-Current reloading nor auto-chaining were enabled, the No Auto-Reload or Chaining (NAC) bit will be set. The NAC bit will be reset when the channel receives a "Start Chain Command." If two interrupts are queued, the Second Interrupt Pending bit (SIP) will be set and the channel will be inhibited from further activity until an interrupt acknowledge occurs. See the "Interrupt" section for details. Finally, if the channel is issued an $\overline{EOP}$ during chaining, the Chaining Abort (CA) and the NAC will be set. These bits are also set when a "reset" is issued to the UDC. The CA bit holds the NAC bit in the set state. The CA bit is cleared when a new Chain Upper Address and Tag word or Lower Address word is loaded into the channel.

The Hardware Interface Field provides a Hardware Request (HRQ) bit which provides a means of monitoring the channels $\overline{DREQ}$ input pin. When the $\overline{DREQ}$ pin is LOW, the HRQ bit will be "1" and vice-versa. The Hardware Mask (HM) bit, when set, prevents the UDC from responding to a LOW on $\overline{DREQ}$. Note, however, that the Hardware Request bit always reports the true (unmasked) status of $\overline{DREQ}$ regardless of the setting of the HM bit.

The Completion Field stores data at the end of each DMA operation. This data indicates why the DMA operation ended. When the next DMA operation ends, new data is loaded into these bits overwriting, thereby erasing the old setting. Three bits indicate whether the DMA operation ended as a result of a TC, MC or $\overline{EOP}$ termination. The TC bit will be "1" if the Operation Count reaching zero ended the DMA operation. The MC bit will be "1" if an MC termination occurred regardless of whether Stop-on-Match or Stop-on-no-Match was selected. The EOP bit is set only when an external $\overline{EOP}$ ends a DMA transfer; it is not set for $\overline{EOP}$ issued during chaining. Note that two or even all /hree of MC, TC and EOP may be set if multiple reasons exist for ending the DMA operation. The MCH and MCL bits report on the match state of the upper and lower comparator bytes, respectively. These bits are set when the associated comparator byte has a match and are reset otherwise, regardless of whether Stop-on-Match or Stop-on-no-Match is programmed. Regardless of the DMA operation performed, these bits will reflect the comparator status at the end of the DMA operation. These two bits are provided to help determine which byte matched or did not match when using 8-bit matches with word searches and transfer-and-searches. The three reserved bits return zeroes during reads.

Interrupt Vector and Interrupt Save Registers

Each channel has an Interrupt Vector register and an Interrupt Save register. The Interrupt Vector is 8-bit wide and is written to and read from on $AD_0 - AD_7$. The Interrupt Save register may be read by the CPU without wait states. The Interrupt Vector register contains the vector or identifier to be output during an Interrupt Acknowledge cycle. When an interrupt occurs (IP = 1), either because a DMA operation terminated or because $\overline{EOP}$ was driven LOW during chaining, the contents of the Interrupt Vector register and part of the Channel Status register are stored in the 16-bit Interrupt Save register (See Figure 7).

Because the vector and status are stored, a new vector can be loaded into the Interrupt Vector register during chaining, and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. If another interrupt occurs on the channel before the first is acknowledged, further channel activity is suspended.

As soon as the first clear IP command is issued, the status and vector for the second interrupt are loaded into the Interrupt Save register and Channel Operation resumes. The UDC can retain only two interrupts for each channel; a third operation cannot be initiated until the first interrupt has been cleared. See the "Interrupt" section for further details.

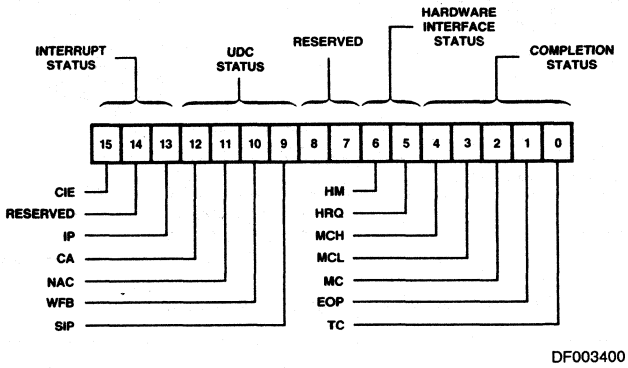


Figure 6. Status Register

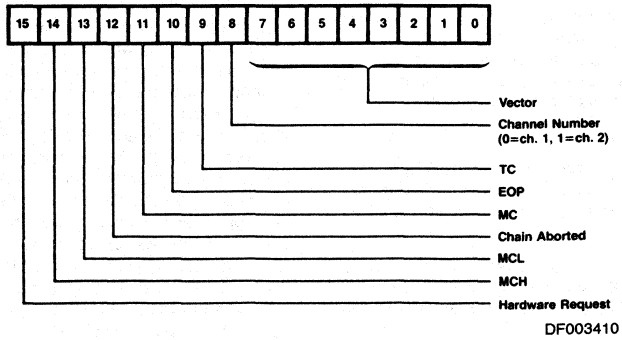


Figure 7. Interrupt Save Register

Channel Mode Register

The Channel Mode registers are two words wide. There are 21 bits defined in each Channel Mode register; the other 11 bits are unused. See Figure 8. The Channel Mode registers may be loaded during chaining and may be read by the host CPU. CPU reads of the Channel Mode register are slow reads and require insertion of multiple wait states. The Channel Mode Low word (bits 0 – 15) may be written to directly by the host CPU. The Channel Mode register selects what type of DMA operation the channel is to perform, how the operation is to be executed, and what action, if any, is to be taken when the channel finishes.

The Data Operation Field and the Transfer Type field select the type of operation the channel is to perform. It also selects the operand size of bytes or words (see Figure 9 for code-definition). The different types of operations are described in detail in the "DMA Operations" section. The Flip bit is used to select whether the Current ARA register points to the source and the Current ARB register points to the destination or vice-versa.

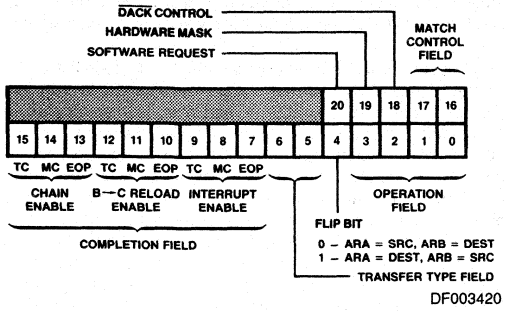


Figure 8. Channel Mode Register

The Completion Field is used to program the action taken by the channel at the end of a DMA operation. This field is discussed in the "Completion Options" section. The 2-bit Match Control field selects whether matches use an 8-bit or 16-bit pattern and whether the channel is to stop-on-match or stop-on-no-match. See Figure 9 and the "Search" section for details. The Software Request bit and Hardware Mask bit can be set and cleared by software command in addition to being loaded in parallel with other Channel Mode bits. These bits are described in detail in the "Initiating DMA Operations" section.

The $\overline{\text{DACK}}$ Control bit is used to specify when the $\overline{\text{DACK}}$ pin is driven active. When this bit is cleared, the channel's $\overline{\text{DACK}}$ pin will be active whenever the channel is performing a DMA Operation, regardless of the type of transaction. Note that the pin will not be active while the channel is chaining. If this bit is set, the $\overline{\text{DACK}}$ pin will be inactive during chaining, during both Flowthru Transfers and Flowthru Transfer-and-Searches, and during Searches, but $\overline{\text{DACK}}$ will be pulsed active during Flyby Transfers and Flyby Transfers-and-Searches at the time necessary to strobe data into or out of the Flyby peripheral. Flyby operations are discussed in detail in the "Flyby Transactions" section.

DATA OPERATION FIELD			
	Operand Size		
Code/Operation	ARA	ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

TRANSFER TYPE FIELD AND MATCH CONTROL FIELD		
Transfer Type	Code	Match Control
Single Transfer	00	Stop on No Match
Demand Dedicated/Bus Hold	01	Stop on No Match
Demand Dedicated/Bus Release	10	Stop on Word Match
Demand Interleave	11	Stop on Byte Match

Figure 9. Channel Mode Coding

Chain Address Register

Each channel has a Chain Address register which points to the chain control table in memory containing data to be loaded into the channel's registers. The Chain Address register, as shown in Figure 10, is two words long. The first word consists of an Upper Address and Tag field. The second word contains the 16-bit Lower Address portion of the memory address. The Tag field contains 2 bits used to designate the number of wait states to be inserted during accesses to the Chain Control Table.

The Chain Address register may be loaded during chaining and may be read from and written to by the host CPU without wait states. If an EOP is issued to the UDC during chaining, the Chain Address register holds the old address. This is true even if the access failure occurred while new Chain Address data was being loaded, since the old data is restored unless both words of the new data are successfully read. Note, however, that EOPs that occur when chaining and while loading a new Chain Address cause the new data to be lost.

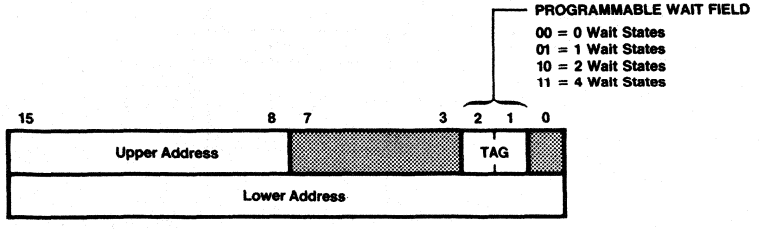


Figure 10. Chain Address Register

DETAILED DESCRIPTION

Any given DMA operation, be it a Transfer, a Search, or a Transfer-and-Search operation, consists of three phases. In the first phase, the channel's registers are initialized to specify and control the desired DMA operation. In the second phase, the DMA operation itself is started and performed. The final phase involves terminating the DMA operation and performing any actions selected to occur on termination. Each of these different phases is described in detail in the following sections.

Reset

The UDC can be reset either by hardware or software. The software reset command is described in the "Commands" section. Hardware resets are applied by pulling $\overline{RESET}$ LOW. The UDC may be in control of the bus when a reset is applied. BACK is removed internally causing the outputs to go tri-state. If BACK remains HIGH after reset, the UDC will not drive the bus unless BREQ is active. As soon as BACK goes inactive, the UDC places the $AD_0 - AD_{15}$, $A_{16} - AD_{23}$, R/$\overline{W}$, $\overline{DS}$, N/$\overline{S}$, M/$\overline{IO}$ B/$\overline{W}$, $\overline{TBEN}$ and $\overline{RBEN}$ signals in the high-impedance state.

Both software and hardware resets clear the Master Mode register, clear the CIE, IP and SIP bits, and set the CA and NAC bits in each Channel's Status register. The contents of all other UDC registers will be unchanged for a software reset. Since a hardware reset may have been applied partway through a DMA operation being performed by a UDC channel, the channel's registers should be assumed to contain indeterminate data following a hardware reset.

The Master Mode register contains all zeroes after a reset. The UDC is disabled, and the CPU interleave and hardware wait are inhibited.

Because the CA and NAC bits in the Status register are set by a reset, the channel will be prevented from starting a DMA operation until its Chain Address register's Segment, Tag and Offset fields are programmed and the channel is issued a "Start Chain" Command.

Channel Initialization

The philosophy behind the Am9516A UDC design is that the UDC should be able to operate with a minimum of interaction with the host CPU. This goal is achieved by having the UDC load its own control parameters from memory into each channel. The CPU has to program only the Master Mode register and each Channel's Chain Address register. All other registers are loaded by the channels themselves from a table located in the System memory space and pointed to by the Chain Address register. This reloading operation is called chaining, and the table is called the Chain Control Table.

The Upper and Lower Address fields of the Chain Address register form a 24-bit address which points to a location in system memory space. Chaining is performed by repetitively reading words from memory. Note that the Chain Address register should always be loaded with an even Address; loading an odd Address will cause unpredictable results. The 2-bit Tag field facilitates interfacing to slow memory by allowing the user to select 0, 1, 2 or 4 programmable wait states. The UDC will automatically insert the programmed number of wait states in each memory access during chaining.

The Chain Address register points to the first word in the Chain Control Table. This word is called the Reload Word. See Figure 11. The purpose of the Reload Word is to specify which registers in the channel are to be reloaded. Reload Word bits 10 – 15 are undefined and may be 0 or 1. Each of bits 0 through 9 in the Reload Word correspond to either one or two registers in the channel (see Figure 12). When a Reload Word bit is "1," it means that the register or registers corresponding to that bit are to be reloaded. If a Reload Word bit is "0," the register or registers corresponding to that bit are not to be reloaded. The data to be loaded into the selected register(s) follow the Reload Word in memory (i.e., the data are stored at successively larger memory addresses). The Chain Control Table is a variable length table. Only the data to be loaded are in the table, and the data are packed together.

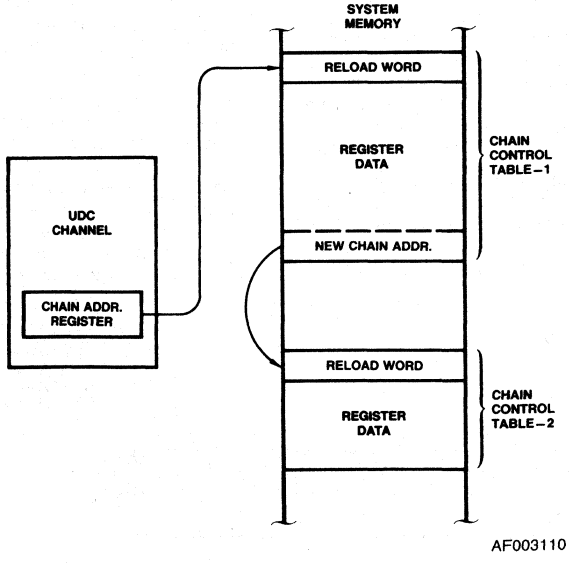


Figure 11. Chaining and Chain Control Tables

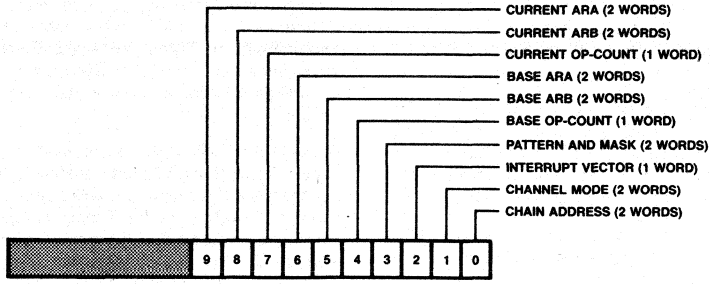


Figure 12. Reload Word/Chain Control Register

When the channel is to reload itself, it first uses the Chain Address register contents to load the Reload Word into the UDC's Chain Control register. Next, the Chain Address register contents are incremented by two to point to the next word in memory. The channel than scans the Reload Word register from bit 9 down to bit 0 to see which registers are to be reloaded. If no registers are specified (bits 9 – 0 are all 0), no registers will be reloaded. If at least one of bits 9 – 0 is set to "1," the register(s) corresponding to the set bit are reloaded, the bit is cleared and the Chain Address register is incremented by 2. The channel continues this operation of scanning the bits from the most significant to least significant bit position, clearing each set bit after reloading its associated registers and incrementing the Chain Address register by 2. If all of bits 9 to 0 are set, all the registers will be reloaded in the order: Current ARA, Current ARB, Current Operation Count,. . . Channel Mode and Chain Address. Figure 13 shows two examples of Chain Control Tables. Example 1 shows the ordering of data when all registers are to be reloaded. In example 2 only some registers are reloaded. Once the channel is reloaded, it is ready to perform a DMA operation. Note when loading address registers, the Upper Address and Tag word are loaded first, then the Lower Address word. Also, the Pattern register is loaded before the Mask register.

Initiating DMA Operations

DMA operations can be initiated in one of three ways – by software request, by hardware request and by loading a set software request bit into the Channel Mode register during Chaining.

Starting After Chaining

If the software request bit of the Channel Mode register is loaded with a "1" during chaining, the channel will perform the programmed DMA operation at the end of chaining. If the channel is programmed for Single Operation or Demand, it will perform the operation immediately. The channel will give up the bus after chaining and before the operation if the CPU interleave bit in the Master Mode register is set. See the "Channel Response" section for details. Note that once a channel starts a chaining operation by fetching a Reload Word, it retains bus control at least until chaining of the last register's data is performed.

Software Requests

The CPU can issue Software Request commands to start DMA operations on a channel. This will cause the channel to request the bus and perform transfers. See the description of the software request command for details.

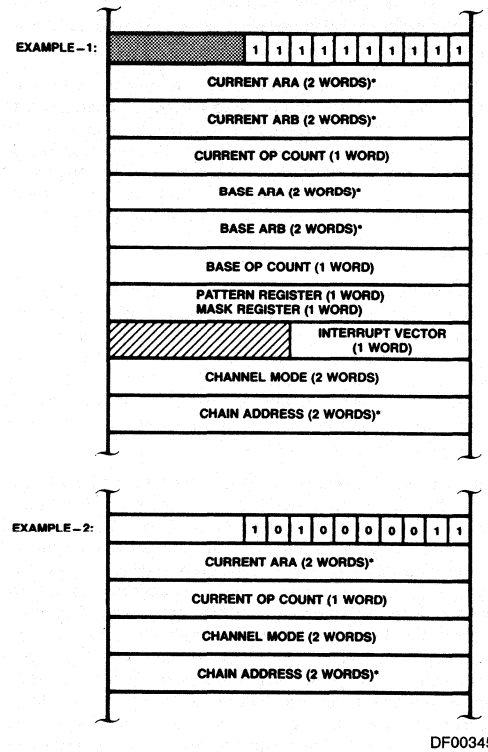


Figure 13. Examples of Chain Control Table

*Load the Upper Address and Tag Word first, then the Lower Address Word.

Hardware Requests

DMA operations will often be started by applying a LOW on the channel's $\overline{\text{DREQ}}$ input. The "Channel Response" section describes when the LOW $\overline{\text{DREQ}}$ signals are sampled and when the $\overline{\text{DREQ}}$ requests can be applied to start the next DMA operation after chaining (see Timing Diagrams 1 and 2).

Bus Request/Grant

Before the UDC can perform a DMA Operation, it must gain control of the system bus. The BREQ and BACK interface pins

provide connections between the UDC and the host CPU and other devices, if present, to arbitrate which device has control of the system bus. When the UDC wants to gain bus control, it drives BREQ HIGH.

Some period of time after the UDC drives BREQ HIGH, the CPU will relinquish bus control and drive its HLDA signal HIGH. When the UDC's BACK input goes HIGH, it may begin performing operations on the system bus. When the UDC finishes its operation, it stops driving BREQ HIGH.

When more than one device is used, a priority encoder and a priority decoder are used to decide the bus grant priority.

DMA Operations

There are three types of DMA operations: Transfer, Search, and Transfer-and-Search. Transfers move data from a source location to a destination location. Two types of transfers are provided: Flowthru and Flyby. Searches read data from a source and compare the read data to the contents of the Pattern register. A Mask register allows the user to declare "don't care" bits.

The user can program that the search is to stop either when the read data matches the masked pattern or when the read data fails to match the masked pattern. This capability is called Stop-on-Match and Stop-on-no-Match. Transfer-and-Search combines the two functions to facilitate the transferring of variable length data blocks. Like Transfer, Transfer-and-Search can be performed in either Flowthru or Flyby mode.

Transfers

Transfers use four of the Channel registers to control the transfer operation: the Current ARA and ARB registers; the Current Operation Count register; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select whether ARA is to point to the source and ARB is to point to the destination or vice-versa. The Current Operation Count register specifies the number of words or bytes to be transferred.

Bits $CM_3 - M_0$ in the Channel Mode register program whether a Flowthru or Flyby transfer is to be performed. Flowthru transfers are performed in either two or three steps. First, the channel outputs the address of the source and reads the source data into the UDC's Temporary register. In two-step Flowthru Transfer, the channel will then address the destination and write the Temporary register data to the destination location. The three-step Flowthru operation (i.e., the byte-word funneling) is described later in this section. The source and destination for Flowthru Transfers can both be memory locations or both peripheral devices, or one may be a memory location and the other a peripheral device. The $\overline{DACK}$ output for the transferring channel may be programmed to be inactive throughout the transfer or active during the transfer. This is controlled by bit CM_{18} in the Channel Mode register.

Flyby transfers provide improved transfer throughput over Flowthru but are restricted to transfers between memory and peripherals or between two peripherals. Flyby operations are described in detail in the "Flyby Transactions" section.

Transfers can use both byte- and word-sized data. Flowthru byte-to-byte transfers are performed by reading a byte from the source and writing a byte to the destination. The Current Operation Count register must be loaded with the number of bytes to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±1 if the register points to a memory space ($TG_6 = 2$) and by ±2 if the register points to an I/O space ($TG_6 = 0$).

Flowthru word-to-word transfers require that the Current Operation Count specify the number of words to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±2 regardless of whether the register points to memory or I/O space.

Byte-word funneling provides packing and unpacking of byte data to facilitate high speed transfers between byte and word peripherals and/or memory. This funneling option can only be used in Flowthru mode. Funneled Flowthru transfers are performed in three steps. For transfers from a byte source to a word destination, two consecutive byte reads are performed from the source address. The data read is assembled into the UDC's Temporary register. In the third step, the Temporary register data is written to the destination address in a word transfer. Funneled transfers from a word source to a byte destination are performed by first loading a word from the source into the UDC's Temporary register. The word is then written out to the destination in two byte writes. For funnel operations, the byte-oriented address must be in the Current ARA register, and the word-oriented address must be in the Current ARB register. The Flip bit (CM_4) in the Channel Mode register is used to specify which address is the source and which is the destination. When the byte address is to be incremented or decremented, the increment/decrement operation occurs after each of the two reads or writes. The Current Operation Count Register must be loaded with the number of words to be transferred.

In byte-to-word funneling operations, it is necessary to specify which half of the Temporary register (upper or lower byte) is loaded with the first byte of data. Similarly, for word-to-byte funneling operations, it is necessary to define which half of the Temporary register is written out first. Figure 14 summarizes these characteristics for both byte-to-word and word-to-byte funneling operations. The criterion used to determine the packing/unpacking order is based on whether the Current ARB register is programmed for incrementing or decrementing of the address. Note that if the address is to remain unchanged (i.e., if bit TG_4 in the Tag Field of the Current ARB register is 1), the increment/decrement bit (bit TG_3) still specifies the packing order.

Search

Searches use five of the Channel registers to control the operation: either the Current ARA or ARB, the Operation Count, the Pattern and Mask registers, and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select either Current ARA or ARB as the register specifying the source for the search. Only one of the Current Address registers is used for search operations since there is no destination address required. The Current Operation Count register specifies the maximum number of words or bytes to be searched.

Search operations involve repetitive reads from the peripheral or memory until the specified match condition is met. The search then stops. This is called a Match Condition or MC termination. Each time a read is performed, the Source address, if so programmed, is incremented or decremented and the Operation Count is decremented by 1. If the match condition has not been met by the time the Operation Count reaches zero, the zero value will force a TC termination, ending the search. Searches can also stop due to a LOW being applied to the $\overline{EOP}$ interface pin. During a Search operation, the channel's $\overline{DACK}$ output will be either inactive or active throughout the search. This is controlled by bit CM_{18} in the Channel Mode register. The reads from the peripheral or memory performed during Search follow the timing sequences described in the "Flowthru Transactions" sections.

On each read during a Search operation, the UDC's Temporary register is loaded with data and compared to the Pattern register. The user can select that the Search is to stop when the Pattern and Temporary register contents match or when they don't match. This Stop-On-Match/Stop-On-No-Match feature is programmed in bit CM_{17} of the Channel Mode register. CM_2 is an enable for the output of the comparator and allows the MC signal to be generated. A Mask register allows the user to exclude or mask selected Temporary register bits from the comparison by setting the corresponding Mask register bit to "1." The masked bits are defined to always match. Thus, in Stop-On-Match, successful matching of the unmasked bits, in conjunction with the always-matched masked bits, will cause the search to stop. For Stop-On-No-Match, the always-matched masked bits are by definition excluded from not matching and therefore excluded from stopping the search.

For word reads the user may select either 8-bit or 16-bit compares through the Channel Mode register bit CM_{16}. In an 8-bit, Stop-On-Match, word-read operation, successful matching of either the upper or lower byte of unmasked Pattern and Temporary registers bits will stop the search. Both bytes do not have to match. In 16-bit Stop-On-Match with word reads, all unmasked Pattern and Temporary register bits must match to stop the search. In an 8-bit or 16-bit, Stop-On-No-Match, word-read Search operation, failure of any bit to match will terminate the Search operation.

In an 8-bit Stop-On-Match with byte-reads, the Search will stop if either the upper or lower byte of unmasked Pattern and Temporary register bits match. For an 8-bit Stop-On-No-Match with byte reads, failure of matching in any unmasked Pattern and Temporary register bit will cause the Search to stop. For 8-bit searches, the upper and lower bytes of the Pattern and Mask register should usually be programmed with the same data. Failure to set the upper and lower bytes of the Pattern and Mask registers to identical values will result in different comparison criteria being used for the upper and lower bytes of the Temporary register. Users failing to program identical values for the upper and lower bytes can predict the results by recognizing that in 8-bit Stop-On-Match, the search will end if all the unmasked bits in either the upper or lower bytes match, and for 8-bit Stop-On-No-Match, the failure of any unmasked bit to match will end the Search. For accurate predictions, it is also necessary to know that for word reads the Temporary register high and low bytes are loaded from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. In byte reads, the read byte is duplicated in both halves of the Temporary register except in funneling.

Funneling Direction	Current ARB Tag Field		Increment/Decrement and Packing/Unpacking Rules
	TG_4	TG_3	
Word-to-Byte ($CM_4 = 1$)	0	0	Increment ARB, Write High Byte First
	0	1	Decrement ARB, Write Low Byte First
	1	0	Hold ARB, Write High Byte First
	1	1	Hold ARB, Write Low Byte First
Byte-to-Word ($CM_4 = 0$)	0	0	Increment ARB, Read High Half of Word First
	0	1	Decrement ARB, Read Low Half of Word First
	1	0	Hold ARB, Read High Half of Word First
	1	1	Hold ARB, Read Low Half of Word First

Figure 14. Byte/Word Funneling

Transfer-and-Search

Transfer-and-Search combines the operations of the Transfer and the Search functions. The registers used to control Transfer-and-Searches are the Current ARA and ARB registers, the Operation Count register, the Pattern and Mask registers, and the Channel Mode register.

A Transfer-and-Search operation will end when the data transferred meets the match condition specified in Channel Mode register bits $CM_{17} - CM_{16}$. The Mask and Pattern registers indicate those bits being compared with the Temporary register contents. Like Transfers and Searches, Transfers-and-Searches will also be terminated if the operation count goes to zero or if a LOW is applied to the $\overline{EOP}$ pin. Regardless of whether Transfer-and-Search stops because of a TC, MC or EOP, it will always complete the iteration by writing to the destination address before ending (writing twice for word-to-byte funneling).

In Flowthru mode, the Transfer-and-Search timing is identical to Flowthru Transfer. While the data is in the Temporary register, it is masked by the Mask register and compared to the Pattern register. For word Transfer and Transfer-and-Search, the high and low bytes of the Temporary register are always written to and read from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. For byte Transfer and Transfer-and-Search, the byte read is always loaded into both halves of the Temporary register, and the entire register is driven directly out onto the $AD_{15} - AD_0$ bus. Transfer-and-Search can also be used with byte word funneling. In funneling, the match is an 8-bit match or 16-bit match as determined by the setting of bit CM_{16}.

Flyby Transfer-and-Search can be used to increase throughput for transfer between two peripherals or between memory and a peripheral. Memory-to-Memory Flyby is not supported. Also, in Flyby, the operand sizes of the source and destination must be the same, funneling is not supported. A complete discussion of Flyby timing is given the "Flyby Transactions" section. During a Flyby Transfer-and-Search, data is loaded into the Temporary register to facilitate the comparison operation, and at the same time, data is transferred from the source to the destination. When byte operands are used, data is loaded into both bytes of the Temporary register, from the $AD_{15} - AD_8$ bus if the Current ARA register is even and from $AD_7 - AD_0$ line if the Current ARA register is odd. This will alternate for memory bytes so the user must drive both halves of the bus to use the search. When word operands are used, data is loaded directly from $AD_{15} - AD_8$ and $AD_7 - AD_0$ into the Temporary register's high and low bytes respectively.

Channel Response

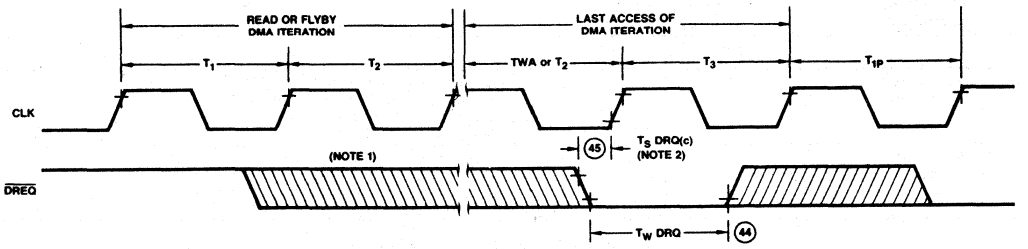
Channel Mode register bits $CM_6 - CM_5$ select the channel's response to the request to start a DMA operation. The response falls into either of two types: Single Operation or Demand. There are three subtypes for Demand operations: Demand Dedicated with Bus Hold, Demand Dedicated with Bus Release, and Demand Interleave. To make discussions clear, it is necessary to define the term "single iteration of a DMA operation." For Search operations, one iteration consists

of a single read operation and a comparison of the read data to the unmasked Pattern register bits. The Operation count will be decremented by 1, and the Current Address register used incremented or decremented if so programmed. For Transfer and Transfer-and-Search operations, a single iteration comprises reading a datum from the source, writing it to the destination, comparing the read datum to the unmasked Pattern register bits (Transfer-and-Search only), decrementing the Operation count by 1 and incrementing/decrementing the Current ARA and ARB registers if so programmed. In byte-word funneling, a single iteration consists of two reads followed by a write (Byte-to-Word funneling) or one read followed by two writes (Word-to-Byte funneling). In all Transfer and Transfer-and-Search cases, the iteration will not stop until the data in the Temporary register is written to the destination. See Appendix B for flowchart.

Single Operation

The Single Operation response is intended for use with peripherals which transfer single bytes or words at irregular intervals. Each application of a Software request command will cause the channel to perform a single iteration of the DMA operation. Similarly, if the Software request bit is set by chaining, at the end of chaining the channel will perform a single iteration of the DMA operation. Each application of a HIGH-to-LOW transition on the $\overline{DREQ}$ input will also cause a single iteration of the DMA operation. If the Hardware mask bit is set when the transition is made, the iteration will be performed when the mask is cleared, providing the DMA operation has not terminated. See the Set/Clear Hardware mask bit command for details. Each time a Single Operation ends, the channel will give up control of the bus unless a new transition has occurred on $\overline{DREQ}$. The new transition can occur anytime after the HIGH-to-LOW ALE transition of a read or Flyby memory or I/O access of the DMA iteration. Timing Diagram 1 shows the times after which a new transition can be applied and recognized to avoid giving up the us at the end of the current iteration.

*TIMING DIAGRAM 1. Sampling $\overline{DREQ}$ During Single Transfer DMA Operations



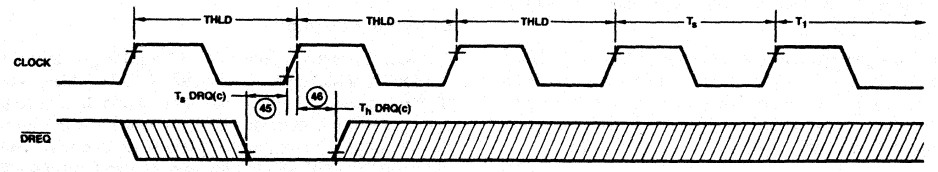
WF007460

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur TsDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.
3. Flyby and Search transactions have only a single access; parameter TsDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

*See Appendix D for timing parameters.

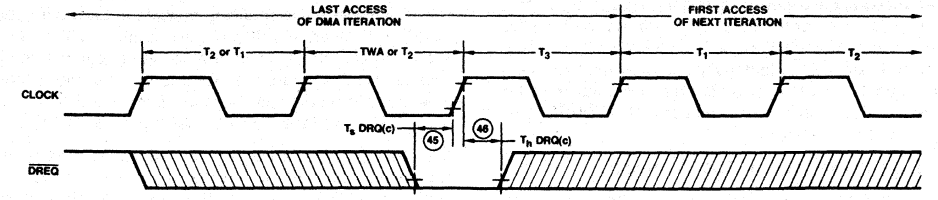
TIMING DIAGRAM 2. $\overline{DREQ}$ Sampling in Demand Mode

(a) Sampling of $\overline{DREQ}$ while in Bus Hold Mode



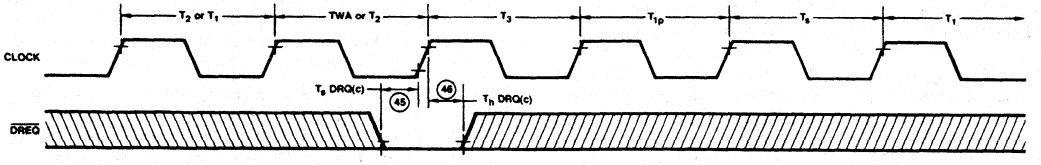
WF007480

(b) $\overline{DREQ}$ Sampling in Demand Mode During DMA Operations



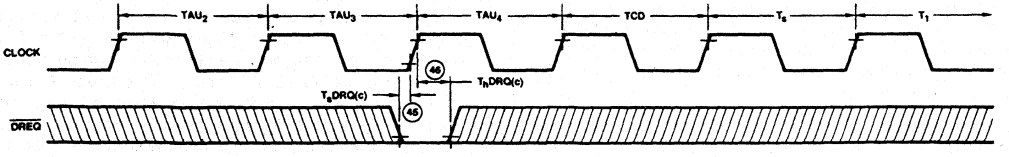
WF007490

(c) Sampling $\overline{DREQ}$ at the End of Chaining



WF007500

(d) Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading



WF007510

Notes: 1. $\overline{DREQ}$ must be LOW from the start of TsDRQ(c) to the end of ThDREQ(c) to ensure that the request is recognized.
2. Failure to meet this setup time will result in the channel releasing the bus.
3. Ts is a setup state, generated before entering DMA operation cycle.
4. TAU_2, TAU_3 and TAU_4 are auto-reload states, followed by TCD (chain decision) state.

Demand Dedicated With Bus Hold

In Demand Dedicated with Bus Hold (abbreviated Bus Hold), the application of a Software request command or the setting of the software request bit during chaining or applying a LOW level on the $\overline{DREQ}$ input will cause the channel to acquire bus control.

If $\overline{DACK}$ is programmed as a level output ($CM_{18} = 0$), $\overline{DACK}$ will be active from when the channel acquires bus control to when it relinquishes control. A Software Request will cause the channel to request the bus and perform the DMA operations until TC, MC or EOP.

Once the channel gains bus control due to a LOW $\overline{DREQ}$ level, it samples $\overline{DREQ}$ as shown in Timing Diagram 2. If $\overline{DREQ}$ is LOW, an iteration of the DMA operation is performed. If $\overline{DREQ}$ is HIGH, the channel retains bus control and continues to drive all bus control signals active or inactive, but performs no DMA operation. Thus the user can start or stop execution of DMA operations by modulating $\overline{DREQ}$. Once TC, MC or EOP occurs, the channel will either release the bus or, if chaining or Base-to-Current reloading is to occur, perform the desired operation. After chaining or Base-to-Current reloading, if the channel is still in Bus Hold mode and does not have a set software request bit (set either by chaining or command), the channel will relinquish bus control unless a LOW $\overline{DREQ}$ level occurs within the time limits.

Demand Dedicated With Bus Release

In Demand Dedicated with Bus Release (abbreviated Bus Release), the application of a Software Request will cause the channel to request the bus and perform the programmed DMA operation until TC, MC or EOP. If the channel was programmed for Bus Release and the software request bit was set during chaining, the channel will start the DMA operation as soon as chaining ends, without releasing the bus and will continue performing the operation until TC, MC or EOP.

When an active LOW $\overline{DREQ}$ is applied to a channel programmed for Bus Release, the channel will acquire the bus and perform DMA operations: (a) until TC, MC or EOP or (b) until $\overline{DREQ}$ goes inactive. Timing Diagram 2 shows when $\overline{DREQ}$ is sampled to determine if the channel should perform another cycle or release the bus. Note that this sampling also occurs on the last cycle of a chaining operation. If a channel has an active $\overline{DREQ}$ at the end of chaining, it will begin performing DMA operations immediately, without releasing the bus. When a TC, MC or EOP occurs, terminating a Bus Release mode operation, the channel, if enabled for chaining and/or Base-to-Current reloading, will perform reloading and/or chaining (assuming the Status register's SIP bit is clear) without releasing the bus.

If the SIP bit in the Channel Status register is set when a DMA termination occurs, the channel will relinquish the bus control until an Interrupt Acknowledge has been received and the SIP bit is cleared. After an interrupt has been serviced, the channel will perform the Base-to-Current reloading and/or chaining if enabled for the termination.

If an active request is not applied and the channel is in Demand Dedicated with Bus Hold, the channel will go into state THLD (see Timing Diagram 2(a)). If an active request is not applied and the channel is in Demand Dedicated with Bus Release or Demand Interleave mode, it will release the bus. Note that even if an active request is applied in Demand Interleave, the channel may still release the bus. The request for Demand Interleave should continue to be applied to ensure that the channel eventually responds to the request by acquiring the bus (i.e., the request is not latched by the channel).

Demand Interleave

Demand Interleave behaves in different ways depending on the setting of Master Mode register bit MM_2. If MM_2 is set, the UDC will always relinquish bus control and then re-request it after each DMA iteration. This permits the CPU and other devices to gain bus control. If MM_2 is clear, control can pass from one UDC channel to the other without requiring the UDC to release bus control. If both channels have active requests, control will pass to the channel which did not just have control. For instance if MM_2 is clear and both channels have active requests and are in Demand Interleave mode, control will toggle between the channels after each DMA operation iteration and the UDC will retain bus control until both channels are finished with the bus. If MM_2 is set and both channels have active requests and are in Demand Interleave mode, each channel will relinquish control to the CPU after each iteration resulting in the following control sequence: channel 1, CPU, channel 2, CPU, etc. Note that if there are other devices on the bus, they may gain control during the part of the sequence labelled CPU. See Appendix B for flowchart.

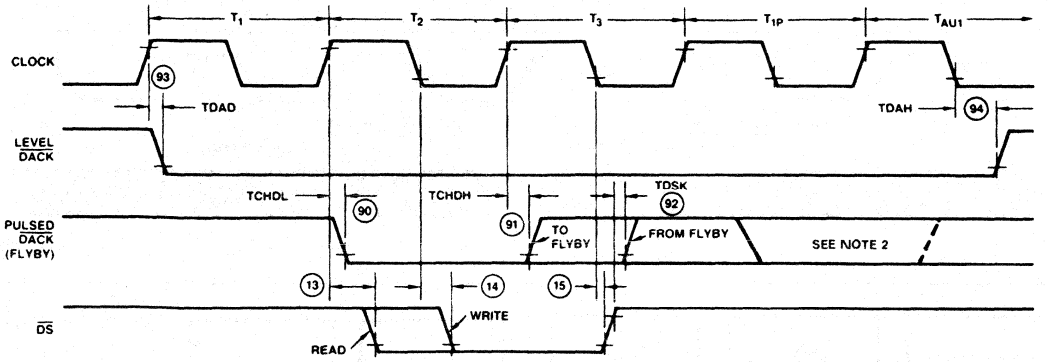
A software or hardware request will cause a channel programmed for Demand Interleave to perform interleaved DMA operations until TC, MC or EOP. If the Software request bit is set during chaining, the channel will retain the bus after chaining and will immediately start performing a DMA iteration and will interleave all DMA iterations after the first. If $\overline{DREQ}$ is LOW on the last cycle during chaining, the channel will perform a single iteration immediately after chaining and interleave thereafter until: (a) TC, MC or EOP or (b) $\overline{DREQ}$ goes HIGH. If (b) occurs, the channel will relinquish the bus until $\overline{DREQ}$ goes LOW again and the channel again starts performing interleaved operations. If (a) occurs, the channel will not interleave before first performing chaining and/or Base-to-Current reloading (assuming SIP is cleared).

The waveform of $\overline{DACK}$ is programmed in Channel Mode Register (CM_{18}). The Pulsed $\overline{DACK}$ is for flyby transaction only. See Timing Diagram 3. Note: This figure shows a single Search or Flyby iteration. State TWA is optionally inserted if programmed. For more than one iteration, the level $\overline{DACK}$ output would stay active during the time the channel had bus control. When CM_{18} is set, the $\overline{DACK}$ output will be inactive for all nonflyby modes.

Wait States

The number of wait states to be added to the memory or I/O transfer can be programmed by the user as 0, 1, 2 or 4 and can be separately programmed for the Current Address registers A and B and for the Chain Address register. This allows different speed memories and peripherals to be associated with each of these addresses. The Base Address registers A and B also have a Tag Field which is loaded into the Current ARA and ARB registers during Base-to-Current reloading. Because many users utilizing the software programmable wait states will not need the ability to generate hardware wait states through the $\overline{WAIT}$ pin, the wait function can be disabled by clearing the Wait Line Enable bit (MM_2) in the Master Mode register.

TIMING DIAGRAM 3. $\overline{DACK}$ Timing



WF007521

Notes: 1. Level $\overline{DACK}$ RE occurs as shown if auto-reloading is not programmed; otherwise, it stays LOW for three additional clocks.

2. This extra $\overline{DACK}$ pulse occurs only at EOP. It should be used to distinguish which channel got the EOP.

During DMA transactions, the $\overline{WAIT}$ input is sampled in the middle of the T_2 state. If $\overline{WAIT}$ is HIGH, and if no programmable wait states are selected, the UDC will proceed to state T_3. Otherwise, at least one wait state will be inserted. The $\overline{WAIT}$ line is then sampled in the middle of state TWA. If $\overline{WAIT}$ is HIGH, the UDC will proceed to state T_3. Otherwise additional wait states will be inserted. (See Timing Diagram 4.)

Consider what happens in a transaction when both hardware and software wait states are inserted. Each time the $\overline{WAIT}$ line is sampled, if it is LOW, a hardware wait state will be inserted in the next cycle. The software wait state insertion will be suspended until $\overline{WAIT}$ is sampled and is HIGH. The hardware wait states may be inserted anytime during the software wait state sequence. It is important to know that hardware wait states are served consecutively rather than concurrently with software wait states. For example, assume for a Flowthru I/O Transaction that a user has programmed 4 software wait states. Driving a LOW on the $\overline{WAIT}$ input during T_2 for 2 cycles would insert 2 hardware wait states. Driving $\overline{WAIT}$ HIGH for 3 cycles would allow insertion of three of the four software wait states. Driving $\overline{WAIT}$ LOW for 2 more cycles would insert 2 more hardware wait states. Finally, driving $\overline{WAIT}$ HIGH would allow the final software wait state to be inserted. During this last software wait state, the $\overline{WAIT}$ pin would be sampled for the last time. If it is HIGH, the channel will proceed to state T_3. If the pin is LOW, the channel will insert hardware wait states until the pin goes HIGH and the channel would then enter state T_3 to complete the I/O transaction.

DMA Transactions

There are three types of transactions performed by the Am9516A UDC: Flowthru, Flyby and Search. Figures 15 and 16 show the configurations of Flowthru and Flyby Transactions.

Flowthru Transactions

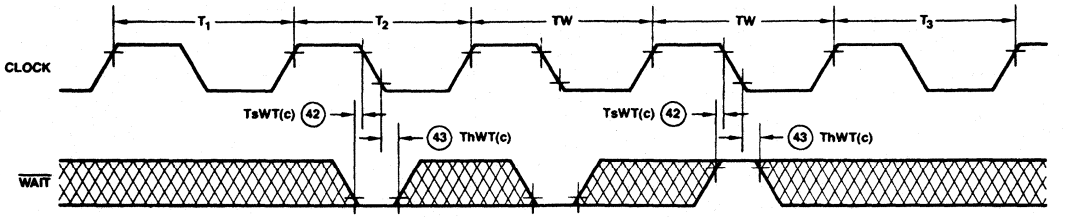
A Flowthru Transaction consists of Read and Write cycles. Each cycle consists of three states: T_1, T_2, and T_3 as shown in Timing Diagram 5. The user may select to insert software wait states through the Tag fields of the Current ARA and ARB registers. In addition, if Master Mode register bit $MM_2 = 1$, hardware wait states may be inserted by driving a LOW signal on the $\overline{WAIT}$ pin.

The M/$\overline{IO}$ and N/$\overline{S}$ lines will reflect the appropriate level for the current cycle early in T_1. The TG_6 and TG_7 bits of the current ARA and ARB registers should be programmed properly. The ALE output will be pulsed HIGH to mark the beginning of the cycle. The offset portion of the address for the peripheral being accessed will appear on $AD_0 - AD_{15}$ during T_1. The R/$\overline{W}$ and B/$\overline{W}$ lines will select a read or write operation for bytes or words. The R/$\overline{W}$, N/$\overline{S}$, M/$\overline{IO}$ and B/$\overline{W}$ lines will become stable during T_1 and will remain stable until after T_3.

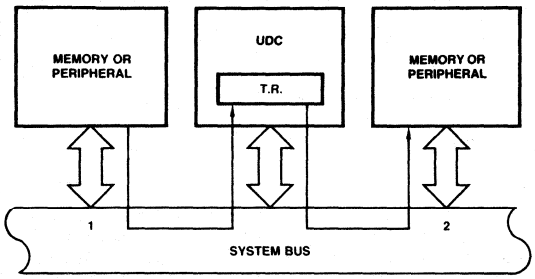
I/O address space is byte-addressed, but both 8- and 16-bit data sizes are supported. During I/O transactions the B/$\overline{W}$ output signal will be HIGH for byte transactions and LOW for word transactions. For I/O transactions, both even and odd addresses can be output; hence, the address bit output on AD_0 may be 0 or 1.

The channel can perform both I/O read and I/O write operations; the M/$\overline{IO}$ line will be LOW. During an I/O read, the $AD_0 - AD_{15}$ bus will be placed in the high-impedance state by the UDC during T_2. The UDC will drive the $\overline{DS}$ output LOW to signal the peripheral that data can be gated onto the bus. The UDC will strobe the data into its Temporary register during T_3. $\overline{DS}$ will be driven HIGH to signal the end of the I/O transaction. During I/O write, the UDC will drive the contents of the Temporary register onto the $AD_0 - AD_{15}$ bus and shortly after will drive the $\overline{DS}$ output LOW until T_3. Peripherals may strobe the data on AD bus into their internal registers on either the falling or rising edge. If the peripheral is to be accessed in a Flyby transaction also, data should be written on the rising edge of $\overline{DS}$ only.

TIMING DIAGRAM 4. $\overline{WAIT}$ Timing

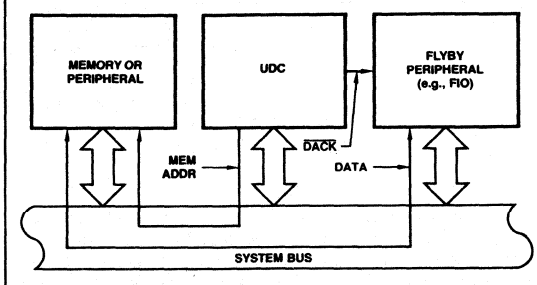


WF007680



AF003120

Figure 15. Configuration of Flowthru Transaction



AF003130

Figure 16. Configuration of Flyby Transaction

For byte I/O writes, the channel will drive the same data on data bus lines $AD_0 - AD_7$ and $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 0, the UDC will strobe data in from data lines $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 1, the UDC will strobe data in from data lines $AD_0 - AD_7$. Thus, when an 8-bit peripheral is connected to the bus, its internal registers will typically be mapped at all even or all odd addresses. To simplify accesses to 8-bit peripherals, byte oriented I/O addresses are incremented/decremented by 2.

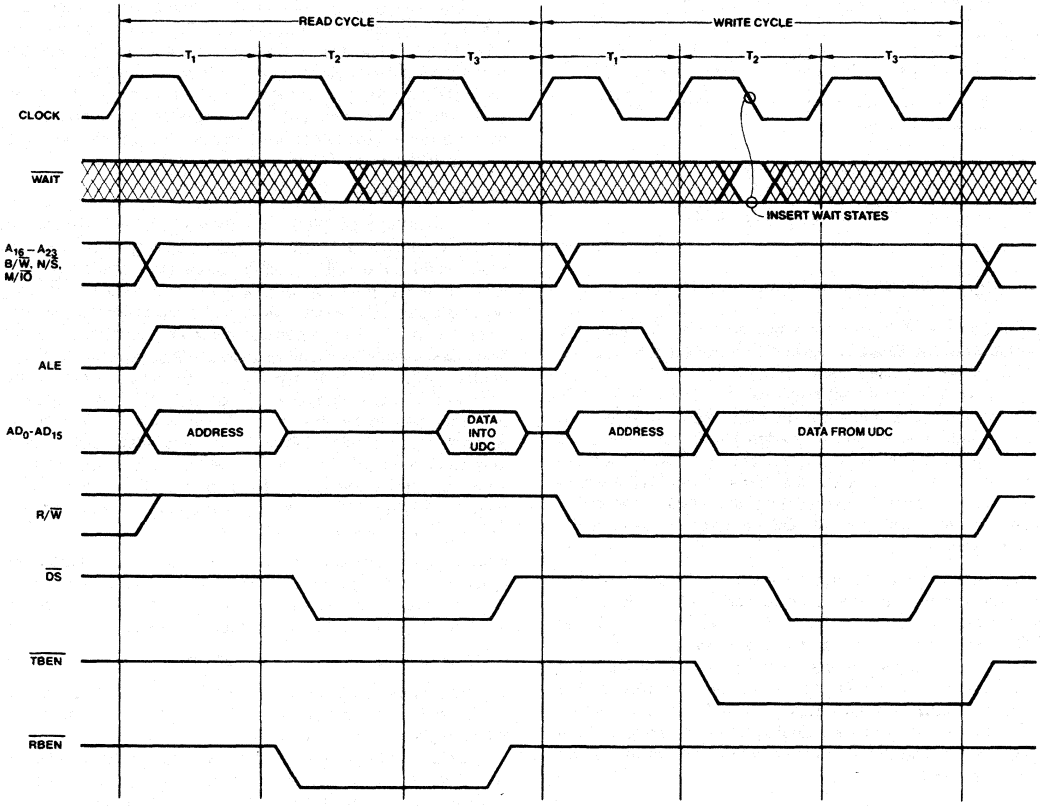
The channel can perform the I/O read and memory write operation, the memory read and I/O write operation, and the memory read and memory write operation, also. The timing for all Flowthru transactions is the same.

During chaining operations the UDC reads words from an address in System memory pointed to by the active channel's Chain Address register. Those chaining operations are performed identically to the Flowthru memory read transactions, except that the data is loaded into an internal UDC channel register rather than the Temporary register. Note that chaining never causes a write or a byte read; thus, all memory writes or all byte accesses are due to DMA operations. A typical memory operation consists of three states: T_1, T_2, and T_3, as shown in Timing Diagram 5. The user may select to insert 1, 2 or 4 software wait states after state T_2 and before state T_3 by programming the Tag field of the Current Address register or the Chain Address register. If the Wait Line Enable bit in the Master Mode register is set, the user may also insert hardware wait states after state T_2 and before state T_3 by driving a LOW on the $\overline{WAIT}$ line. The operation of Flowthru memory transactions is performed identically to the Flowthru I/O transactions. (See Timing Diagram 5.)

Flyby Transactions

Flyby Transfer and Flyby Transfer-and-Search operations are performed in a single cycle, providing a transfer rate significantly faster than that available from Flowthrus. In Flyby, operations can only be performed between memory and peripheral or between peripheral and peripheral. Memory-to-Memory operations cannot be performed in Flyby mode; these must be done using Flowthru.

TIMING DIAGRAM 5. Flowthru Transactions



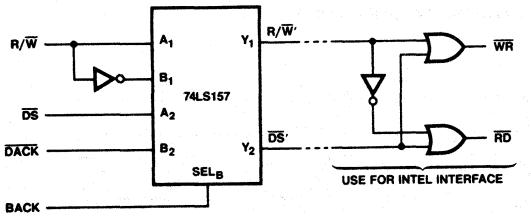
WF007540

The Flyby Transaction can only be used with peripherals having a special Flyby signal input or with external logic. This Flyby input is connected to the channel's $\overline{DACK}$ output. For memory-peripheral Flyby, the address of the source memory location must be programmed in the Current ARA register. The Current ARB register must be programmed with the destination memory location for peripheral-memory Flyby. For Flyby peripheral-to-peripheral transaction, if both peripherals have a Flyby input, only one (called "flyby peripheral") should be connected to $\overline{DACK}$; the other peripheral's Flyby input should be held high during the Flyby operation. The address of the peripheral (called "non-flyby peripheral") not connected to the channel's $\overline{DACK}$ output should be programmed in the Current ARB register when it is a destination. When the non-flyby peripheral is a source, its address should be programmed in the current ARA register. Note that a set Flip bit ($CM_4 = 1$) is for Flyby peripheral to Non-Flyby peripheral or Memory Write transaction (defined as "From Flyby Transaction"), and a clear Flip bit ($CM_4 = 0$) is for the memory or non-flyby peripheral read to Flyby peripheral transaction (defined as "To Flyby Transaction").

Transaction	CM_4	R/$\overline{W}$	Address of Memory or Non-Flyby Peripheral
To Flyby	0	HIGH	ARA
From Flyby	1	LOW	ARB

A Flyby operation is performed using three states: T_1, T_2, and T_3. During T_1 the channel pulses ALE and outputs the address information. See Timing Diagram 6. The R/$\overline{W}$ line is HIGH for "To Flyby" Transaction, and the R/$\overline{W}$ line is LOW for "From Flyby" Transaction.

The channel's M/$\overline{IO}$ and N/$\overline{S}$ lines are coded as specified by the Current ARA or ARB Tag field. The B/$\overline{W}$ line indicates the operand size programmed in the Channel Mode register Operation field. During state T_1 the channel drives R/$\overline{W}$ line to indicate the transaction direction. During state T_2 the channel drives both $\overline{DS}$ and $\overline{DACK}$ active. The Flyby Peripheral connected to $\overline{DACK}$ inverts the R/$\overline{W}$ signal to determine whether it is being read from or written to (see Figure 17).



$$R/\overline{W}' = \overline{BACK \cdot R/\overline{W} + \overline{BACK} \cdot R/\overline{W}}$$
$$\overline{DS}' = \overline{BACK \cdot DACK + \overline{BACK} \cdot DS}$$
$$\overline{RD} = \overline{DACK \cdot R/\overline{W} \cdot BACK + DS \cdot R/\overline{W} \cdot \overline{BACK}}$$
$$\overline{WR} = \overline{DACK \cdot R/W \cdot BACK + \overline{DS} \cdot R/W \cdot \overline{BACK}}$$

AF003140

Figure 17. Flyby Peripheral Interface

The pulsed $\overline{DACK}$ input serves two purposes: to select the peripheral for the Read/Write, and to provide timing information on when to drive data onto or input data from the $AD_0 - AD_{15}$ bus. Note that because the "Flyby Peripheral" never gets explicitly addressed by $AD_0 - AD_{15}$, it must know which internal register is to be loaded from or driven onto the $AD_0 - AD_{15}$ bus. On state T_3, the $\overline{DS}$ and $\overline{DACK}$ lines are driven inactive to conclude the transfer. In Transfer-and-Search mode, data is loaded into the UDC's Temporary register on the LOW-to-HIGH $\overline{DS}$ transition to perform the Search function.

To provide adequate data setup time, the rising edge of $\overline{DS}$ or $\overline{DACK}$ should be the edge used to perform the write to the transfer destination. To extend the active time of $\overline{DS}$ and $\overline{DACK}$, wait states can be inserted between T_2 and T_3. Software wait states can be inserted by programming the appropriate code in the Tag field of the Current ARA or ARB registers. Hardware wait states can be inserted by pulling $\overline{WAIT}$ LOW if the Wait Line Enable bit in the Master Mode register is set. The $\overline{WAIT}$ line is sampled in the middle of the T_2 or TWA state.

Termination

There are three ways a Transfer-and-Search or Search operation can end and two ways a Transfer operation can end. When a channel's Current Operation count goes to 0, the DMA operation being performed will end. This is called a TC or Terminal Count termination. A DMA operation can also be stopped by driving the $\overline{EOP}$ pin LOW with external logic. This is called an EOP termination. Search and Transfer-and-Search operations have a third method of terminating called Match Condition or MC termination. An MC termination occurs when the data being Transferred-and-Searched or Searched meets the match condition programmed in Channel Mode register bits $CM_{17} - CM_{16}$. These bits allow the user to stop when a match occurs between the unmasked Pattern register bits and the data read from the source, or when a no-match occurs. Both byte and word matches are supported. MC terminations do not apply to Transfer operations since the pattern matching logic is disabled in Transfer mode.

End-of-Process

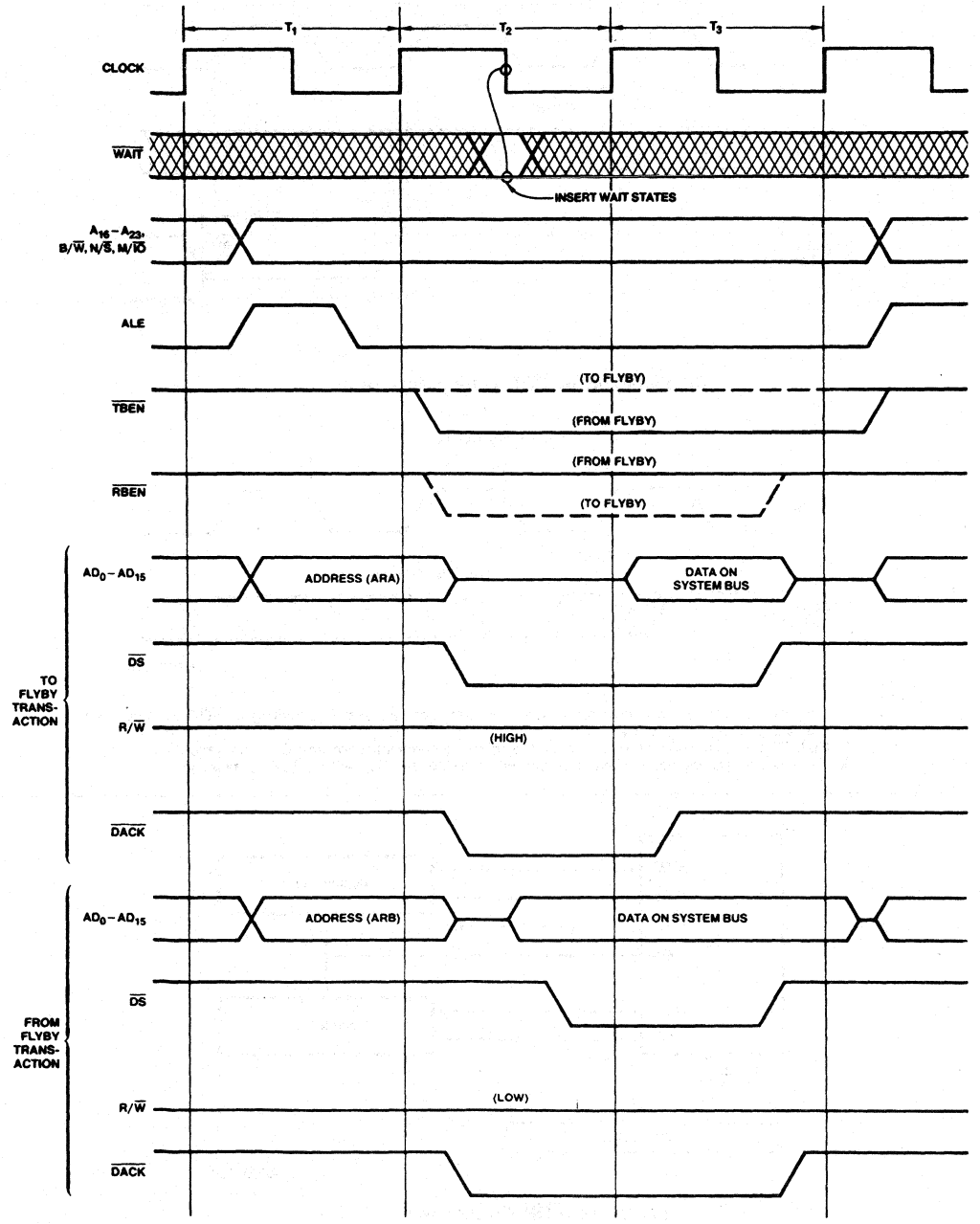
The End-of-Process (EOP) interface pin is a bi-directional signal. Whenever a TC, MC or EOP termination occurs, the UDC will drive the $\overline{EOP}$ pin LOW. During DMA operations, the $\overline{EOP}$ pin is sampled by the UDC to determine if the $\overline{EOP}$ is being driven LOW by external logic. Timing Diagram 7 shows when internal EOPs are generated marking termination of all Transfers. These figures also show the point during the DMA iteration when the EOP pin is sampled. The generation of internal EOPs and sampling of external EOPs for Transfer-and-Searches follow the same timing used for Transfers. Since there is a single $\overline{EOP}$ pin for both channels, $\overline{EOP}$ should only be driven LOW by a channel while that channel is being serviced. This can be accomplished by selecting a level $\overline{DACK}$ output ($CMR_{18} = 0$) and gating each channel's $\overline{EOP}$ request with $\overline{DACK}$, as shown in Figure 18.

If an EOP is detected while the channel is trying to reload the Chain Address register, the new Chain Address Offset and Segment are discarded and the old address +2 is preserved to allow inspection of the erroneous address.

Programming Completion Options

When a channel ends a DMA operation, the reason for ending is stored in the Completion Status Field of the channel's Status register. See Figure 6. This information is retained until the next DMA operation ends at which time the Status register is updated to reflect the reason(s) for the latest termination. Note that it is conceivable that more than one bit in the Completion Field could be set. An as extreme example, if a channel decremented its Current Operation count to zero, causing a TC termination; input data from the source generated a match causing an MC termination; and there was a LOW on the $\overline{EOP}$ pin resulting in an EOP termination, all three of the channel's Status register completion bits would be set.

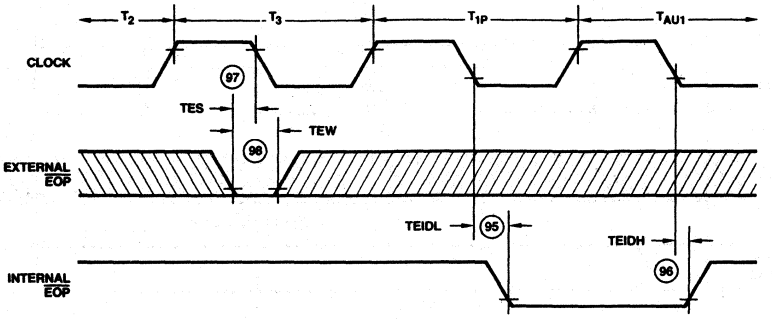
TIMING DIAGRAM 6. Flyby Transactions



WF007550

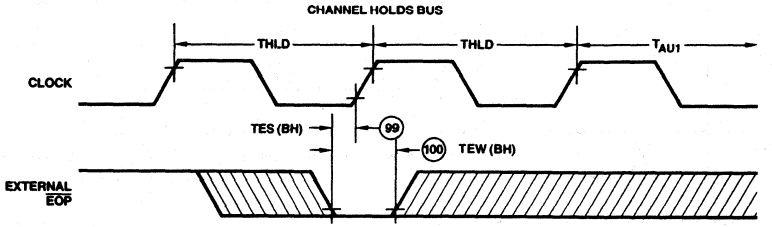
TIMING DIAGRAM 7. $\overline{EOP}$ Timing

(a) $\overline{EOP}$ Sampling and Generation During DMA Operations



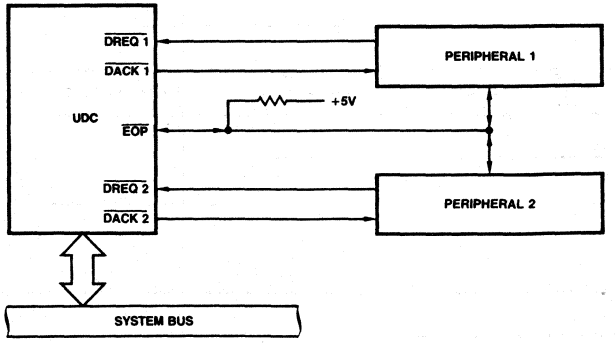
WF007560

(b) Sampling of $\overline{EOP}$ During Bus Hold



WF007570

Notes: 1. The diagram lists state names for both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_{1P} is a pseudo-T_1 state, generated following termination of any DMA operation.
3. T_{AU1} is an auto-intialization state, generated following the TC, MC or $\overline{EOP}$ termination.



AF003150

Figure 18. $\overline{EOP}$ Connection

When a DMA operation ends, the channel can:

(a) Issue an Interrupt request (i.e., setting the IP or SIP bit of the channel's Status register);

(b) Perform Base-to-Current reloading;

(c) Chain reload the next DMA operation;

(d) Perform any combination of the above; or

(e) None of the above.

The user selects the action to be performed by the channel in the Completion option field of the Channel Mode register. For each type of termination (TC, MC or $\overline{EOP}$), the user can choose which action or actions are to be taken. If no reloading is selected for the type of termination that occurred, the NAC bit in the Status register will be set.

More than one action can occur when a DMA operation ends. This may arise because more than one action was programmed for the applicable termination. The priorities of those actions are Interrupt request first, Base-to-Current reloading second, and then chaining. The Interrupt cannot be serviced unless the UDC has relinquished the bus.

Interrupts

To allow the UDC to start executing a new DMA operation after issuing an Interrupt, but before an Interrupt acknowledge is received, a two-deep Interrupt queue is implemented on each channel. The following discussion will describe the standard Interrupt structure and then elaborate on the additional Interrupt queuing capability of the UDC.

A complete Interrupt cycle consists of an Interrupt request followed by an Interrupt-acknowledge transaction. The request, which consists of $\overline{INT}$ being pulled LOW, notifies the CPU that an Interrupt is pending. The Interrupt-acknowledge transaction, which is initiated by the CPU as a result of the request, performs two functions: it selects the peripheral whose Interrupt is to be acknowledged, and it obtains a vector that identifies the selected device and operation – the cause of the Interrupt.

A peripheral can have one or more sources of Interrupt. Each Interrupt source has two bits that control how it generates Interrupts. These bits are a Channel Interrupt Enable bit (CIE) and an Interrupt Pending bit (IP). On the UDC, each channel is an Interrupt source. The two Interrupt control bits are located in bits CM_{15} and CM_{13} of each channel's Status register.

Each channel has its own vector register for identifying the source of the Interrupt during an Interrupt acknowledge transaction. There is one bit (MM_3) in the Master Mode register used for controlling Interrupt behavior for the whole device.

Once a channel issues an Interrupt, it is desirable to allow the channel to proceed with the next DMA operation before the Interrupt is acknowledged. This could lead to problems if the UDC channel attempted to chain reload the Vector register contents. In such a situation, it may not be clear whether the old or new vector would be returned during the acknowledge. This dilemma is resolved in the UDC by providing each channel with an Interrupt Save register. When the channel sets IP as part of the procedure followed to issue an Interrupt, the contents of the Vector register and some of the Status register bits are saved in an Interrupt Save register. See Figure 8. When an Interrupt Acknowledge cycle is performed, the contents of the Interrupt Save register are driven onto the bus. Although the use of an Interrupt Save register allows the channel to proceed with a new task, problems can still potentially arise if a second Interrupt is to be issued by the channel before the first Interrupt is acknowledged. To avoid conflicts between the first and second Interrupt, each channel has a Second Interrupt Pending (SIP) bit in its Status register. When a second Interrupt is to be issued before the first Interrupt is acknowledged, the SIP bit is set and the channel relinquishes the bus until an acknowledge occurs. For compatibility with polled Interrupt schemes, the Interrupt save register can be read by the host CPU without wait states. As an aid to debugging a system's Interrupt logic, whenever IP is set, the Interrupt Save register is loaded from the Vector and Status registers.

Note that the SIP bit is transferred to the IP bit when IP is cleared by the host CPU. Whenever CIE is set, $\overline{INT}$ will go LOW as soon as IP is set.

Base-to-Current Reloading

When a channel finishes a DMA operation, the user may select to perform a Base-to-Current reload. (Base-to-Current reloading is also referred to as Auto-reloading in this document.) In this type of reload, the Current Address registers A and B are loaded with the data in the Base Address registers A and B respectively, and the Current Operation Count register is loaded with the data in the Base Operation Count. The Base-to-Current reload operation facilitates repetitive DMA operations without the multiple memory accesses required by chaining. Although the channel must have bus control to perform Base-to-Current reloading, the complete reloading operation occurs in four clock cycles (i.e., TAU_1 through TAU_4). Note that if the channel had to relinquish the bus because two unacknowledged interrupts were queued, it will have to regain bus control to perform any Base-to-Current reloading (or chaining, for that matter). In this case it acquires the system bus once an Interrupt acknowledge is received, even if it immediately afterward will relinquish the bus because no hardware/software request is present.

Chaining

If the channel is programmed to chain at the end of a DMA operation, it will use the Chain Address register to point to a Chain Control Table in memory. The first word in the table is a Reload word, specifying the register(s) to be loaded. Following the Reload word are the data values to be transferred into the register(s). Chaining is described in detail in the "Channel Initialization" section.

Because chaining occurs after Base-to-Current reloading, it is possible to reset the Current Address registers A and B and the Current Operation Count register to the values used for previous DMA operations, then chain reload one or two of these registers to some special value to be used, perhaps, for this DMA operation only. If the Base values are not reloaded during chaining, the channel can revert back to the Base values at a later cycle.

If an all zero Reload word is fetched during chaining, the chain operation will not reload any registers, but in all other respects, it will perform like any other chaining operation. Thus, the Chain Address will be incremented by 2 to point to the next word in memory, and at the end of the all Zero-Reload word chain operation, the channel will be ready to perform a DMA operation. All Zero-Reload words are useful as "Stubs" to start or terminate linked lists of DMA operations traversed by chaining. On the other hand, care must be taken in their use since the channel may perform an erroneous operation if it is unintentionally started after the chaining operation.

COMMAND DESCRIPTIONS

Figure 19 shows a list of UDC commands. The commands are executed immediately after being written by the host CPU into the UDC's Command register (Figure 20). A description of each command follows.

Reset (00)

This command causes the UDC to be set to the same state generated by a Hardware Reset. The Master Mode register is set to all zeros; the CIE, IP and SIP bits are cleared; the NAC and CA bits in each channel's Status register are set; and the channel activity is forbidden. The Chain Address must be programmed since its state may be indeterminate after a Reset. The lockout preventing channel activity is cleared by issuing a Start Chain command.

Start Chain Channel 1/Channel 2 (A_0/A_1)

This command causes the selected channel to clear the No Auto-Reload or Chain (NAC) bit in the channel's Status register and to start a chain reload operation of the channel's registers, as described in the "Channel Initialization" section. These effects will take place even if the fetched Reload word is all zeros. This command will only be honored if the Chain Abort (CA) bit and the Second Interrupt Pending (SIP) bit in the channel's Status register are clear. If either the CA or SIP bit is set, this command is disregarded.

When the Waiting For Bus (WFB) bit of Status register is set, if the "Start Chain" command is issued, the channel will honor the command after one DMA iteration. It is nearly impossible for the CPU to issue a command when WFB = 1 and the UDC is enabled.

Command	Opcode Bits 7654	Opcode Bits 3210	Example Code HEX
Reset	000X	XXXX	00
Start Chain Channel 1	101X	XXX0	A0
Start Chain Channel 2	101X	XXX1	A1
Set Software Request Channel 1	010X	XX10	42
Set Software Request Channel 2	010X	XX11	43
Clear Software Request Channel 1	010X	XX00	40
Clear Software Request Channel 2	010X	XX01	41
Set Hardware Mask Channel 1	100X	XX10	82
Set Hardware Mask Channel 2	100X	XX11	83
Clear Hardware Mask Channel 1	100X	XX00	80
Clear Hardware Mask Channel 2	100X	XX01	81
Set CIE, or, IP Channel 1	001E	XP10	32*
Set CIE, or, IP Channel 2	001E	XP11	33*
Clear CIE, or, IP Channel 1	001E	XP00	30*
Clear CIE, or, IP Channel 2	001E	XP01	31*
Set Flip Bit Channel 1	011X	XX10	62
Set Flip Bit Channel 2	011X	XX11	63
Clear Flip Bit Channel 1	011X	XX00	60
Clear Flip Bit Channel 2	011X	XX01	61

*Notes: 1. E = Set to 1 to perform set/clear on CIE; Clear to 0 for no effect on CIE.
2. P = Set to 1 to perform set/clear on IP; Clear to 0 for no effect on IP.
3. X = "don't care" bit. This bit is not decoded and may be 0 or 1.

Figure 19. UDC Command Summary

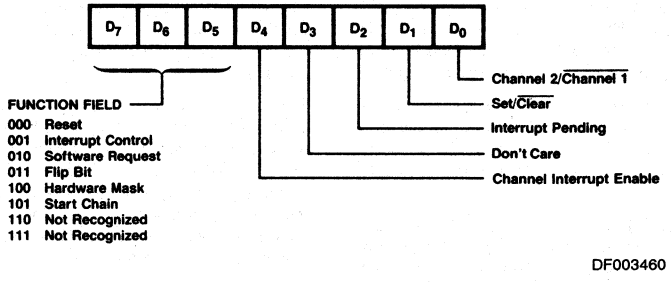


Figure 20. Command Register

Software Request Channel 1/Channel 2 (Set: 42/43, Clear: 40/41)

This command sets or clears the software request bit in the selected channel's Mode register. If the Second Interrupt Pending (SIP) bit and No Auto-Reload or Chain (NAC) bit in the channel's Status register are both cleared, the channel will start executing the programmed DMA operation. If either the SIP or NAC bit is set, the channel will not start executing a DMA operation until both bits are cleared. The SIP bit will clear when the channel receives an Interrupt acknowledge. One way to clear the NAC bit is to issue a Start Chain command to the channel. If the fetched Reload Word is all zeros, the channel's registers will remain unchanged and the software request bit, if set earlier by command, will cause the programmed DMA operation to start immediately. If during chaining new information is loaded into the Channel Mode register, this new information will, of course, overwrite the software request bit.

Set/Clear Hardware Mask 1/Mask 2 (Set: 82/83; Clear: 80/81)

This command sets or clears the Hardware Mask bit in the selected channel's Mode register. This command always takes effect. The Hardware Mask bit inhibits recognition of an active signal on the channel's $\overline{DREQ}$ input; this bit does not affect recognition of a software request. If the channel is in single transfer mode, it performs DMA operations upon receipt of a transition on $\overline{DREQ}$ rather than in response to a $\overline{DREQ}$ level. Transitions occurring while the Hardware Mask bit is set will be stored and serviced when the Hardware Mask is cleared, assuming the Channel has not chained. The UDC will request the system bus 1 1/2 to 2 clocks after the receipt of any $\overline{DREQ}$, after which a minimum of one DMA iteration is unavoidable. $\overline{DREQ}$ transitions are only stored for the current DMA operation. If the channel performs a chain operation of single transfer mode, any $\overline{DREQ}$ transition stored for later service is cleared.

Timing Diagrams 1 and 2 show the minimum times when a new $\overline{DREQ}$ can be applied if it is to be serviced by the new DMA operation. Note in Diagram 1 the notation of First iteration and Last iteration. This means, for example, $\overline{DREQ}$ may be asserted during the write cycle T_1 of a Flowthru transaction, but may never be asserted during T_1 of a Flyby transaction because Flyby is done in one iteration.

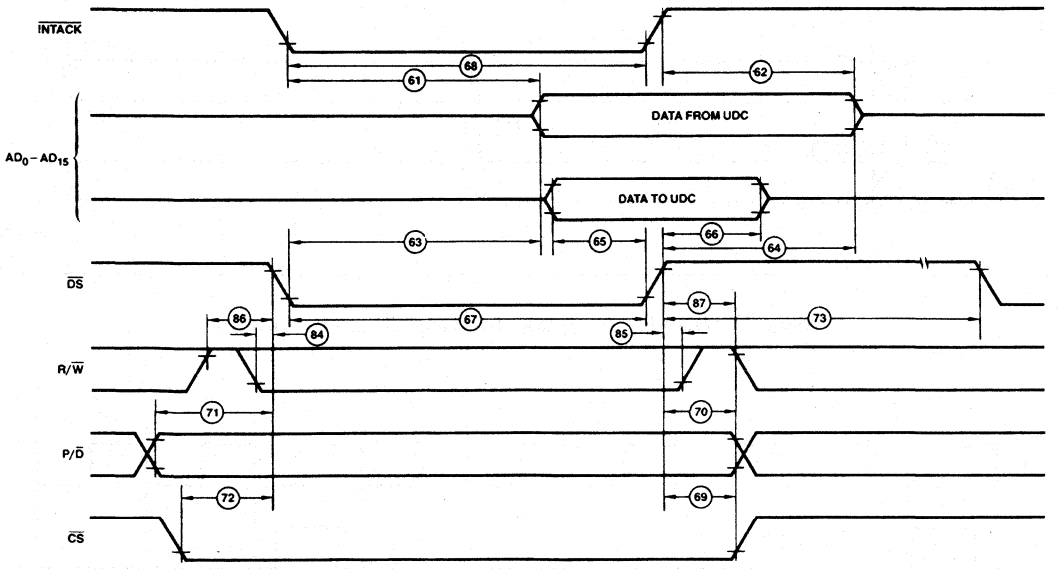
Set/Clear CIE, and IP Channel 1/Channel 2 (see Figure 19)

This command allows the user to either set or clear any combination of the CIE and IP bits in the selected channel's Status register. These bits control the operation of the channel's Interrupt structure and are described in detail in the "Interrupts" section. Setting the IP bit causes the Interrupt Save register to be loaded with the current Vector and Status. The IP bit is cleared to facilitate an efficient conclusion to the processing of an interrupt.

Set/Clear Flip Bit Channel 1/Channel 2 (Set: 62/63; Clear: 60/61)

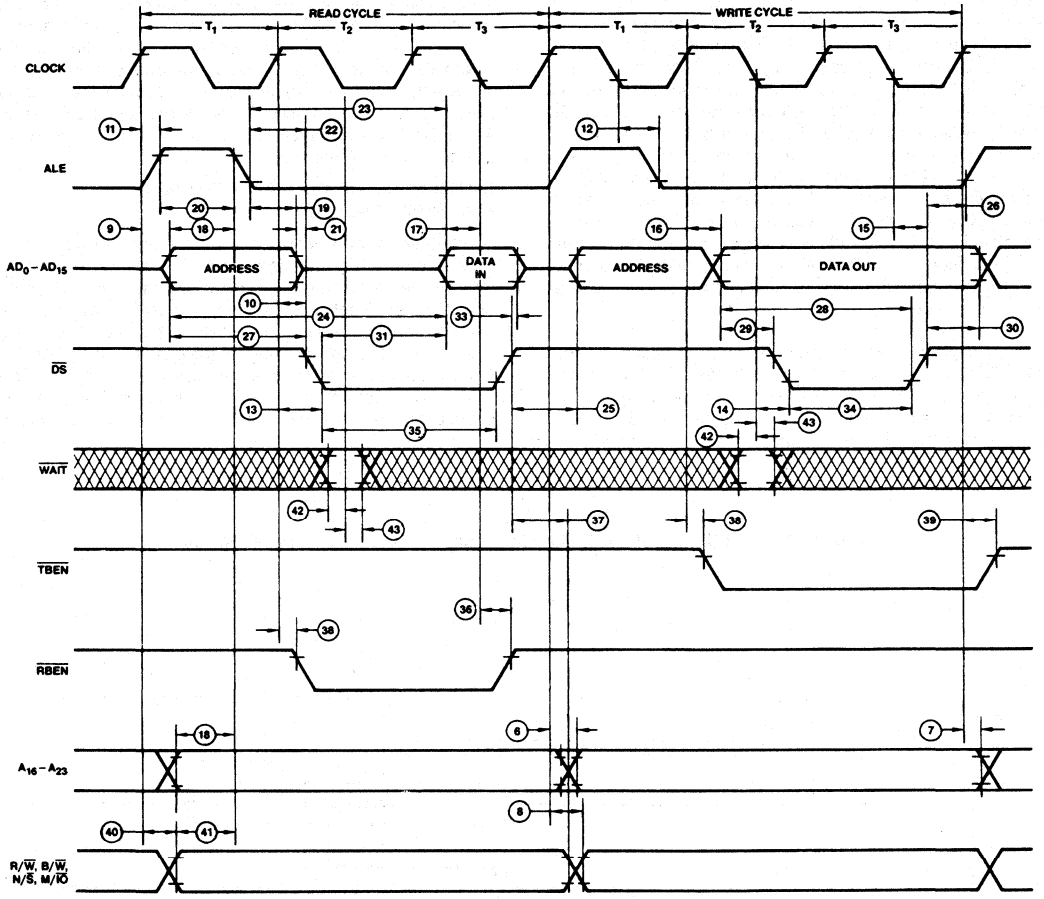
The Flip Bit in the selected channel's Mode register can be cleared and set by this command. This allows the user to reverse the source and destination and thereby reverse the data transfer direction without reprogramming the channel. This command will be most useful when repetitive DMA operations are being performed by the channel, using Base-to-Current reloading for channel reinitialization and using this command to control the direction of transfer. Chaining new information into the Channel Mode register will, of course, overwrite the Flip bit.

TIMING DIAGRAM 8. AC Timing when UDC is a Bus Slave



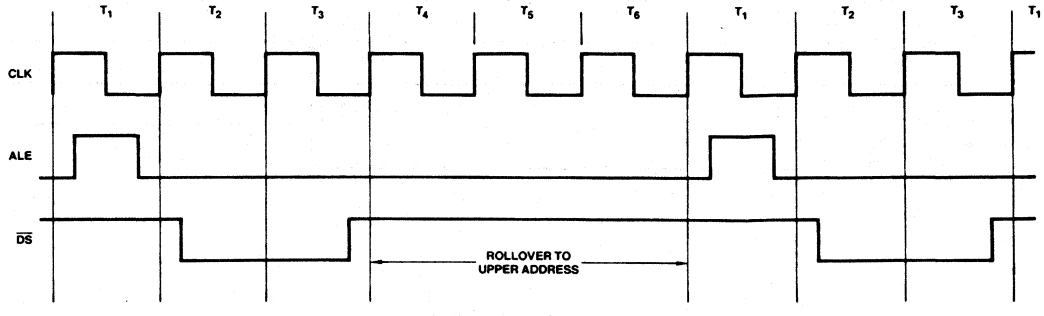
WF007580

TIMING DIAGRAM 9. AC Timing when UDC is a Bus Master



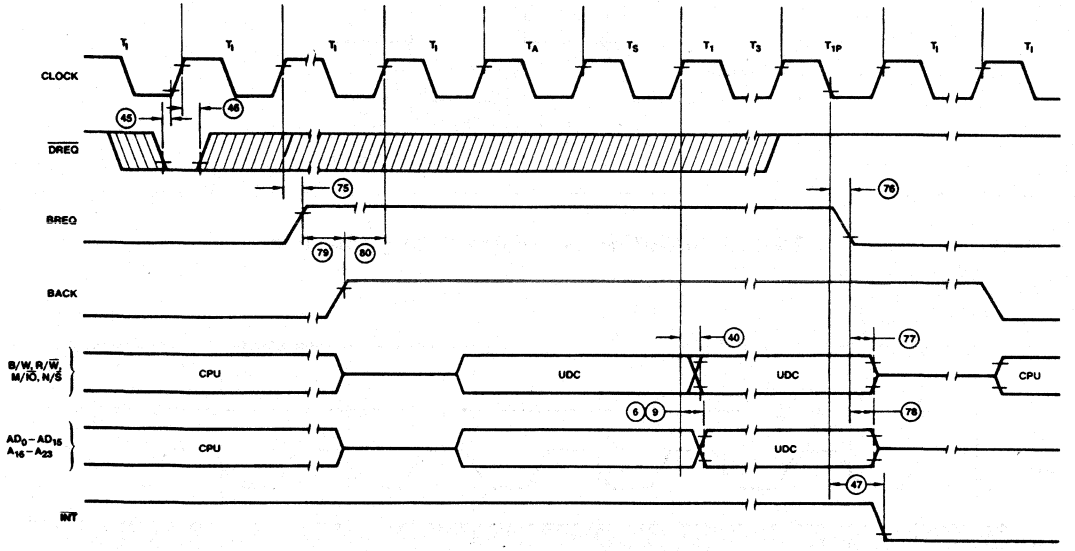
WF007590

TIMING DIAGRAM 10. Upper Address Rollover Timing



WF007600

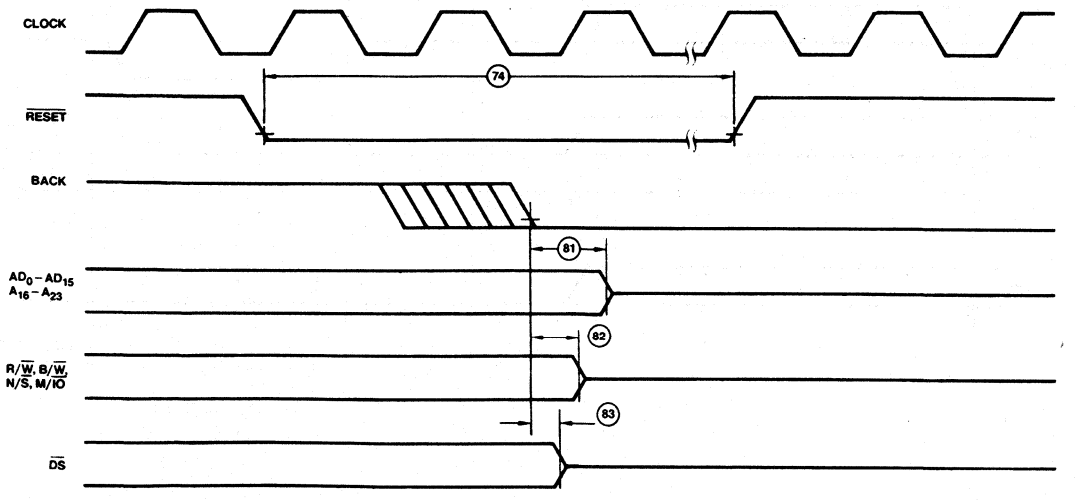
TIMING DIAGRAM 11. Bus Exchange Timing



WF007610

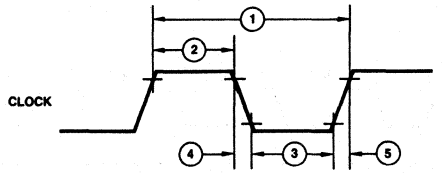
Notes: 1. Under no circumstance can BACK be removed prior to BREQ.
2. One extra ALE occurs each time the 9516 releases the bus. No $\overline{DS}$ accompanies it, so this should not present a problem.

TIMING DIAGRAM 12. Reset Timing



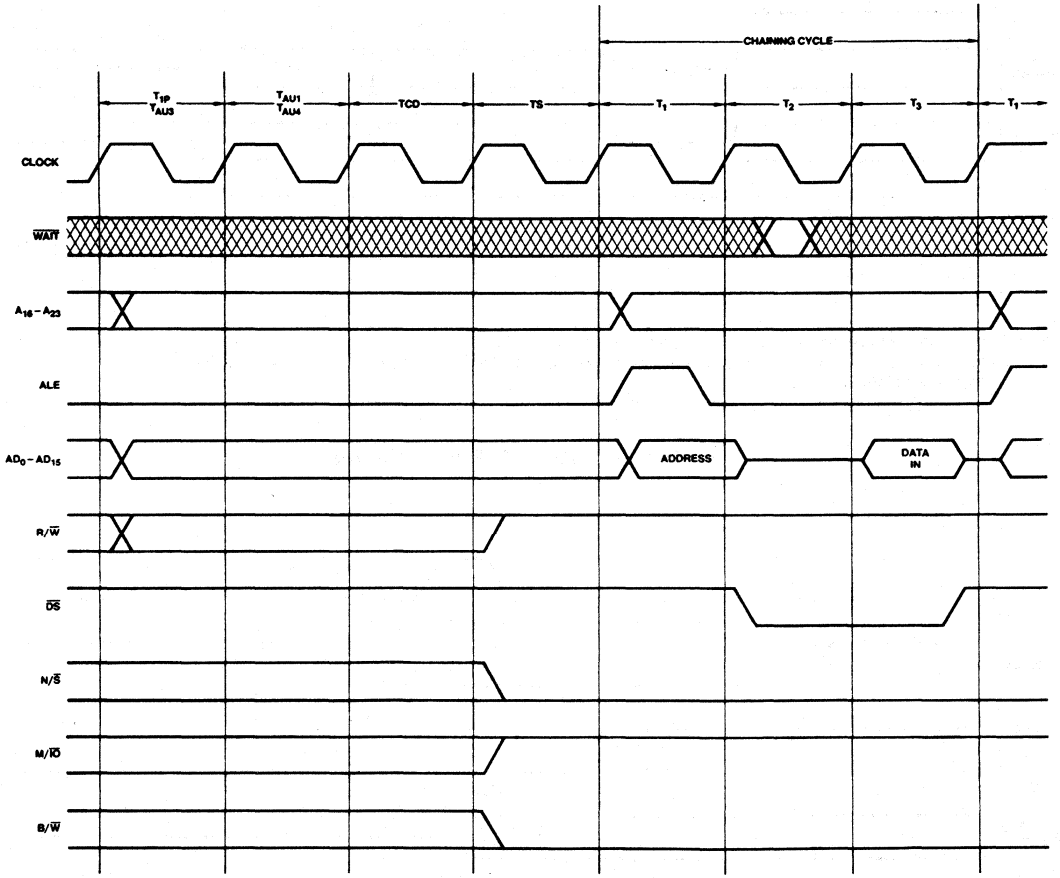
WF007620

TIMING DIAGRAM 13. Clock Waveform



WF007630

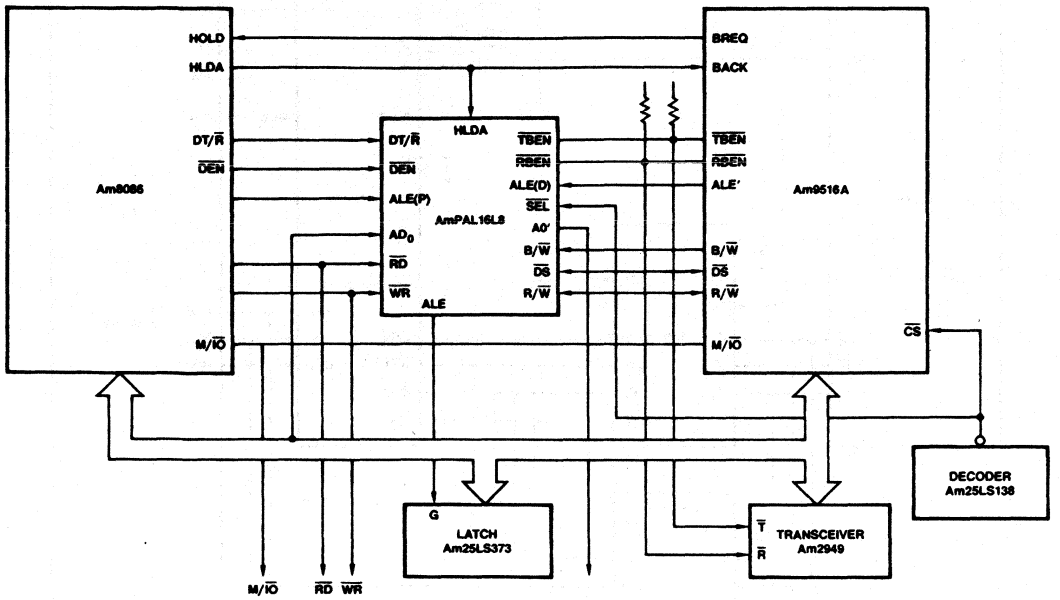
TIMING DIAGRAM 14. Timing During Chaining



WF007640

APPLICATIONS INFORMATION

Figures 21(a) and 21(b) show the configuration of an Am9516A UDC and an Am8086 microprocessor on the same board. Figure 22 shows a configuration for them when the Am9516A UDC is on a different board. The configuration of an Am9516A UDC to 68000 CPU interface is shown in Figure 23. An example of an Am8086 initialization program is shown in Figure 24. Figure 25 shows the reload table for chaining. The details of the Programmable Array Logic (PAL*) for those interfaces are described in Appendix B.



AF003161

Figure 21(a). Am9516A UDC to Am8086 CPU Interface (Minimum Mode)

AmPAL16L8 PALASM FILE

PAL16L8

Pat 001

Am9516A to Am8086 min mode interface chip

Advanced Micro Devices

NC ALED ALEP HLDA BW AD_0 DT/DEN/SEL GND

NC/RBEN/RD ALE A_0/RW/DS/WR/TBEN V_{CC}

If (/HLDA) DS = RD + WR

If (/HLDA) RW = DT

If (/HLDA) TBEN = /DT*/SEL*DEN

If (/HLDA) RBEN = DT*/SEL*DEN

If (HLDA) RD = /RW * DS

If (HLDA) WR = RW * DS

ALE = /ALEP * /ALED

A_0 = /AD_0*/BW*HLDA*ALED +
/AD_0*BW*HLDA*ALED +
/AD_0*/HLAD*ALEP + A_0*/ALEP + A_0*/ALED

DESCRIPTION

This PAL converts the control signals to interface the Am8086 in min mode to the Am9516A DMA controller. Another example shows how this is done in max mode.

*PAL is a trademark of Monolithic Memories, Inc.

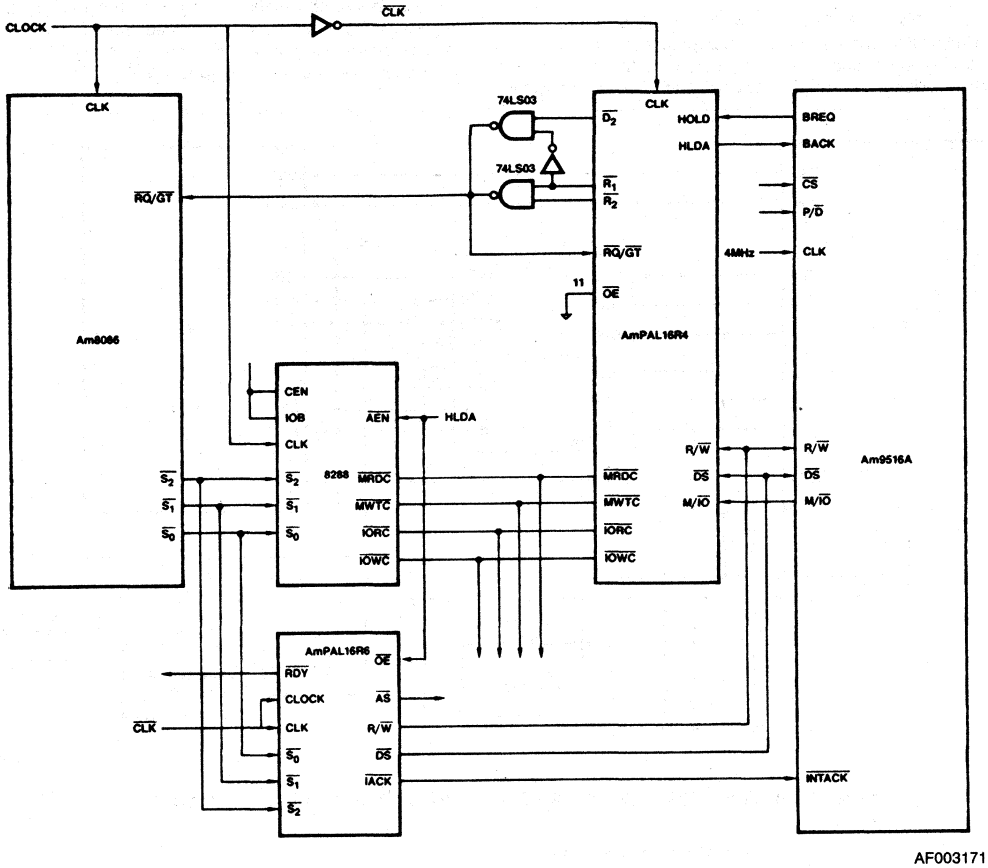
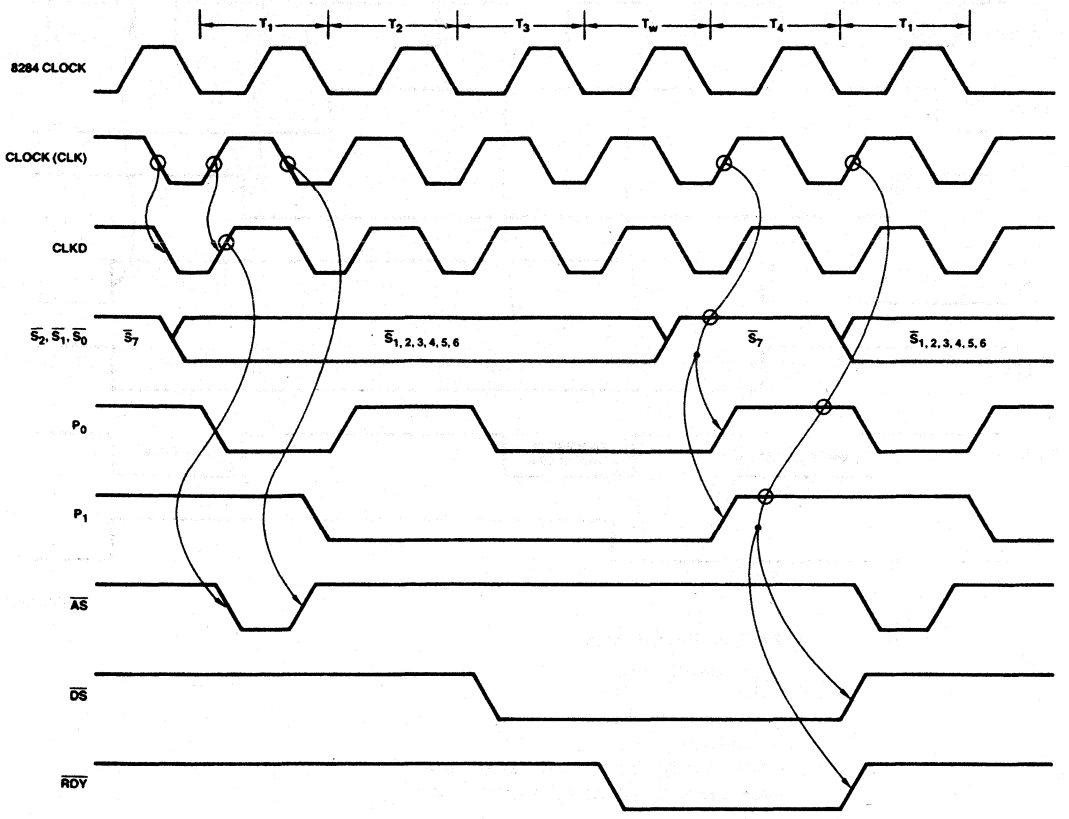


Figure 21(b). Am9516A UDC to 8086 CPU Interface (Maximum Mode)

Timing Diagram of AmPAL16R6



WF007650

AmPAL16R6 PALASM FILE

```
AmPAL16R6
PAT003
Am8086 to AmZ85XX Peripheral Interface
Advanced Micro Devices

CLOCK RESET CLK/S0/S1/S2 NC NC NC GND
/OE/AS/P1/RW/DS/PO/LACK/RDY CLKD VCC

P0: = /RESET*S0*/P0*/P1  +
      /RESET*S1*/P0*/P1  +
      /RESET*S2*/P0*/P1  +
      /RESET*S0*P1       +
      /RESET*S1*P1       +
      /RESET*S2*P1

P1: = /RESET*P0*/P1      +
      /RESET*P1*S0       +
      /RESET*P1*S1       +
      /RESET*P1*S2

DS: = /IACK*/P0*P1*S0*/S1*S2 +
      /IACK*/P0*P1*/S0*S1*S2 +
      IACK*S0*S1*S2      +
      DS*P0*P1

RW: = S0*/S1

IACK: = /RESET*S0*S1*S2 + IACK*P0*P1*/DS +
        IACK*/P0*/P1

RDY: = /RESET*S0*/S1*S2*P0*P1  +
       /RESET*/S0*S1*S2*P0*P1  +
       /RESET*DS*RDY*P0*P1

/CLKD = CLK

AS = /CLKD*P0*/P1*/IACK*CLK
```

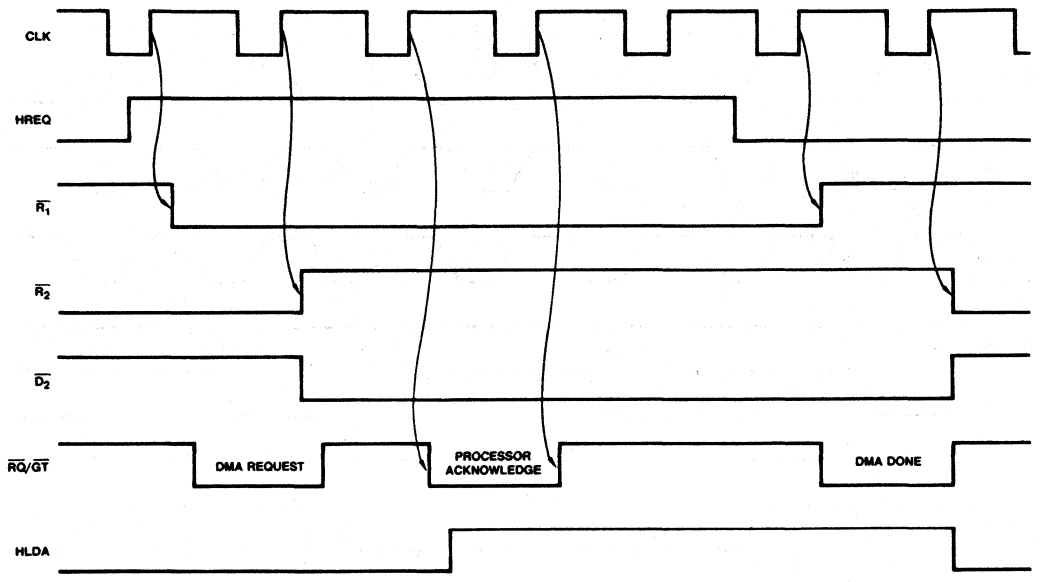
DESCRIPTION

This PAL translates Am8086 bus signals into compatible signals for the Am9516A. It is also applicable to AmZ85XX peripherals by altering/RW and /DS to/RD and /WR. One flip-flop is available to give the necessary delay to the falling edge of/WR.

Note: The CLK signal must be externally inverted for this design.

A >

AmPAL16R4 Timing



WF007660

AmPAL16R4 PALASM FILE

B > Type Am9516A PAL
PAL16R4
Am8086 to Am9516A interface
Advanced Micro Devices
CLK/RQGT HOLD NC NC NC/RW/DS MIO GND
/OE/MWTC/MRDC HLDA/D_2/R_2/R_1/IOWC/IORC V_{CC}

If (HLDA) IORC = /MIO*DS*/RW
If (HLDA) IOWC = /MIO*DS*RW
If (HLDA) MRDC = MIO*DS*/RW
If (HLDA) MWTC = MIO*DS*RW

R_1: = HOLD
R_2: = /R_1
D_2: = R_1
/HLDA: = /R_1 + /D_2*/HLDA + /RQGT*/HLDA

DESCRIPTION

This device converts the min mode signals HOLD and HLDA to the max mode /RQGT protocol. Additionally, it generates the 8288 equivalent control outputs /MRDC, /MWTC, /IORC, and /IOWC. This PAL was used to connect the Am9516A to the Am8086 in max mode.
B >

Note: If HOLD is taken away prior to grant pulse, design will not work correctly because the release pulse will overlap the grant pulse.

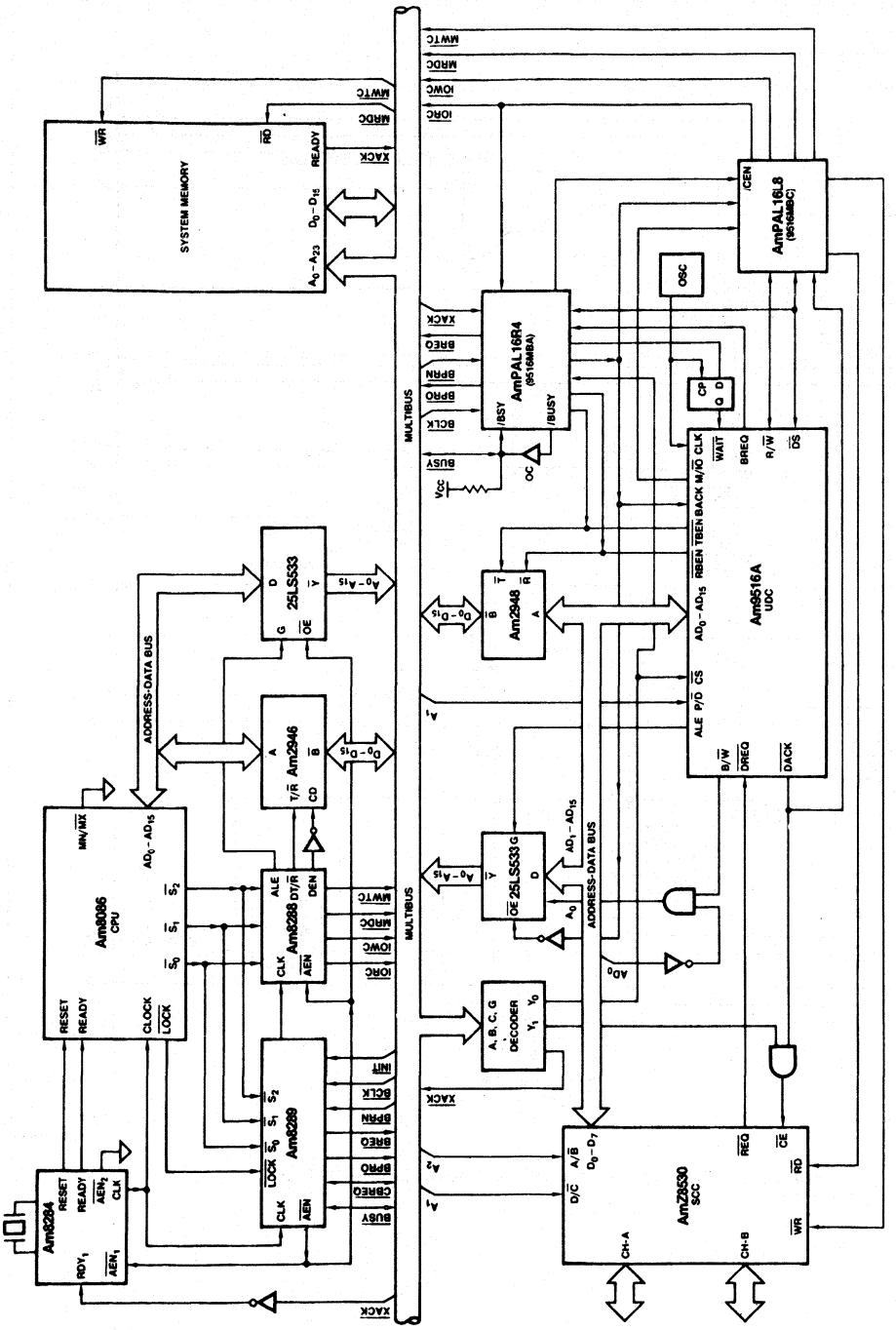
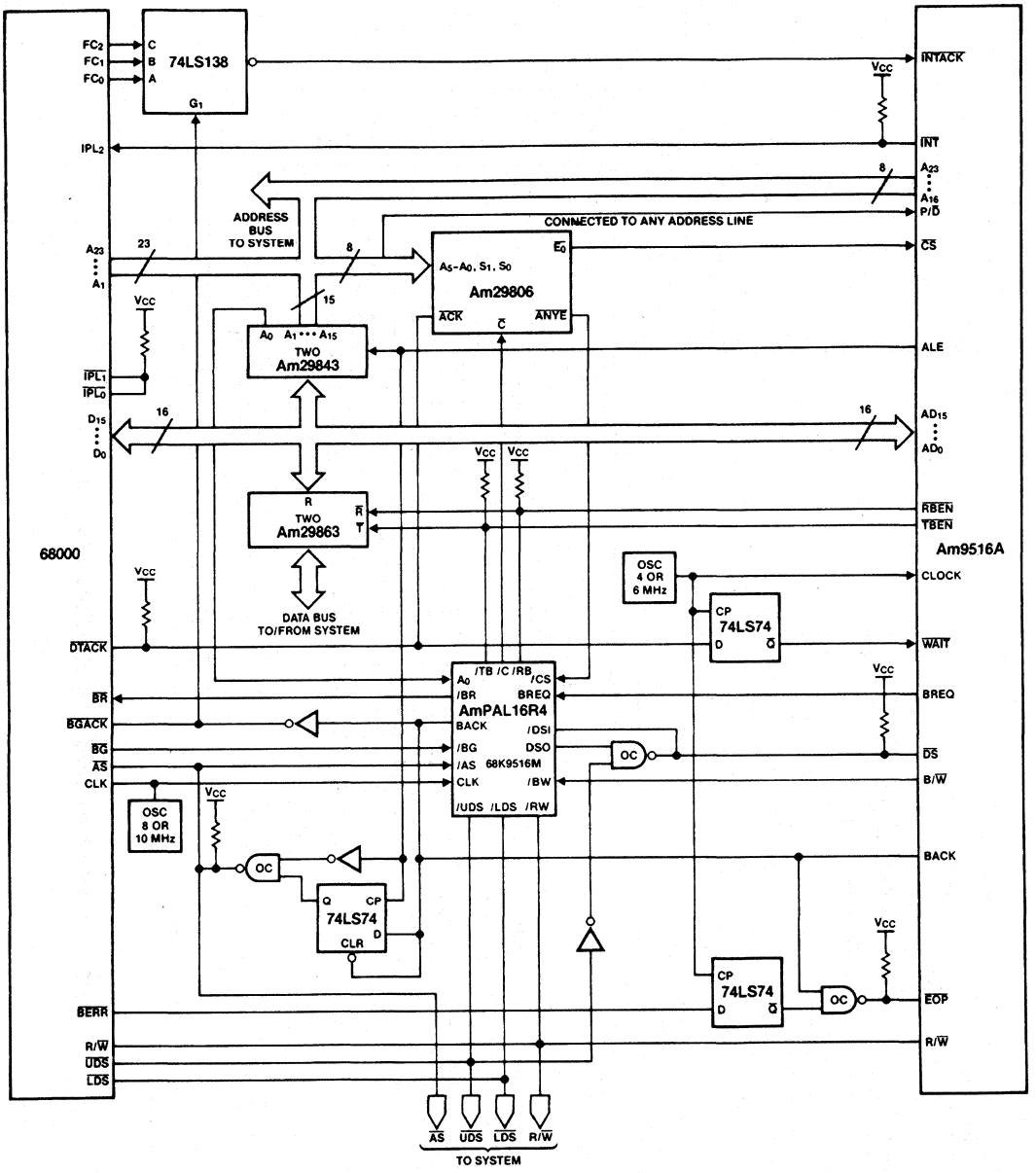


Figure 22. Am9516A to 8086 in a MULTIBUS Environment

03242C

Refer to page 7-1 for Essential Information on Military Devices



BD003941

Figure 23. The Am9516A UDC to 68000 CPU Interface

```
PAL16R4                          PAL DESIGN SPECIFICATION
PAT006                              JOE BRCICH 9/01/83
68000 TO Am9516A INTERFACE WITH DATA HOLD CORRECTION
ADVANCED MICRO DEVICES

CLK   RW    AO    BREQ   /BG    /DSI    /AS    /BW   /CS
/OE   /LDS  /UDS  DSO    /C     BACK    /BR    /TB   /RB

IF (/BACK) RB  =  /CS * RW * UDS      +
                  /CS * RW * LDS
IF (/BACK) TB  =  /CS * /RW
IF (BACK) UDS  =  DSI * /A0 * /BW     +
                  BW * DSI
IF (BACK) LDS  =  DSI * A0 * /BW      +
                  BW * DSI
BR := BREQ * BG * BR * AS             +
      BREQ * /BG * /BACK
/BACK :=   /BREQ                        +
           /BREQ * /BG                  +
           /BREQ * AS                   +
           /BREQ * /BACK                +
           /BG * /BACK                  +
           AS * /BACK
C := UDS * /BACK                      +
     LDS * /BACK
/DSO := BACK                          +
        /BACK * /RW * C
```

DESCRIPTION

IF BREQ*BACK IS TRUE THE Am9516A HAS THE BUS, OTHERWISE THE 68000 HAS THE BUS. THIS PAL CONNECTS THE Am9516A TO THE 68000 WITH ONE WAIT STATE DURING WRITES WHILE SHORTENING /DS TO ACHIEVE PROPER DATA HOLD TIME. IT ALSO CONVERTS THE BUS EXCHANGE PROTOCOL INTO 68000 FORMAT. THIS DESIGN ASSUMES NO OTHER BUS MASTERS IN THE SYSTEM. /RB AND /TB CONTROL THE TRANCEIVERS WHEN CPU IS BUS MASTER. /CS MUST BE FUNCTION OF ALL DEVICES CONNECTED TO THE CPU BUS NOT JUST THE Am9516A /CS AS SHOWN HERE.

The /CS to /DS set-up time of 30ns is met in the following ways:

(1) During a read cycle the only effect from not meeting this set-up time is that the data valid access time from the Am9516A will be delayed by a proportional amount. Since the minimum /DS Low width from the 10-MHz 68000 (during a read) is 193ns and the minimum /DS Low width to the Am9516A is 150ns, we have 43ns margin not counting gate delays which will further increase this margin.

(2) During a write cycle this is not an issue since the /DS comes later and is stretched longer due to the Wait state.

AmPAL16R4 68K9516M PALASM File

```
PAL16L8                    PAL DESIGN SPECIFICATION 9516MBC
PAT 003                          JOE BRCICH 26 JULY 84
MULTIBUS CONTROL FOR Am9516A
ADVANCED MICRO DEVICES

BACK   MIO    NC     NC     /DACK  NC     NC     NC     /CEN   GND
NC     /RD    /IORC  /DS    /MWTC  /MRDC  /IOWC  /RW    /WR    VCC

IF (BACK) IORC  = /MIO * DS * /RW * CEN

IF (BACK) IOWC  = /MIO * DS * RW * CEN

IF (BACK) MRDC  = MIO * DS * /RW * CEN

IF (BACK) MWTC  = MIO * DS * RW * CEN

RD  = DACK * RW * BACK +
      IORC * /BACK

WR  = DACK * /RW * BACK +
      IOWC * /BACK

IF (/BACK) DS  = IORC + IOWC

IF (/BACK) RW  = IOWC
```

DESCRIPTION

THIS PAL CONVERTS MULTIBUS SIGNALS INTO Am9516A COMPATIBLE SIGNALS AND VICE VERSA. IT ALSO SUPPORTS THE 8530 IN FLYBY MODE.

MULTIBUS Control for Am9516A (AmPAL16L8)

```
PAL16R4                    PAL DESIGN SPECIFICATION 9516MBA
PAT 004                          JOE BRCICH 30 July 84
MULTIBUS ARBITER FOR Am9516A
ADVANCED MICRO DEVICES

/BCLK  /XACK  BRQ    /BSY   /BPRN  /DS    NC     /IORC  /CS    GND
/OE    /RBEN  /TBEN  BACK   /CEN   /BREQ  /BUSY  /BPRO  /WAIT  VCC

IF (/BACK) TBEN  = IORC * CS

IF (/BACK) RBEN  = /IORC * CS

WAIT  = /XACK * BACK

BREQ := BRQ

BPRO  = /BRQ * BPRN

/BACK := /BUSY

BUSY := BREQ * BPRN * /BSY * /BUSY  +
        BREQ * BUSY * BPRN          +
        BREQ * BUSY

CEN := BACK
```

DESCRIPTION

/CEN DELAYS THE COMMANDS TO MEET THE MULTIBUS REQUIREMENT THAT ADDRESS AND DATA BE VALID AT LEAST 50NS PRIOR TO CONTROL ACTIVE. /IOWC WAS NOT USED SINCE USING /IORC IMPROVES HOLD TIME. THIS DESIGN DOES NOT SUPPORT THE /CBRQ FUNCTION.

MULTIBUS Arbiter for Am9516A (AmPAL16R4)

```
•                •
•                •
•                •
B0 38            MOV       AL,38H      ;LOADING POINTER OF MASTER
E6 12            OUT       12H         ;MODE REGISTER
B8 07 00         MOV       AX,007H     ;LOADING MMR CODE
E7 10            OUTW      10H         ;
B0 26            MOV       AL,26H      ;LOADING POINTER OF CHAIN
E6 12            OUT       12H         ;ADDRESS REGISTER'S SEGMENT
B8 00 00         MOV       AX,0000H    ;LOADING SEGMENT OF CAR-1
E7 10            OUTW      10H         ;
B0 22            MOV       AL,22H      ;LOADING POINTER OF CHAIN
E6 12            OUT       12H         ;ADDRESS REGISTER'S OFFSET
B8 20 10         MOV       AX,1020H    ;LOADING OFFSET OF CAR-1
E7 10            OUTW      10H         ;
B0 2C            MOV       AL,2CH      ;LOADING POINTER OF COMMAND
E6 12            OUT       12H         ;REGISTER
B0 A0            MOV       AL,A0H      ;LOADING "START CHAIN" COMMAND
E6 10            OUT       10H         ;ISSUING "START CHAIN" COMMAND
•                •
•                •
•                •
```

Figure 24. Initialization Program for 8086 CPU

Notes: The P/$\overline{D}$ input is connected to A1 line; $\overline{CS}$ is decoded from A7 through A4 (all O).

ADDRESS	0	2	4	6	8	A	C	E
1000	0000	1020	0000	1020	0007	0005	0006	0005
1010	0002	AAAA	0009	00A0	0004	0042	0042	0001
1020	03FF	0000	1F00	0000	1080	0010	0000	1F00
1030	0000	1080	0012	0000	FFFF	0001	0000	8020
1040	0000	1020	1111	1111	0000	FFFF	2004	0000
1050	0010	0000	0000	1020	0018	1020	2222	1007
1060	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA
1070	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA

Reload Word

TB000084

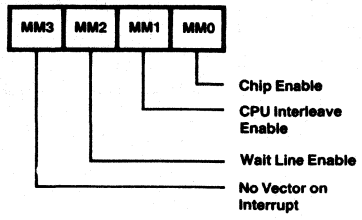
Figure 25. Reload Table for Chaining

APPENDIX A
UDC REGISTER SUMMARY

Master Mode Register

	A7							A0
Address	X	0	1	1	1	0	0	X

Fast Readable
Writable



DF003470

Miscellaneous Registers

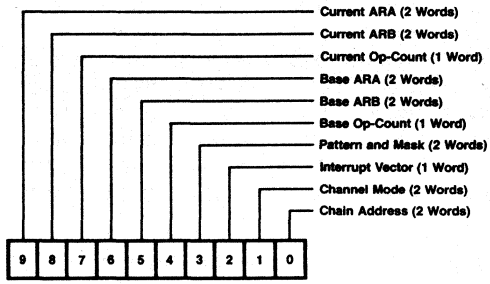
	A7							A0		
Address	X	0	1	1	0	0	1	X	Current Operation Count	CH1
	X	0	1	1	0	0	0	X	Current Operation Count	CH2
	X	0	1	1	0	1	1	X	Base Operation Count	CH1
	X	0	1	1	0	1	0	X	Base Operation Count	CH2
	X	1	0	0	1	0	1	X	Pattern	CH1
	X	1	0	0	1	0	0	X	Pattern	CH2
	X	1	0	0	1	1	1	X	Mask	CH1
	X	1	0	0	1	1	0	X	Mask	CH2

Chain Loadable
Writable
Pattern and Mask – Slow Readable
Operation Count – Fast Readable

D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Chain Control Register

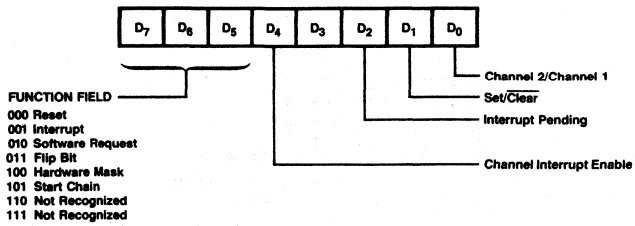
Chain Loadable Only



DF003480

Command Register

	A7							A0	
Address	X	0	1	0	1	1	1	X	CH1
Writable Only	X	0	1	0	1	1	0	X	CH2

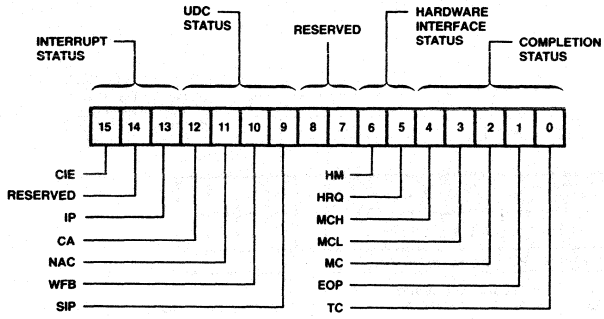


DF003490

Status Register

	A7						A0	
Address	X	0	1	0	1	1	1	X CH1
	X	0	1	0	1	1	0	X CH2

Fast Readable

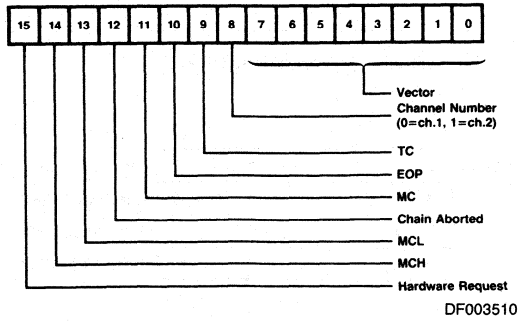


DF003500

Interrupt Save Register

	A7						A0	
Address	X	0	1	0	1	0	1	X CH1
	X	0	1	0	1	0	0	X CH2

Fast Readable

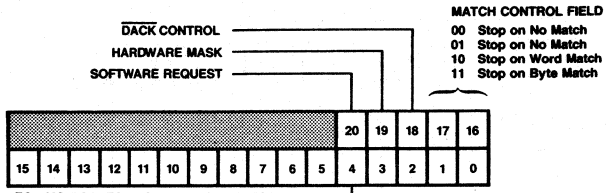


DF003510

Channel Mode Register

A7							A0	
X	1	0	1	0	1	1	X	High CH1
X	1	0	1	0	1	0	X	High CH2
X	1	0	1	0	0	1	X	Low CH1
X	1	0	1	0	0	0	X	Low CH2

Chain Loadable
Writable (Lower 16 bits)
Slow Readable



TRANSFER TYPE FIELD

Code	Transfer Type
00	Single Transfer
01	Demand Dedicated/Bus Hold
10	Demand Dedicated/Bus Release
11	Demand Interleave

FLIP BIT

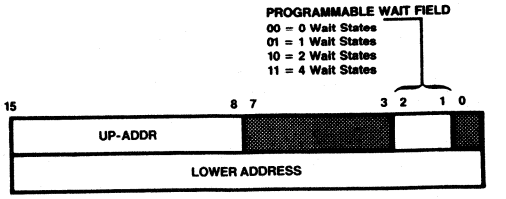
0 – ARA = SRC, ARB = DEST
1 – ARA = DEST, ARB = SRC

Code/Operation	Operand Size ARA	ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

DF003520

Chain Address Register

	A7							A0	
Address	X	0	1	0	0	1	1	X	Up-Addr CH1
	X	0	1	0	0	1	0	X	Up-Addr CH2
	X	0	1	0	0	0	1	X	Low-Addr CH1
	X	0	1	0	0	0	0	X	Low-Addr CH2

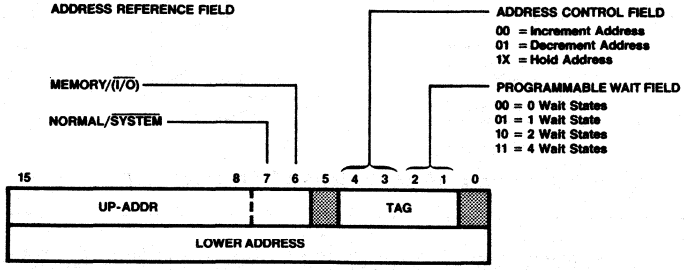


DF003530

Address Registers

Address	A7							A0		
	X	0	0	1	1	0	1	X	Current ARA Up-Addr/Tag	CH1
	X	0	0	1	1	0	0	X	Current ARA Up-Addr/Tag	CH2
	X	0	0	0	1	0	1	X	Current ARA Low-Addr	CH1
	X	0	0	0	1	0	0	X	Current ARA Low-Addr	CH2
	X	0	0	1	0	0	1	X	Current ARB Up-Addr/Tag	CH1
	X	0	0	1	0	0	0	X	Current ARB Up-Addr/Tag	CH2
	X	0	0	0	0	0	1	X	Current ARB Low-Addr	CH1
	X	0	0	0	0	0	0	X	Current ARB Low-Addr	CH2
	X	0	0	1	1	1	1	X	Base ARA Up-Addr/Tag	CH1
	X	0	0	1	1	1	0	X	Base ARA Up-Addr/Tag	CH2
	X	0	0	0	1	1	1	X	Base ARA Low-Addr	CH1
	X	0	0	0	1	1	0	X	Base ARA Low-Addr	CH2
	X	0	0	1	0	1	1	X	Base ARB Up-Addr/Tag	CH1
	X	0	0	1	0	1	0	X	Base ARB Up-Addr/Tag	CH2
	X	0	0	0	0	1	1	X	Base ARB Low-Addr	CH1
	X	0	0	0	0	1	0	X	Base ARB Low-Addr	CH2

Chain Loadable
Fast Readable and Writable

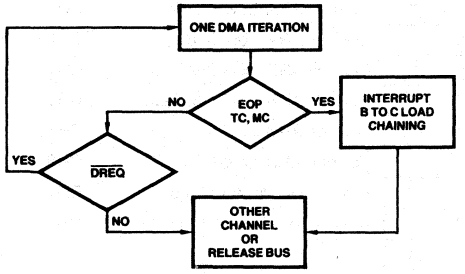


DF003540

APPENDIX B

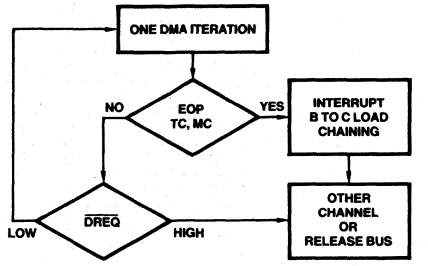
Flow Charts of DMA Operations:
Figure B-1 shows the basic DMA operations with software or hardware request. The Demand Interleave operations are shown in Figure B-2.

(a) Single Operation



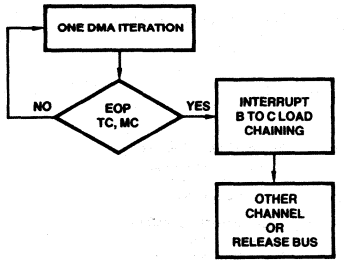
PF001250

(c) Demand Dedicated with Bus Release (Hardware Request)



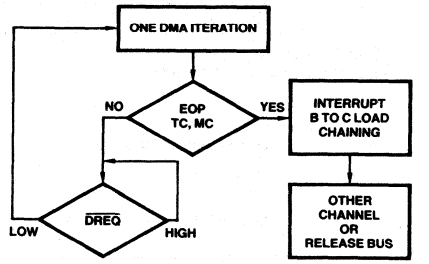
PF001260

(b) Demand Operation when Software Requesting



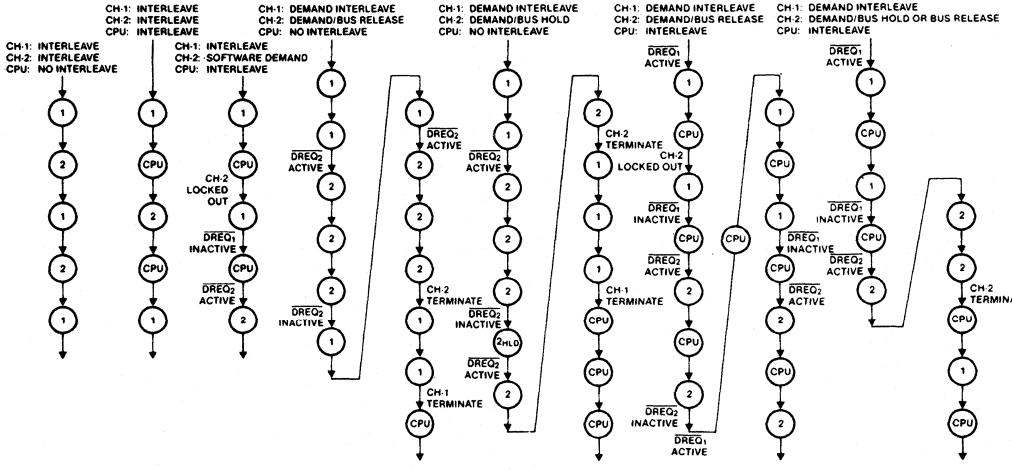
PF001270

(d) Demand Dedicated with Bus Hold (Hardware Request)



PF001280

Figure B-1. Basic DMA Operations of Am9516A UDC



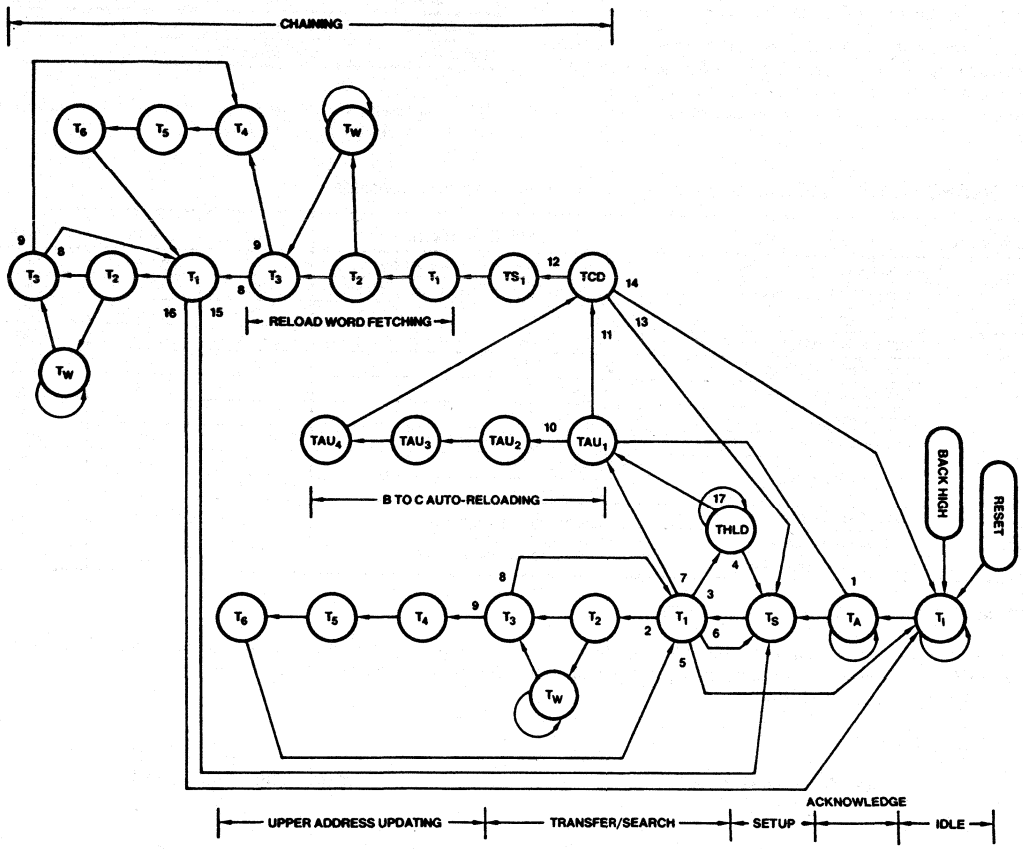
PF001300

Figure B-2. Demand Interleave Operations of Am9516A UDC

**Multibus is a trademark of Intel Corp.

APPENDIX C

Am9516A STATE DIAGRAM



AF003180

Am9516A INTERNAL OPERATION ROUTINES

1. "Start Chain" command issued or start updating routine* after an interrupt has been served.**
2. Normal DMA operation.
3. Demand with Bus hold while DREQ is inactive.
4. DREQ is active while bus held.
5. Single transfer, CPU interleave enabled, or demand with bus release while current DREQ is inactive and no DMA request is pending.
6. Single Transfer or Demand/Bus release while current DREQ is inactive, but the other DMA request is pending.
7. TC, MC or EOP termination occurs.
8. One DMA or chain transaction is done and the upper address is not changed.
9. One DMA or chain transaction is done and the upper address is changed.
10. Base-to-current auto-reloading is enabled.
11. Base-to-current auto-reloading is disabled.
12. Chaining is enabled.
13. Chaining is disabled and another DMA request is pending.
14. Chaining is disabled and no DMA request is pending.
15. Chaining ends and another DMA request is pending.
16. Chaining ends and no DMA request is pending.
17. EOP termination of Bus Hold.

*Updating routine includes base-to-current auto-reloading and chaining.

**When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit of a Status register is set and the channel relinquishes the bus until the first interrupt has been served. If the channel was to perform the updating routine, once the SIP bit is cleared, DTC will reacquire the bus and perform the appropriate operation (i.e., 1).

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5V to +7.0V
All Signal Voltages with Respect to V_{SS} .. −0.5V to +7.0V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5.0V±5%	0V
Industrial	−40°C to 85°C	5.0V±10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

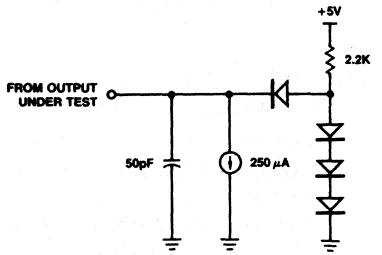
Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	3.8	V_{CC} +0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.5	0.45	Volts
V_{IH}	Input High Voltage		2.0	V_{CC} +0.3	Volts
V_{IL}	Input Low Voltage		−0.5	0.8	Volts
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		Volts
V_{OL}	Output Low Voltage	I_{OL} = 3.2		0.4	Volts
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$		±10	μA
I_{OL}	Output Leakage	$V_{SS} \leq V_{OUT} \leq V_{CC}$		±10	μA
I_{CC}	V_{CC} Supply Current	T_A = 0°C		350	mA
		T_A = 70°C		200	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.		10	pF
C_{OUT}	Output Capacitance			15	pF
$C_{I/O}$	Bidirectional Capacitance			20	pF

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

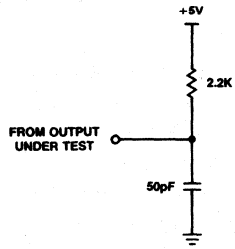
+4.75 ≤ V<CC> ≤ +5.25V
GND = 0V
0°C ≤ T_A ≤ +70°C

Standard Test Load



TC002010

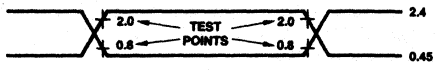
Open-Drain Test Load



TC002020

TIMING REFERENCES FOR AC TESTS

Input Waveform



WF007690

Output Waveform

WF007700

All AC parameters assume a load capacitance of 100pF max, except for parameter 6 TdC(SNv) (50pF max).

03242C

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING CHARACTERISTICS
TIMING FOR UDC AS BUS MASTER

			Advanced Information								
			4MHz		6MHz		8MHz		10MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
1	TcC	Clock Cycle Time	250	2000	165	2000	125		100		ns
2	TwCh	Clock Width (HIGH)	105	1000	70	1000	55		45		ns
3	TwCl	Clock Width (LOW)	105	∞	70	∞	55		45		ns
4	TfC	Clock Fall Time		20		10		5		5	ns
5	TrC	Clock Rise Time		20		15		10		5	ns
6	TdC(AUv)	Clock RE to Upper Address (A_{16}-A_{23}) Valid Delay		90		80		60		50	ns
7	ThC(AUv)	Clock RE to Upper Address Valid Hold Time	20		10		10		10		ns
8	TdC(ST)	Clock RE to R/$\overline{W}$ and B/$\overline{W}$ Valid Delay		110		90		60		50	ns
9	TdC(A)	Clock RE to Lower Address (A_0-A_{15}) Valid Delay		90		90		60		50	ns
10	TdC(Az)	Clock RE to Lower Address (A_0-A_{15}) Float Delay		60		60		50		40	ns
11	TdC(ALr)	Clock RE to ALE RE Delay		70		60		40		30	ns
12	TdC(AL)	Clock FE to ALE FE Delay		70		60		65		55	ns
13	TdC(DS)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60		50		40	ns
14	TdC(DSf)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60		60		50	ns
15	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		60		60		60		60	ns
16	TdC(DO)	Clock RE to Data Out Valid Delay		90		90		65		60	ns
17	TsDI(C)	Data in to Clock FE Set-up Time	20		15		10		10		ns
18	TdA(AL)	Address Valid to ALE FE Delay	50		35		30		30		ns
19	ThAL(A)	ALE FE to Lower Address Valid Hold Time	60		40		30		30		ns
20	TwAL	ALE Width (HIGH)	80		60		45		40		ns
21	TdAz(DS)	Lower Address Float to $\overline{DS}$ FE Delay	0		0		0		0		ns
22	TdAL(DS)	ALE FE to $\overline{DS}$ (Read) FE Delay	75		35		45		40		ns
23	TdAL(DI)	ALE FE to Data in Required Valid Delay		300		215		205		175	ns
24	TdA(DI)	Address Valid to Data in Required Valid Delay		410		305		260		230	ns
25	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		50		40		ns
26	TdDS(AI)	$\overline{DS}$ RE 10 ALE RL Delay	75		40		45		35		ns
27	TdA(DS)	Address Valid to $\overline{DS}$ (Read) FE Delay	160		110		90		70		ns
28	TdDO(DSr)	Data Out Valid to $\overline{DS}$ RE Delay	230		200		150		130		ns
29	TdDO(DSf)	Data Out Valid to $\overline{DS}$ FE Delay	55		35		40		35		ns
30	ThDS(DO)	$\overline{DS}$ RE to Data Out Valid Hold Time	85		45		40		25		ns
31	TdDS(DI)	$\overline{DS}$ (Read) FE to Data in Required Valid Delay		205		155		150		140	ns
33	ThDI(DS)	$\overline{DS}$ RE to Data in Hold Time	0		0		0		0		ns
34	TwDSmw	$\overline{DS}$ (Write) Width (LOW)	185		110		120		95		ns
35	TwDSmr	$\overline{DS}$ (Read) Width (LOW)	275		220		160		130		ns
36	TdC(RBr)	Clock FE to $\overline{RBEN}$ RE Delay*		70		65		30		30	ns
37	ThDS(ST)	$\overline{DS}$ RE to B/$\overline{W}$, N/$\overline{S}$, R/$\overline{W}$ and M/$\overline{IO}$ Valid Hold Time	75		45		40		25		ns
38	TdC(TRf)	Clock RE to $\overline{TBEN}$ or $\overline{RBEN}$ FE Delay		60		60		45		35	ns
39	TdC(TRr)	Clock RE to $\overline{TBEN}$ RE Delay		60		60		45		45	ns
40	TdC(ST)	Clock RE to M/$\overline{IO}$ and N/$\overline{S}$ Valid Delay		90		75		65		50	ns
41	TdS(AL)	R/$\overline{W}$, M/$\overline{IO}$, B/$\overline{W}$ and N/$\overline{S}$ Valid to ALE FE Delay	60		35		45		35		ns
42	TsWT(C)	$\overline{WAIT}$ to Clock FF Set-up Time	20		20		10		10		ns
43	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	20		20		35		35		ns
44	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		20		20		ns
45	TsDRQ(C)	$\overline{DREQ}$ Valid to Clock RE Set-up Time	60		50		30		20		ns
46	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		20		20		ns
47	TdC(INTf)	Clock FE to $\overline{INT}$ FE Delay		150		150		105		105	ns

*These must not occur simultaneously.

Note: RE = rising edge
FE = falling edge

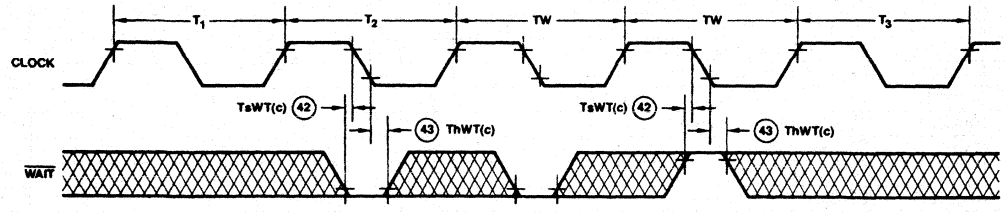
Am9516A CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

The parameters listed below are also shown in the Switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence so that the exact limit for these parameters may be determined for any given system in relation to its specific clock characteristics.

Number	Parameters	Derivation
18	TdA (AL)	0.5TcC – #9 + (#12 – tr)
19	ThAL (A)	0.5TcC – #12 (ALE FE @ 0.8V) + #10
21	TdAz (DS)	#13 – #10
22	TdAL (DS)	0.5TcC – #12 + #13
23	TdAL (DI)	2TcC – #12 – #17
24	TdA (DI)	2.5TcC – #9 – #17
25	TdDS (A)	0.5TcC – #15 + #9
26	TsDS (AL)	0.5TcC – #15 + #11 (ALE RE)
27	TdA (DS)	TcC – #9 + #13
28	TdDO (DSr)	1.5TcC – #16 + #15
29	TdDO (DSf)	0.5TcC – #16 + #14
30	ThDS (DO)	0.5TcC – #15 + #32
31	TdDS (DI)	1.5TcC – #13 – #17
34	TwDSmw	TcC – #14 + #15
35	TwDSmr	1.5TcC – #13 + #15
37	ThDS (ST)	0.5TcC – #15 + (#40 – tr)
41	TdS (AL)	0.5TcC – #40 + (#12 – tr)

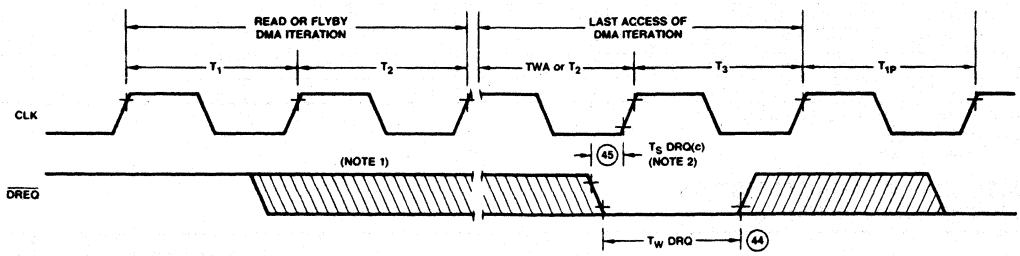
NOTE: tr (nominal) = 10ns
#32 CLCK RE to Data Out Not Valid Delay = 20ns (4 and 6 MHz)

$\overline{\text{WAIT}}$ Timing



WF007680

Sampling $\overline{\text{DREQ}}$ During Single Transfer DMA Operations



WF007670

Notes: 1. HIGH-to-LOW $\overline{\text{DREQ}}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.

2. A HIGH-to-LOW $\overline{\text{DREQ}}$ transition must meet the conditions in Note 1 and must occur TsDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{\text{DREQ}}$ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.

3. Flyby and Search transactions have only a single access; parameter TsDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

SWITCHING CHARACTERISTICS
UDC AS BUS SLAVE BUS EXCHANGE

			Advanced Information								
			4MHz		6MHz		8MHz		10MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
61	TdIN(DO)	$\overline{INTACK}$ FE to Data Output Valid Delay		135		135		120		110	ns
62	TdIN(DOz)	$\overline{INTACK}$ RE to Data Output Float Delay		80		80		45		35	ns
63	TdDS(DO)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135*		135		120		110	ns
64	TdDS(DOz)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		80		45		35	ns
65	TsDI(DS)	Data Valid to $\overline{DS}$ RE (IOW) Set-up Time	40		40		40		35		ns
66	ThDS(DI)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		30		0		0		ns
67	TwDS	$\overline{DS}$ Low Width	150*		150*		125		100		ns
68	TwIN	$\overline{INTACK}$ Low Width	150		150		125		100		ns
69	ThDS(CS)	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	20		20		15		10		ns
70	ThDS(PD)	$\overline{DS}$ RE to P/$\overline{D}$ Valid Hold Time	20		20		15		10		ns
71	TsPD(DS)	P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	10		10		10		10		ns
		P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	50		50		40		30		
72	TsCS(DS)	$\overline{CS}$ Valid to $\overline{DS}$ FE Set-up Time	30		30		20		10		ns
73	TrDS	$\overline{DS}$ RE to $\overline{DS}$ FE Recovery Time (for Commands Only)	4TcC		4TcC		4Tcc		4Tcc		ns
74	TwRST	$\overline{RESET}$ Low Width	3TcC		3TcC		3Tcc		3Tcc		ns
75	TdC(BRQf)	Clock RE to BREQ RE Delay		150		150		125		100	ns
76	TdC(BRQr)	Clock FE to BREQ FE Delay		165		150		125		100	ns
77	TdBRQ(CTRz)	BREQ FE to Control Bus Float Delay		140		140		100		60	ns
78	TdBRQ(ADz)	BREQ FE to AD Bus Float Delay		140		140		100		60	ns
79	TdBRQ(BAK)	BREQ RE to BACK RE Required Delay	0		0		0		0		ns
80	TsBAK(C)	BACK Valid to Clock RE Set-up Time	50		45		30		20		ns
81	TdBAK(ADz)	BACK FE to A and AD Buses Float Delay (Reset)		135		135		125		100	ns
82	TdBAK(CTRz)	BACK FE to Control Bus Float Delay (Reset)		100		100		100		75	ns
83	TdBAK(DSz)	BACK FE to $\overline{DS}$ Float Delay (Reset)		90		90		80		60	ns
84	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	2		2		2		2		ns
85	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOW)	−10		−10		−10		−10		ns
86	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	20		20		15		15		ns
87	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOR)	20		20		15		15		ns

*2000ns for slow readable registers (worst case)

Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS
UDC-PERIPHERAL INTERFACE

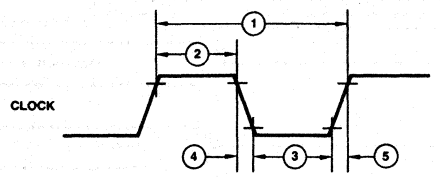
			Advanced Information								
			4MHz		6MHz		8MHz		10MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
90	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		85		40		50	ns
91	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (To Flyby Transactions Only)		100		85		40		50	ns
92	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (FROM Flyby Transactions Only)	10		10		10		10		ns
93	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		85		50		60	ns
94	TDAH	Clock FE to Level $\overline{DACK}$ Valid Hold Time		100		85		50		60	ns
95	TEIDL	Clock FE to Internal $\overline{EOP}$ LOW Delay		110		90		70		80	ns
96	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		90		70		80	ns
97	TES	External $\overline{EOP}$ Valid to Clock FE Set-up Time During Operation	10		10		10		10		ns
98	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		20		20		ns
99	TES(BH)	External $\overline{EOP}$ Valid to Clock FE Set-up Time During Bus Hold	10		10		10		10		ns
100	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		20		20		ns

Note: RE = rising edge
FE = falling edge

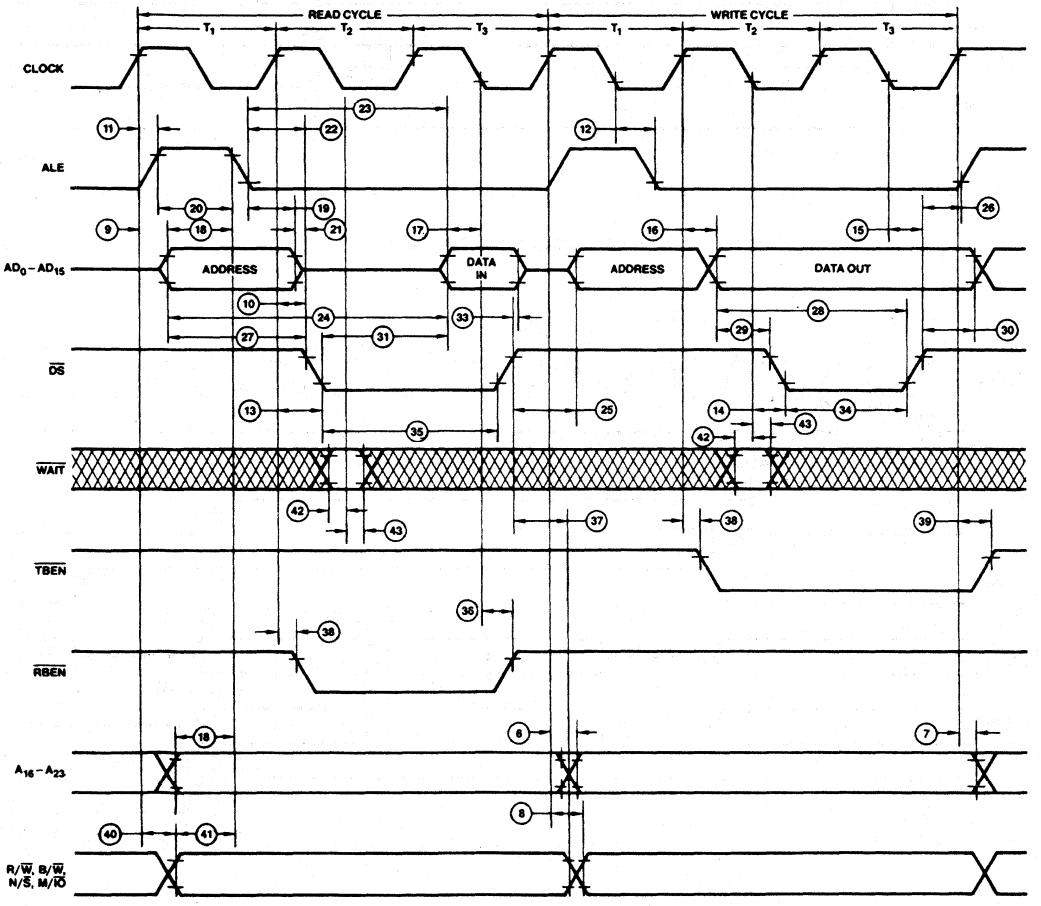
03242C

Refer to page 7-1 for Essential Information on Military Devices

AC Timing when UDC is a Bus Master

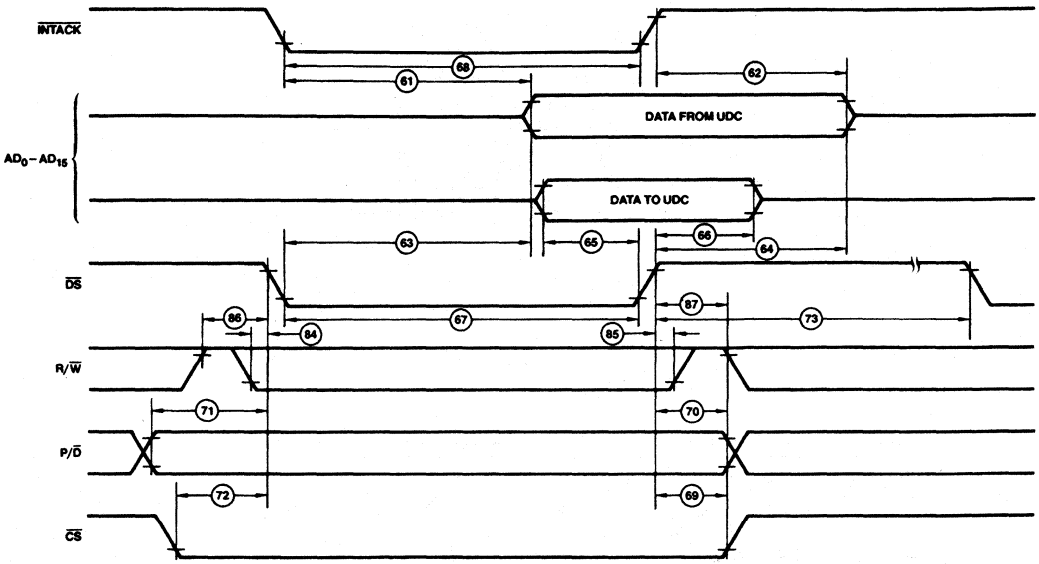


WF007630



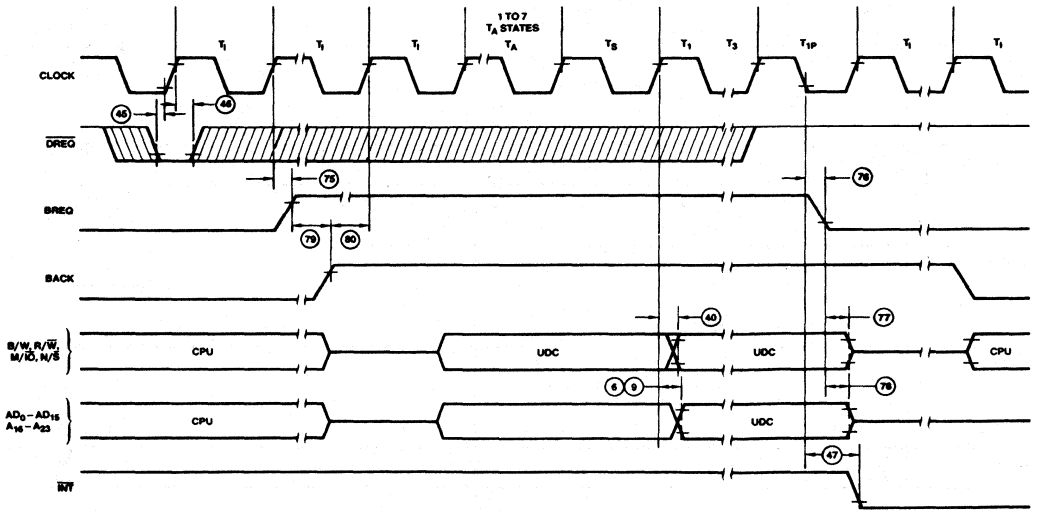
WF007710

AC Timing when UDC is a Bus Slave



WF007720

Bus Exchange Timing



WF007730

Am9517A

Multimode DMA Controller

DISTINCTIVE CHARACTERISTICS

- Four independent DMA channels, each with separate registers for Mode Control, Current Address, Base Address, Current Word Count and Base Word Count
- Transfer modes: Block, Demand, Single Word, Cascade
- Independent Autoinitialization of all channels
- Memory-to-memory transfers
- Memory block initialization
- Address increment or decrement
- Master system disable
- Enable/disable control of individual DMA requests
- Directly expandable to any number of channels
- End of Process input for terminating transfers
- Software DMA requests
- Independent polarity control for DREQ and DACK signals
- Compressed timing option speeds transfers – up to 2.5M bytes/second
- +5 volt power supply
- Advanced N-channel silicon gate MOS technology
- 40-pin Hermetic DIP package
- New 9517A-5 5MHz version for higher speed CPU compatability

GENERAL DESCRIPTION

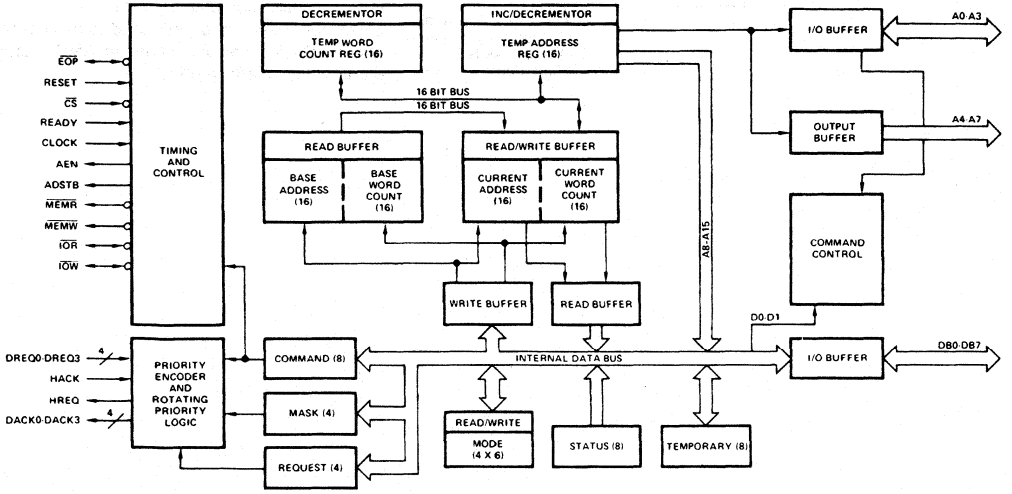
The Am9517A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The Am9517A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The Am9517A is designed to be used in conjunction with an external 8-bit address register such as the Am74LS373. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process ($\overline{EOP}$).

Each channel has a full 64K address and word count capability. An external $\overline{EOP}$ signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

BLOCK DIAGRAM



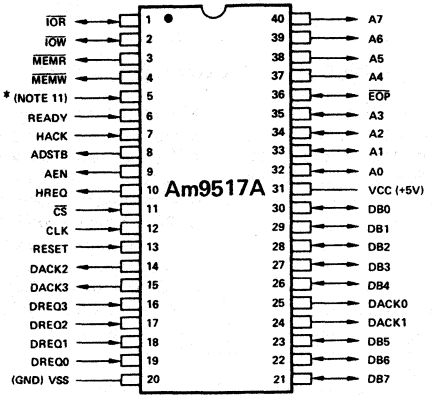
BD003250

03040B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40, P-40

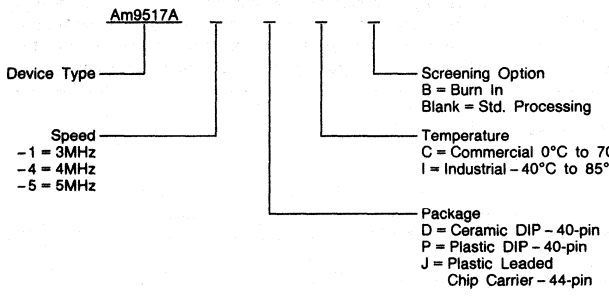


CD005071

Note: Pin 1 is marked for orientation
*See Note 11 under DC Characteristics table.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9517A-1 Am9517A-4 Am9517A-5	PC, DC, DI, DIB, PCB, DCB
Am9517A-1	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
31	V_{CC}		Power: +5 Volt supply.
20	V_{SS}		Ground.
12	CLK	I	Clock Input: Clock Input controls the internal operations of the Am9517A and its rate of data transfers. The input may be driven at up to 3MHz for the standard Am9517A and up to 5MHz for the Am9517A-5.
11	$\overline{CS}$	I	Chip Select: Chip Select is an active low input used to select the Am9517A as an I/O device during the Idle cycle.This allows CPU communication on the data bus.
13	RESET	I	Reset: Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the First/Last Flip/Flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
6	READY	I	Ready: Ready is an input used to extend the memory read and write pulses from the Am9517A to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified setup/hold time.
7	HACK	I	Hold Acknowledge: The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system buses.
19-16	DREQ0-DREQ3	I	DMA Request: The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
30-26, 23-21	DB0-DB7	I/O	DATA Bus: The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the Am9517A control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the Am9517A on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
1	$\overline{IOR}$	I/O	I/O Read: I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the Am9517A to access data from a peripheral during a DMA Write transfer.
2	$\overline{IOW}$	I/O	I/O Write: I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to load information into the Am9517A. In the Active cycle, it is an output control signal used by the Am9517A to load data to the peripheral during a DMA Read transfer.
36	$\overline{EOP}$	I/O	End of Process: End of Process is an active low bidirectional open-drain signal. Information concerning the completion of DMA service is available at the bidirectional $\overline{EOP}$ pin. The Am9517A allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ input low with an external $\overline{EOP}$ signal. The Am9517A also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ Line. The reception of $\overline{EOP}$, either internal or external, will cause the Am9517A to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prvent erroneous end of process inputs.
32-35	A0-A3	I/O	Address: The four least significant address lines are bidirectional three-state signals. In the Idle cycle, they are inputs and are used by the CPU to address the registers to be load or read. In the Active cycle, they are outputs and provide the lower 4 bits of the output address.
37-40	A4-A7	O	Address: The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during DMA service.
10	HREQ	O	Hold Request: This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes the Am9517A to issue the HRQ. After HRQ goes active, at least one clock cycle (TCY) must occur before HLDA goes active.
25, 24 14, 15	DACK0-DACK3	O	DMA Acknowledge: DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
9	AEN	O	Address Enable. Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable in other system bus drivers during DMA transfers. AEN is active-high.
8	ADSTB	O	Address Strobe. The active-high Address Strobe is used to strobe the upper address byte into an external latch.
3	$\overline{MEMR}$	O	Memory Read: The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
4	$\overline{MEMW}$	O	Memory Write: The Memory Write signal is an active low three-state output used to write data to the selected memory location during a DMA Write or a a memory-to-memory transfer.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Am9517 Internal Registers.

DETAILED DESCRIPTION

The Am9517A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The Am9517A contains 344 bits of internal memory in the form of registers. The table shown above lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

The Am9517A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the Am9517A. The Program Command Control block decodes the various commands given to the Am9517A by the microprocessor prior to servicing a DMA Request. It also decodes each channel's Mode Control word. The Priority Encoder block resolves priority contention among DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In Am9080A systems this input will usually be the $\phi 2$ TTL clock from an Am8224. However, any appropriate system clock will suffice.

DMA Operation

The Am9517A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The Am9517A can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the Am9517A has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The Am9517A has requested a hold but the processor has not yet returned an acknowledge. An acknowledge from the CPU will signal that transfers may begin. S1, S2, S3, and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted before S4 by the use of the Ready line on the Am9517A.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for each complete transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23 and S24) for the write-to-memory half of the transfer. The Temporary Data register is used for intermediate storage of the memory byte.

Idle Cycle

When no channel is requesting service, the Am9517A will enter the Idle cycle and perform "SI" states. In this cycle the Am9517A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the Am9517A. When $\overline{CS}$ is LOW and HACK is LOW, the Am9517A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0-A3 are inputs to the device and select which registers will be read or written. The $\overline{IOR}$ and $\overline{IOW}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip/flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip/flop is reset by Master Clear or Reset. A separate software command can also reset this flip/flop.

Special software commands can be executed by the Am9517A in the Program Condition. These commands are decoded as sets of addresses when both $\overline{CS}$ and $\overline{IOW}$ are active and do not make use of the data bus. Functions include Clear First/Last Flip/Flop and Master Clear.

Active Cycle

When the Am9517A is in the idle cycle and a channel requests a DMA service, the device will output a HREQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place in one of four modes:

Single Transfer Mode: In Single Transfer mode, the Am9517A will make a one-byte transfer during each HREQ/HACK handshake. When DREQ goes active, HREQ will go active. After the CPU responds by driving HACK active, a one-byte transfer will take place. Following the transfer, HREQ will go inactive, the word count will be decremented and the address will be either incremented or decremented. When the word count goes to zero, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

To perform a single transfer, DREQ must be held active only until the corresponding DACK goes active. If DREQ is held continuously active, HREQ will go inactive following each transfer and then will go active again and a new one-byte transfer will be made following each rising edge of HACK. In 8080A/Am9080A systems, this will ensure one full machine cycle of execution between DMA transfers. Details of timing between the Am9517A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode: In Block Transfer mode, the Am9517A will continue making transfers until a TC (caused by the word count going to zero) or an external End of Process ($\overline{EOP}$) is encountered. DREQ need be held active only until DACK becomes active. An Autoinitialize will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode: In Demand Transfer mode the device will continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus, the device requesting service may discontinue transfers by bringing DREQ inactive. Service may be resumed by asserting an active DREQ once again. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count may be read from the Am9517A Current Address and Current Word Count registers. Autoinitialization will only occur following a TC or $\overline{EOP}$ at the end of service. Following Autoinitialization, an active-going DREQ edge is required to initiate a new DMA service.

Cascade Mode: This mode is used to cascade more than one Am9517A together for simple system expansion. The HREQ and HACK signals from the additional Am9517A are connected to the DREQ and DACK signals of a channel of the initial Am9517A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel in the initial device is used only for prioritizing the additional device, it does not output any address or control signals of its own. These would conflict with the outputs of the active channel in the added device. The Am9517A will respond to DREQ with DACK but all other outputs except HREQ will be disabled.

Figure 1 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More Am9517As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices forming a third level.

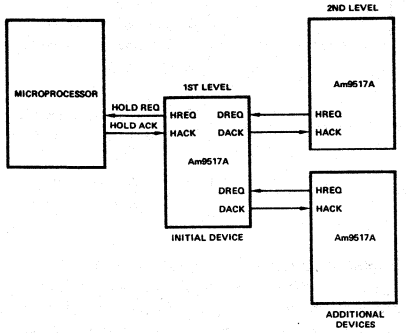


Figure 1. Cascaded Am9517As

Transfer Types

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{IOR}$ and $\overline{MEMW}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers; the Am9517A operates as in Read or Write transfers generating addresses, responding to $\overline{EOP}$, etc. However, the memory and I/O control lines remain inactive.

Memory-to-Memory: The Am9517A includes a block move capability that allows blocks of data to be moved from one memory address space to another. When Bit C0 in the Command register is set to a logical 1, channels 0 and 1 will operate as memory-to-memory transfer channels. Channel 0 forms the source address and channel 1 forms the destination address. The channel 1 word count is used. A memory-to-memory transfer is initiated by setting a software DMA request for channel 0. Block Transfer Mode should be used for memory-to-memory. When channel 0 is programmed for a fixed source address, a single source word may be written into a block of memory.

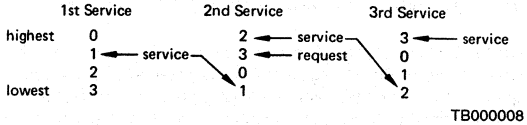
When setting up the Am9517A for memory-to-memory operation, it is suggested that both channels 0 and 1 be masked out. Further, the channel 0 word count should be initialized to the same value used in channel 1. No DACK outputs will be active during memory-to-memory transfers.

The Am9517A will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers may be found in Timing Diagram 2.

Autoinitialize: By programming a bit in the Mode register, a channel may be set up for an Autoinitialize operation. During Autoinitialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set by $\overline{EOP}$ when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to repeat its service without CPU intervention.

Priority: The Am9517A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly. With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.



The priority encoder selects the highest priority channel requesting service on each active-going HACK edge. Once a channel is started, its operation will not be suspended if a request is received by a higher priority channel. The high priority channel will only gain control after the lower priority channel releases HREQ. When control is passed from one channel to another, the CPU will always gain bus control. This ensures generation of rising HACK edge to be used to initiate selection of the new highest-priority requesting channel.

Compressed Timing: To achieve even greater throughput where system characteristics permit, the Am9517A can compress the transfer time to two clock cycles. From Timing Diagram 3 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3 the read pulse width is made equal to the write pulse width, and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur when A8-A15 need updating (see Address Generation). Timing for compressed transfers is found in Timing Diagram 4.

Extended Write: For Flyby Transactions late write is normally used, as this allows sufficient time for the $\overline{IOR}$ signal to get data from the peripheral onto the bus before $\overline{MEMW}$ is activated. In some systems, performance can be improved by starting the write cycle earlier. This is especially true for memory-to-memory transactions.

Address Generation: To reduce pin count, the Am9517A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a 3-state enable. The lower order address bits are output by the Am9517A directly. Lines A0-A7 should be connected to the address bus. Timing Diagram 1 shows the time relationships between CLK, AEN, ADSTB, DB0-DB7 and A0-A7.

During Block and Demand Transfer mode services which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the Am9517A executes S1 states only when updating of A8-A15 in the latch is necessary. This means for long services that S1 states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

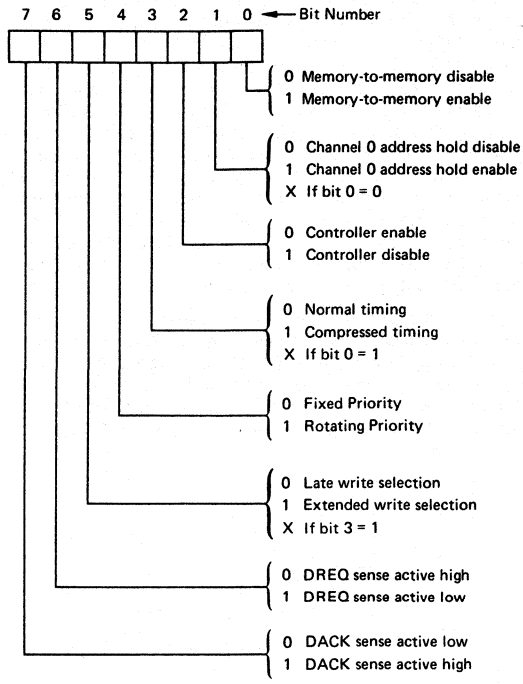
Register Description

Current Address Register: Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialization takes place only after an $\overline{EOP}$.

Current Word Count Register: Each channel has a 16-bit Current Word Count register. This register should be programmed with, and will return on a CPU read, a value one less than the number of words to be transferred. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service, it may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. Note that the contents of the Word Count register will be FFFF (hex) following on internally generated $\overline{EOP}$.

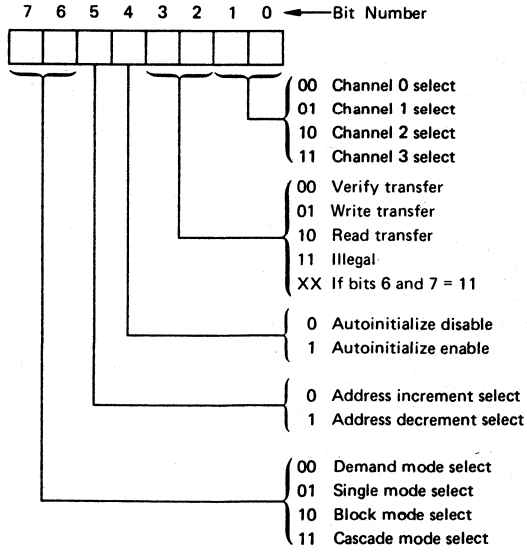
Base Address and Base Word Count Registers: Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original values of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes during DMA, programming by the microprocessor. Accordingly, writing to these registers when intermediate values are in the Current registers will overwrite the intermediate values. The Base registers cannot be read by the microprocessor.

Command Register: This 8-bit register controls the operation of the Am9517A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. The following table lists the function of the command bits. See Figure 2 for address coding.



DF000970

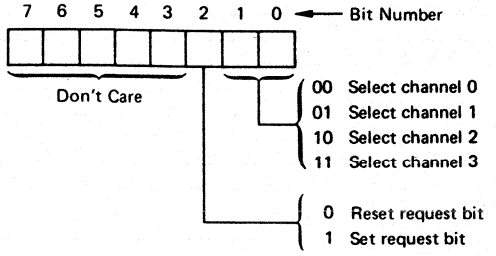
Mode Register: Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written to.



DF000980

Request Register: The Am9517A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are nonmaskable and subject to prioritization by the Priority Encoder network. Each register bit

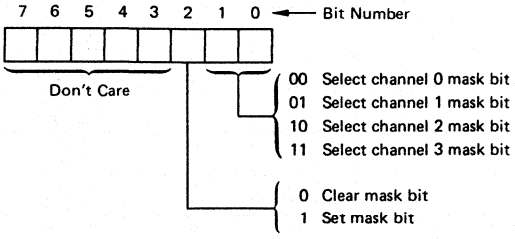
is set or reset separately under software control or is cleared upon generation of a TC or external EOP. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 2 for address coding.



DF000990

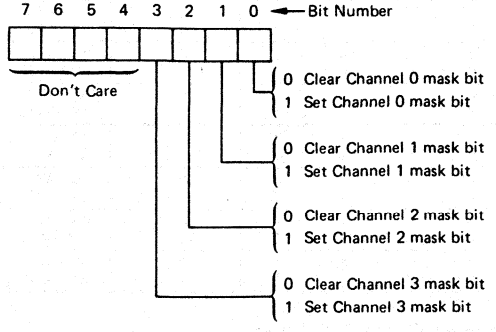
Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

Mask Register: Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an EOP if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 2 for instruction addressing.



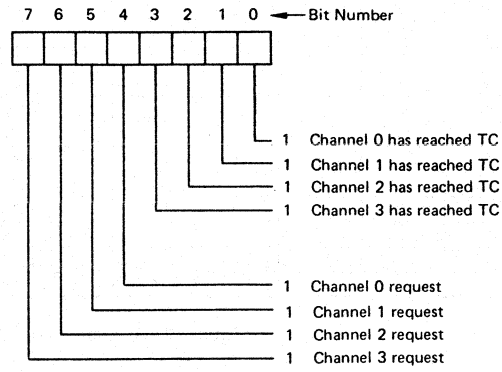
DF001000

All four bits of the Mask Register may also be written with a single command.



DF001010

Status Register: The Status registers may be read out of the Am9517A by the microprocessor. It indicates which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set each time a TC is reached by that channel, including after each Autoinitialization. These bits are cleared by Reset and each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service.



DF001020

Temporary Register: The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands: There are two special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The two software commands are:

Clear First/Last Flip/Flop: This command may be issued prior to writing or reading Am9517A address or word count information. This initializes the Flip/Flop to a known state so that subsequent accesses to register contents by the microprocessor will address lower and upper bytes in the correct sequence. When the Flip/Flop is cleared it addresses the lower byte and when set it addresses the upper byte.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary and Internal First/Last Flip/Flop registers are cleared and the Mask register is set. The Am9517A will enter the Idle cycle.

Figure 2 lists the address codes for the software commands.

Interface Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 2. Register and Function Addressing

Channel	Register	Operation	Signals							Internal Flip/Flop	Data Bus DB0 – DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base & Current Address	Write	0 0	1 1	0 0	0 0	0 0	0 0	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	0 0	0 0	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	0 0	0 0	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	0 0	0 0	1 1	0 1	W0 – W7 W8 – W15
1	Base & Current Address	Write	0 0	1 1	0 0	0 0	0 0	1 1	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	0 0	1 1	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	0 0	1 1	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	0 0	1 1	1 1	0 1	W0 – W7 W8 – W15
2	Base & Current Address	Write	0 0	1 1	0 0	0 0	1 1	0 0	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	1 1	0 0	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	1 1	0 0	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	1 1	0 0	1 1	0 1	W0 – W7 W8 – W15
3	Base & Current Address	Write	0 0	1 1	0 0	0 0	1 1	1 1	0 0	0 1	A0 – A7 A8 – A15
	Current Address	Read	0 0	0 0	1 1	0 0	1 1	1 1	0 0	0 1	A0 – A7 A8 – A15
	Base & Current Word Count	Write	0 0	1 1	0 0	0 0	1 1	1 1	1 1	0 1	W0 – W7 W8 – W15
	Current Word Count	Read	0 0	0 0	1 1	0 0	1 1	1 1	1 1	0 1	W0 – W7 W8 – W15

Figure 3. Word Count and Address Register Command Codes

APPLICATIONS INFORMATION

Figure 4 shows a convenient method for configuring a DMA system with the Am9517A Controller and a microprocessor system. The Multimode DMA Controller issues a Hold Request to the processor whenever there is at least one valid DMA Request from a peripheral device. When the processor replies with a Hold Acknowledge signal, the Am9517A takes control of the Address Bus, the Data Bus and the Control Bus. The address for the first transfer operation comes out in two bytes – the least significant eight bits on the eight Address outputs and the most significant eight bits on the Data Bus. The contents of the Data Bus are then latched into the Am74LS373 register to complete the full 16 bits of the Address Bus. The Am74LS373 is a high-speed, low power, 8-bit, 3-state register in a 20-pin package. After the initial transfer takes place, the register is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one Am9517A is used.

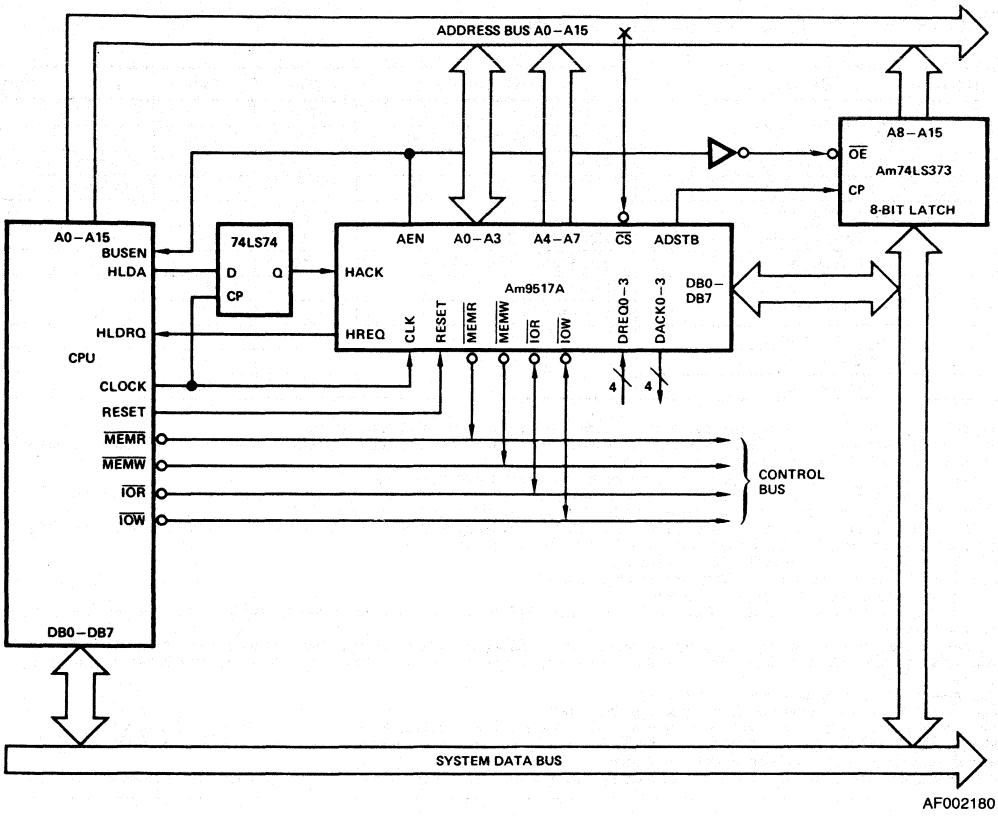


Figure 4. Basic DMA Configuration

ABSOLUTE MAXIMUM RATINGS

Storage temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages with Respect to V_{SS} .. −0.5V to +7.0V
Power Dissipation (Package Limitation) 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	T_A	V_{CC}
Commercial	0°C to +70°C	5.0V ±5%
Industrial	−40°C to 85°C	5.0V ±10%

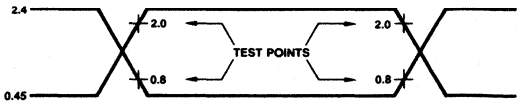
Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
VOH	Output HIGH Voltage	IOH = −200μA	2.4			Volts
		IOH = −100μA, (HREQ Only)	3.3			
VOL	Output LOW Voltage	IOL = 3.2mA			0.45	Volts
VIH	Input HIGH Voltage		2.0		VCC+0.5	Volts
VIL	Input LOW Voltage		−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VI ≤ VCC	−10		+10	μA
IOZ	Output Leakage Current	VCC ≤ VO ≤ VSS + 40	−10		+10	μA
ICC	VCC Supply Current	T_A = +25°C		65	130	mA
		T_A = 0°C		75	150	
CO	Output Capacitance	fc = 1.0MHz, Inputs = 0V		4	8	pF
CI	Input Capacitance			8	15	pF
CIO	I/O Capacitance			10	18	pF

Notes:
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.
2. Input timing parameters assume transition times of 20ns or less. Waveform measurement points for both input and output signals are 2.0V for HIGH and 0.8V for LOW, unless otherwise noted.
3. Output loading is 1 Standard TTL gate plus 50pF capacitance unless noted otherwise.
4. The new $\overline{IOW}$ or $\overline{MEMW}$ pulse width for normal write will be TCY-100ns and for extended write will be 2TCY-100ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for normal read will be 2TCY-50ns and for compressed read will be TCY-50ns.
5. TDQ is specified for two different output HIGH levels. TDQ1 is measured at 2.0V. TDQ2 is measured at 3.3V. The value for TDQ2 assumes an external 3.3kΩ pull-up resistor connected from HREQ to VCC.
6. DREQ should be held active until DACK is returned.
7. DREQ and DACK signals may be active high or active low.
   Timing diagrams assume the active high mode.
8. Output loading on the data bus is 1 Standard TTL gate plus 15pF for the minimum value and 1 Standard TTL gate plus 100pF for the maximum value.
9. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600ns for the Am9517A-1, at least 450ns for the Am9517A-4, and 400ns for the Am9517A-5 as recovery time between active read or write pulses.
10. Parameters are listed in alphabetical order.
11. Pin 5 is an input that should always be at a logic high level.
    An internal pull-up resistor will establish a logic high when the pin is left floating. Alternatively, pin 5 may be tied to VCC.
12. Signals $\overline{READ}$ and $\overline{WRITE}$ refer to $\overline{IOR}$ and $\overline{MEMW}$ respectively for peripheral-to-memory DMA operations and to $\overline{MEMR}$ and $\overline{IOW}$ respectively for memory-to-peripheral DMA operations.
13. If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (TCY).

SWITCHING TEST INPUT WAVEFORM



WF003310

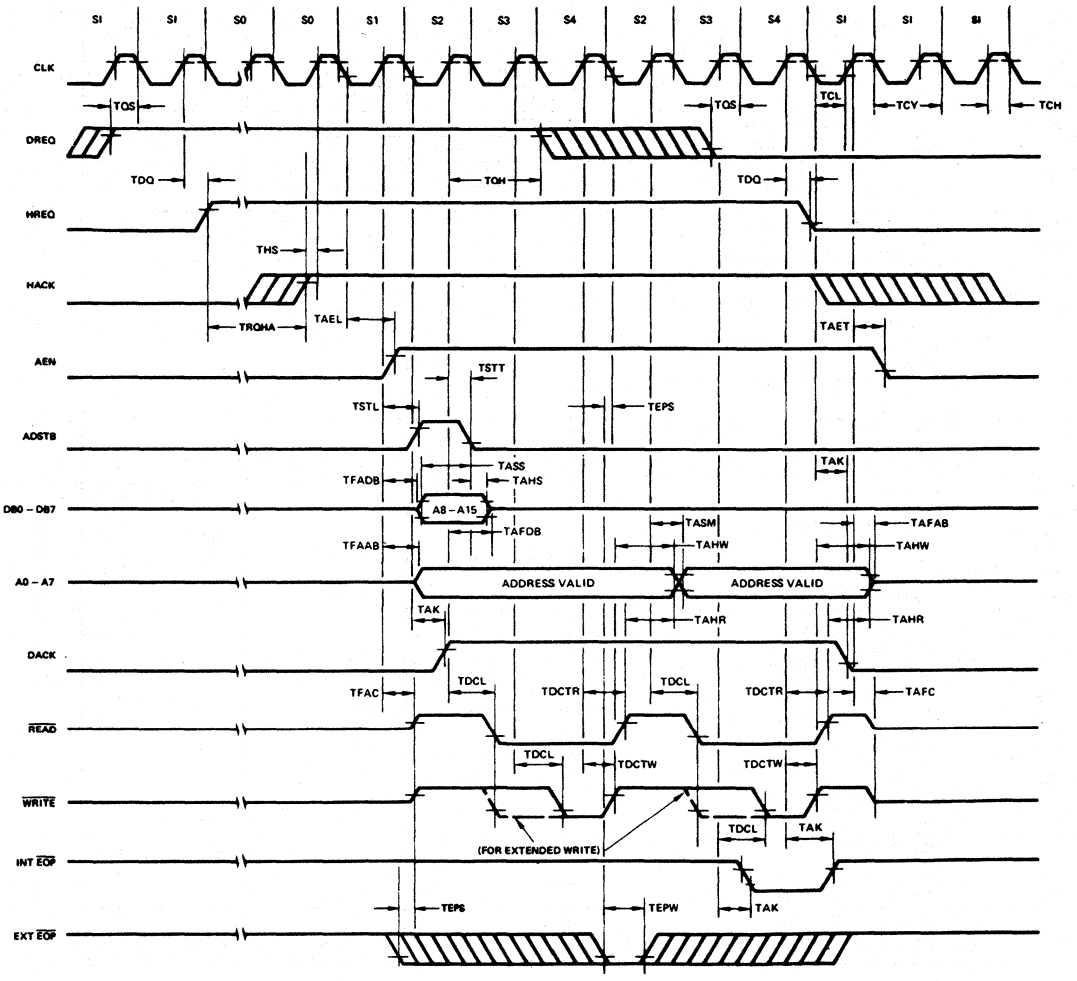
SWITCHING CHARACTERISTICS over operating range unless otherwise specified
ACTIVE CYCLE (Notes 2, 3, 10, 11, and 12 under DC Characteristics)

Parameters	Description	Am9517A-1		Am9517A-4		Am9517A-5		Units
		Min	Max	Min	Max	Min	Max	
TAEL	AEN HIGH from CLK LOW (S1) Delay Time		300		225		200	ns
TAET	AEN LOW from CLK HIGH (S1) Delay Time		200		150		130	ns
TAFAB	ADR Active to Float Delay from CLK HIGH		150		120		90	ns
TAFC	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		150		120		120	ns
TAFDB	DB Active to Float Delay from CLK HIGH		250		190		170	ns
TAHR	ADR from $\overline{READ}$ HIGH Hold Time	TCY-100		TCY-100		TCY-100		ns
TAHS	DB from ADSTB LOW Hold Time	50		40		30		ns
TAHW	ADR from WRITE HIGH Hold Time	TCY-50		TCY-50		TCY-50		ns
TAK	DACK Valid from CLK LOW Delay Time		280		220		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time		250		190		170	ns
	$\overline{EOP}$ LOW to CLK HIGH Delay Time		250		190		170	ns
TASM	ADR Stable from CLK HIGH		250		190		170	ns
TASS	DB to ADSTB LOW Set-up Time	100		100		100		ns
TCH	Clock High Time (Transitions $\leqslant$ 10ns)	120		100		80		ns
TCL	Clock Low Time (Transitions $\leqslant$ 10ns)	150		110		68		ns
TCY	CLK Cycle Time	320		250		200		ns
TDCL	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 4)		270		200		190	ns
TDCTR	$\overline{Read}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		270		210		190	ns
TDCTW	$\overline{WRITE}$ HIGH from CLK HIGH (S4) Delay Time (Note 4)		200		150		130	ns
TDQ1	HREQ Valid from CLK HIGH Delay Time (Note 5)		160		120		120	ns
TDQ2			250		190		120	ns
TEPS	$\overline{EOP}$ LOW from CLK LOW Set-up Time	60		45		40		ns
TEPW	$\overline{EOP}$ Pulse Width	300		225		220		ns
TFAAB	ADR Float to Active Delay from CLK HIGH		250		190		170	ns
TFAC	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		200		150		150	ns
TFADB	DB Float to Active Delay from CLK HIGH		300		225		200	ns
THS	HACK Valid to CLK HIGH Set-up Time	100		75		75		ns
TIDH	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		0		0		ns
TIDS	Input Data to $\overline{MEMR}$ HIGH Set-up Time	250		190		170		ns
TODH	Output Data from $\overline{MEMW}$ HIGH Hold Time	20		20		10		ns
TODV	Output Data Valid to $\overline{MEMW}$ HIGH (Note 13)	200		125		125		ns
TQS	DREQ to CLK LOW (S1, S4) Set-up Time	0		0		0		ns
TRH	CLK to READY LOW Hold Time	20		20		20		ns
TRS	READY to CLK LOW Set-up Time	100		60		60		ns
TSTL	ADSTB HIGH from CLK HIGH Delay Time		200		150		130	ns
TSTT	ADSTB LOW from CLK HIGH Delay Time		140		110		90	ns
TQH	DREQ from DACK Valid Hold Time	0		0		0		ns
TRQHA	HREQ to HACK Delay Time	1		1		1		clk

Note: 14. Because $\overline{EOP}$ high from clock high is load dependent, users wishing to test these parameters should use a 2k pull-up resistor and a tester with 50pf or less load capacitance. Time constant RC = 120ns is added to the specified number in the data sheet for testing.

SWITCHING WAVEFORMS

Timing Diagram 1. Active Cycle Timing Diagram

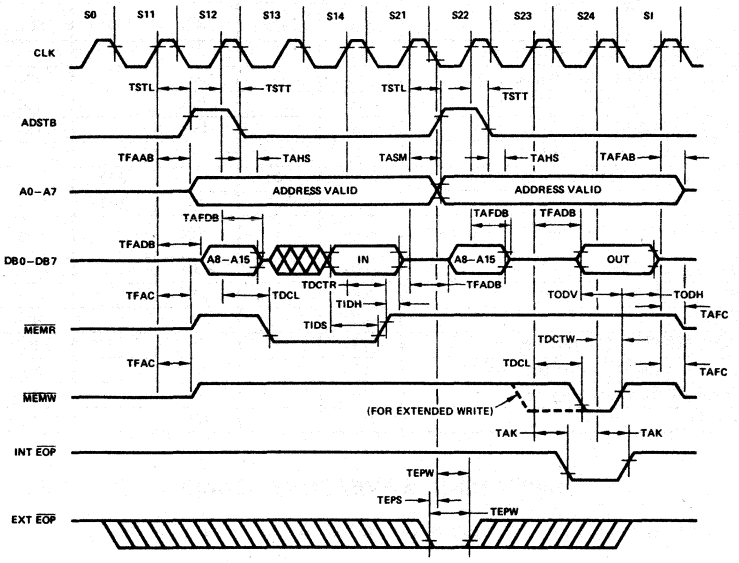


WF003300

Note: $\overline{EOP}$ must precede AEN in single transfer mode.

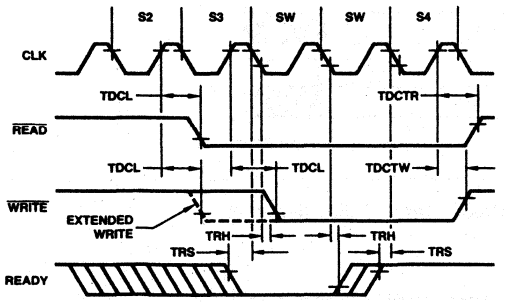
SWITCHING WAVEFORMS (Cont.)

Timing Diagram 2. Memory-to-Memory



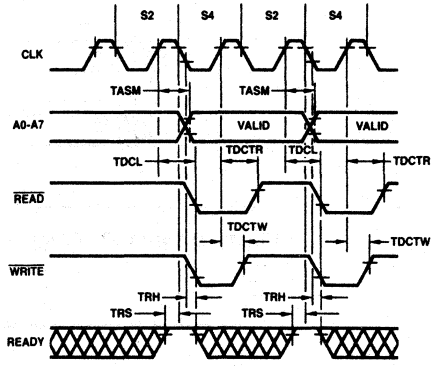
WF003320

Timing Diagram 3. Ready Timing



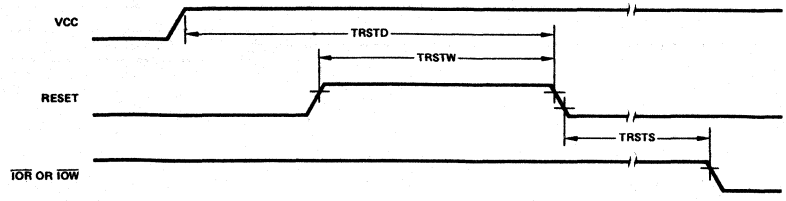
WF003330

Timing Diagram 4. Compressed Timing



WF003340

Timing Diagram 5. Reset Timing



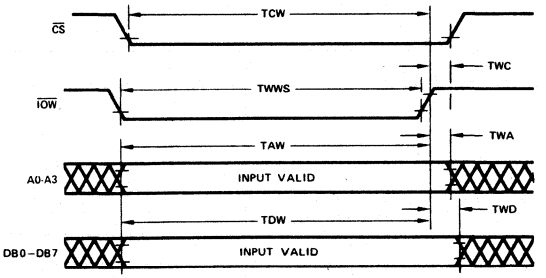
WF003350

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
Program Condition (Idle Cycle) (Notes 2, 3, 10, and 11 under DC Characteristics)

Parameters	Description	Am9517A-1		Am9517A-4		Am9517A-5		Units
		Min	Max	Min	Max	Min	Max	
TAR	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		50		ns
TAW	ADR Valid to $\overline{WRITE}$ HIGH Set-up Time	200		150		130		ns
TCW	CS LOW to $\overline{WRITE}$ HIGH Set-up Time	200		150		130		ns
TDW	Data Valid to $\overline{WRITE}$ HIGH Set-up Time	200		150		130		ns
TRA	ADR or CS Hold from $\overline{READ}$ HIGH	0		0		0		ns
TRDE	Data Access from $\overline{READ}$ LOW (Note 8)		200		200		140	ns
TRDF	DB Float Delay from $\overline{READ}$ HIGH	20	100	20	100	0	70	ns
TRSTD	Power Supply HIGH to RESET LOW Set-up Time	500		500		500		μs
TRSTS	RESET to First $\overline{IOWR}$	2TCY		2TCY		2TCY		ns
TRSTW	RESET Pulse Width	300		300		300		ns
TRW	$\overline{READ}$ Width	300		250		200		ns
TWA	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		20		ns
TWC	CS HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		20		ns
TWD	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		30		ns
TWWS	Write Width	200		200		160		ns

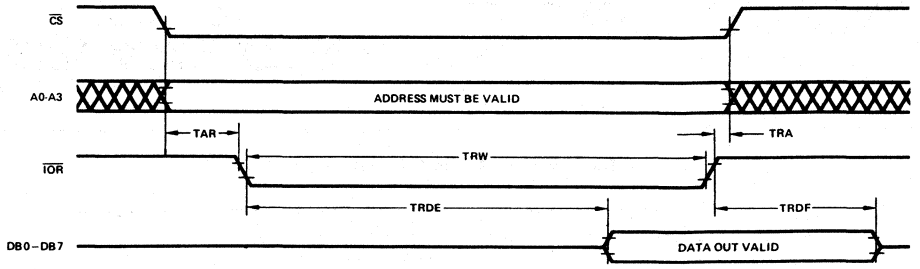
SWITCHING WAVEFORMS (Cont.)

Timing Diagram 6. Program Condition Write Timing (Note 9)



WF003360

Timing Diagram 7. Program Condition Read Cycle (Note 9)



WF003370

Am9518/AmZ8068

Data Ciphering Processor

DISTINCTIVE CHARACTERISTICS

- **Encrypts and decrypts data**
  Implements National Bureau of Standards standard data encryption algorithm.
- **High-Speed Operation**
  Am9518 and AmZ8068 throughput over 1.3 and 1.7M bytes per second, respectively. Operates at data rates fast enough for disk controllers, high-speed DMA, telecommunication channels.
- **Supports three ciphering options**
  Electronic Code Book for disk applications. Chain Block Cipher for high-speed telecommunications. Cipher Feedback for low-to-medium speed, byte-oriented communications.
- **Three separate key registers on-chip**
  Separate registers for encryption key, decryption key and master key improve system security and throughput by eliminating need to reload keys frequently.
- **Three separate data ports provide flexible interface, improved security**
  The DCP utilizes a master port, slave port and key port. Functions of the three ports can be programmed by the user to provide for simple interface to AmZ8000 and Am2900 systems and to provide total hardware separation of encrypted data, clear data and keys.

GENERAL DESCRIPTION

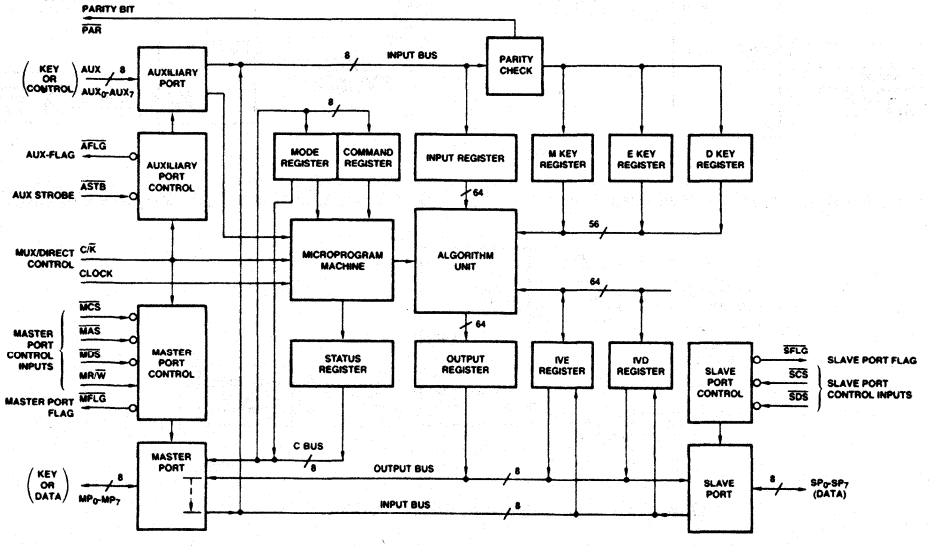
The Am9518/AmZ8068 Data Ciphering Processor is an N-channel silicon gate LSI product containing the circuitry necessary to encrypt and decrypt data using the National Bureau of Standards encryption algorithm. It is designed to be used in a variety of environments, including dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems.

The DCP provides a high throughput rate using Cipher Feedback, Electronic Code Book or Cipher Block Chain operating modes. Separate ports for key input, clear data and enciphered data enhance security.

The system communicates with the DCP using commands entered in the master port and through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities are all performed concurrently. External DMA control can easily be used to enhance throughput in some system configurations.

This device is designed to interface directly to the AmZ8000 CPU bus and, with a minimum of external logic, to the 2900, MC68000, 8086, 8085, and 8051 families of processors.

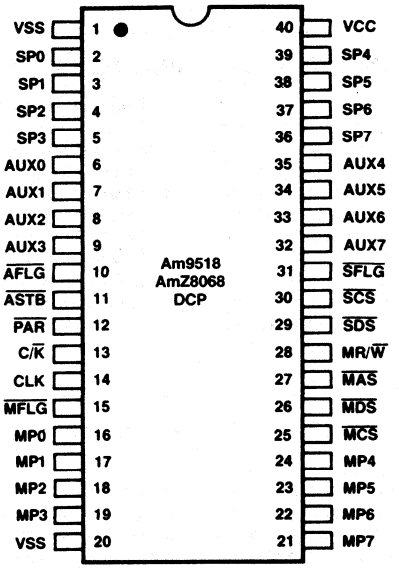
BLOCK DIAGRAM



00618B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40

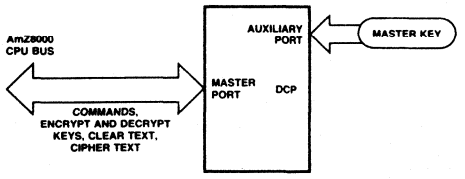


CD005111

Note: Pin 1 is marked for orientation

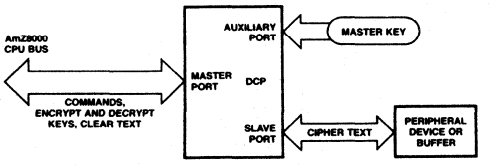
Also available in PLCC. See Section 7 for pinout details.

DCP DATA FLOW OPTIONS



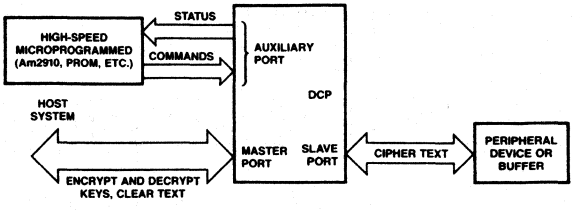
AF002220

Single-Port Configuration, Multiplexed Control



AF002230

Dual-Port Configuration, Multiplexed Control

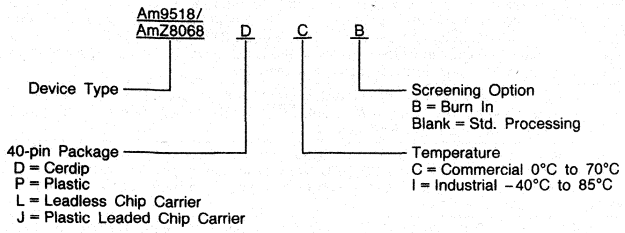


AF002240

Dual-Port Configuration, Direct Control

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9518/ AmZ8068	DC, DCB, DI, DIB, LC, LCB, LI, LIB, PC, PCB, PI, PIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

Export of this device from the United States is subject to control by the U.S. Department of State.

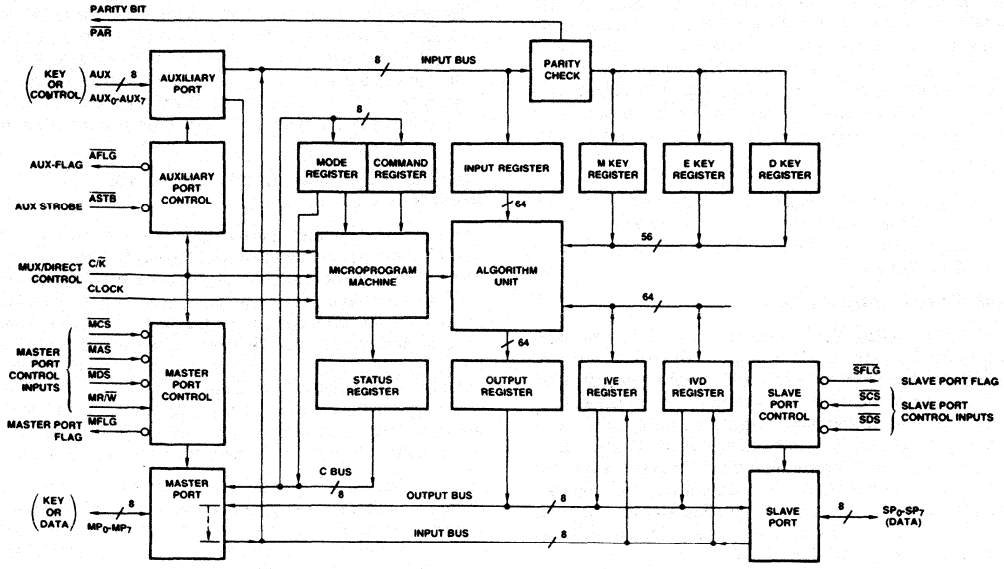
PIN DESCRIPTION

Pin No.	Name	I/O	Description
40	V_{CC}		+5V Power Supply.
1, 20	V_{SS}		Ground (2 pins).
14	CLK	I	(Clock, TTL levels). An external timing source is input via the CLK pin. The Master and Slave Port Data Strobe signals, $\overline{MDS}$, $\overline{SDS}$ and also AUX_5-S/$\overline{S}$ in Direct Control Mode (C/$\overline{K}$ HIGH) must change synchronously with this clock input. In addition, the Auxiliary, Master and Slave Port Flag outputs ($\overline{AFLG}$, $\overline{MFLG}$ and $\overline{SFLG}$) will change synchronously with the clock. When using the DCP with the AmZ8000 in multiplexed control mode, the clock input must agree in frequency and phase with the processor clock; however, the DCP does not require the high voltage levels of the processor clock.
13	C/$\overline{K}$	I	(Control/Key Mode Control). This input is the primary control over the operating characteristics of the DCP. A LOW input on C/$\overline{K}$ places the DCP into Multiplexed Control Mode, enabling programmed access to internal registers through the Master Port and enabling input of keys tnrough the Auxiliary Port. A HIGH input on C/$\overline{K}$ specifies operation in Direct Control Mode, wherein several of the Auxiliary Port pins become direct control/status signals which can be driven/sensed by high-speed controller logic (such as the Am29116 or Am2901/Am2903-based processors), and access to internal registers through the Master Port is limited to the Input or Output Register.
16-19, 21-24	MP_0-MP_7	I/O	(Master Port Bus). These eight bidirectional lines are used to specify internal register addresses in Multiplexed Control Mode (see C/$\overline{K}$) and to input and output data. The Master Port provides software access to the Status, Command and Mode Registers, as well as the Input and Output Registers. The three-state Master Port outputs will be enabled only when the Master Port is selected by Master Port Chip Select ($\overline{MCS}$) LOW, with Master Port Read/Write (MR/$\overline{W}$) HIGH, and strobed by Master Port Data Strobe ($\overline{MDS}$) LOW. MP_0 is the low-order bit. Data and key information is entered into this port with the most significant byte first.
25	$\overline{MCS}$	I	(Master Port Chip Select). This active LOW input signal is used to select the Master Port. In Multiplexed Control Mode (C/$\overline{K}$ low), the level on $\overline{MCS}$ is latched internally on the rising edge of Master Port Address Strobe ($\overline{MAS}$). This latched level is retained as long as $\overline{MAS}$ is HIGH; when $\overline{MAS}$ is LOW, the latch becomes invisible and the internal signal will follow the $\overline{MCS}$ input. In Direct Control Mode (C/$\overline{K}$ HIGH), no latching of Master Port Chip Select occurs; the level on $\overline{MCS}$ is passed directly to the internal select circuitry irrespective of state of Address Strobe ($\overline{MAS}$).
27	$\overline{MAS}$	I	(Master Port Address Strobe). In Multiplexed Control Mode (C/$\overline{K}$ low), an active LOW signal on this pin indicates the presence of valid address and chip select information at the Master Port. This information will be latched internally on the rising edge of Address Strobe. When C/$\overline{K}$ is HIGH (Direct Control Mode), $\overline{MAS}$ may be HIGH or LOW without affecting DCP operation, except that, irregardless of C/$\overline{K}$ state, if both Master Port Address Strobe ($\overline{MAS}$) and Data Strobe ($\overline{MDS}$) are LOW simultaneously, the DCP will be reset to ECB mode and all flags inactive.
26	$\overline{MDS}$	I	(Master Port Data Strobe). This active LOW input is used in coincidence with a valid Master Port Chip Select ($\overline{MCS}$) to indicate that valid data is present on MP_0-MP_7 for an input operation or that data is to be placed on MP_0-MP_7 during output. Master Port Data Strobe and Address Strobe ($\overline{MAS}$) are normally mutually exclusive; if both go LOW simultaneously, the DCP is reset to ECB mode and all flags inactive.
28	MR/$\overline{W}$	I	(Master Port Read/Write). This input signal indicates to the DCP whether the current Master Port operation is a read (HIGH) or a write (LOW), thereby indicating that data is to be transferred from or to an internal register, respectively. MR/$\overline{W}$ is not latched internally and must be held stable while Master Port Data Strobe ($\overline{MDS}$) is LOW.
15	$\overline{MFLG}$	O	(Master Port Flag). This active LOW flag is used to indicate the need for a data transfer into or out of the Master Port during normal ciphering operation. Depending upon control bits written to the Mode Register (see Register Description), the Master Port will be associated with either the Input Register or the Output Register. If data is to be transferred through the Master Port to the Input Register, the $\overline{MFLG}$ reflects the contents of the Input Register; after any Start command is entered, $\overline{MFLG}$ will go active (LOW) whenever the Input Register is not full. $\overline{MFLG}$ is forced HIGH by any command other than a Start. Conversely, if the Master Port is associated with the Output Register, $\overline{MFLG}$ reflects the contents of the Output Register (except in Single Port configuration; see Detailed Description). $\overline{MFLG}$ will go active (LOW) whenever the Output Register is not empty. In Single Port Configuration, the Master Port Flag reflects the contents of the Input Register, while the Slave Port Flag ($\overline{SFLG}$) is associated with the Output Register.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
2-5, 30-36	SP_0-SP_7	I/O	(Slave Port Bus, Bidirectional). The Slave Port provides a second data input/output interface to the DCP, allowing overlapped input, output and ciphering operations. The tri-state Slave Port outputs will be driven only when Slave Port Chip Select ($\overline{SCS}$) and Slave Port Data Strobe ($\overline{SDS}$) are both LOW and $\overline{SFLG}$ = 0, and the internal Port Control Configuration allows output to the Slave Port. SP_0 is the LOW order bit. Data entered or retrieved through this port is most significant byte first.
30	$\overline{SCS}$	I	(Slave Port Chip Select). This active LOW signal is logically combined with Slave Port Data Strobe ($\overline{SDS}$) to facilitate Slave Port data transfers in a bus environment. $\overline{SCS}$ is not latched internally and may be tied permanently LOW without impairing Slave Port operation.
29	$\overline{SDS}$	I	(Slave Port Data Strobe). This active LOW input, in coincidence with Slave Port Chip Select ($\overline{SCS}$) LOW, indicates to the DCP that valid data is on the SP_0-SP_7 lines for an input operation, or that data is to be driven onto the SP_0-SP_7 lines for output. The direction of data flow is determined by control bits in the Mode Register (see Register Description).
31	$\overline{SFLG}$	O	(Slave Port Flag). This active LOW output indicates the state of either the Input Register or the Output Register, depending on control bits in the Mode Register. In Single Port Configuration, $\overline{SFLG}$ will go active whenever the Output Register is not empty during normal processing. In Dual Port Configuration, $\overline{SFLG}$ will reflect the content of whichever register is associated with the Slave Port. If the Input Register is assigned to the Slave Port, $\overline{SFLG}$ will go active whenever the Input Register is not full; once any of the Start commands has been entered, $\overline{SFLG}$ will be forced inactive if any other command is entered. Conversely, if the Slave Port is assigned to the Output Register, $\overline{SFLG}$ will go active whenever the Output Register is not empty.
6-9, 32, 35	AUX_0-AUX_7	I/O	(Auxiliary Port Bus, Bidirectional). When the DCP is operated in Multiplexed Control Mode (C/$\overline{K}$ LOW), these eight lines form a key-byte input port which may be used to enter the Master and Session Keys. In fact, this port is the only path available for entering the Master Key. (Session Keys may alternatively be entered via the Master Port.) AUX_0 is the low-order bit, and is considered to be the parity bit in key bytes. The most significant byte is entered first. When the DCP is operated in Direct Control Mode (C/$\overline{K}$ HIGH), the Auxiliary Port's key-entry function is disabled and five of the eight lines become direct control/status lines for interfacing to high-speed microprogrammed controllers. In this case, AUX_0, AUX_1 and AUX_4 have no function, and the other pins are defined as follows below.
34	AUX_5-S/$\overline{S}$	I	(Start/$\overline{Stop}$). When this pin goes LOW (Stop), the DCP will follow the sequence that would normally occur were a Stop command to be entered. Conversely, when this pin goes HIGH, a sequence equivalent to a Start Encryption or Start Decryption command will be followed. At the time AUX_5-S/$\overline{S}$ goes HIGH, the level on AUX_6-E/$\overline{D}$ (see below) selects either the Start Encryption or Start Decryption interpretation.
32	AUX_7-K/$\overline{D}$	I	(Key/$\overline{Data}$). When this signal goes HIGH, the DCP initiates a key-data input sequence as if a Load Clear E or D Key through Master Port command had been entered. The level on AUX_6-E/$\overline{D}$ will determine whether the subsequently entered clear-key bytes are written into the E Key Register (E/$\overline{D}$ HIGH) or the D Key Register (E/$\overline{D}$ LOW). AUX_7-K/$\overline{D}$ and AUX_5-S/$\overline{S}$ are mutually exclusive control lines; when one goes active (HIGH), the other must be and remain inactive (LOW) until the first returns to an inactive state. In addition, both lines must be inactive (LOW) whenever a transition occurs on C/$\overline{K}$ (entering or exiting Direct Control Mode).
33	AUX_6-E/$\overline{D}$	I	(Encrypt/$\overline{Decrypt}$). When AUX_5-S/$\overline{S}$ goes HIGH, initiating a normal data ciphering operation, this input specifies whether the ciphering algorithm is to encrypt (E/$\overline{D}$ HIGH) or decrypt (LOW). When AUX_7-K/$\overline{D}$ goes HIGH, initiating entry of key bytes, the level on AUX_6-E/$\overline{D}$ specifies whether the bytes are to be written into the E Key Register (E/$\overline{D}$ HIGH) or the D Key Register (E/$\overline{D}$ LOW). The AUX_6-E/$\overline{D}$ input is not latched internally, and must be held constant whenever one or more of AUX_5-S/$\overline{S}$, AUX_7-K/$\overline{D}$, AUX_2-$\overline{BSY}$, or AUX_3-$\overline{CP}$ are active. Failure to maintain the proper level on AUX_6-E/$\overline{D}$ during loading or ciphering operations will result in scrambled data in the internal registers.
8	AUX_2-$\overline{BSY}$	O	(Busy). This active LOW status output gives a hardware indication that the ciphering algorithm is in operation. AUX_2-$\overline{BSY}$ is driven by the BSY bit in the Status Register (see Register Description), such that when the BSY bit is "1" (active), AUX_2-$\overline{BSY}$ is LOW.
9	AUX_3-$\overline{CP}$	O	(Command Pending). This active LOW status output gives a hardware indication that the DCP is ready to accept input of key bytes following a LOW-to-HIGH transition on AUX_7-K/$\overline{D}$. AUX_3-$\overline{CP}$ is driven by the CP bit in the Status Register, such that when the CP bit is "1" (active), AUX_3-$\overline{CP}$ is LOW.
11	$\overline{ASTB}$	I	(Auxiliary Port Strobe). The rising (trailing) edge of $\overline{ASTB}$ strobes the key data on pins AUX_0-AUX_7 into the appropriate internal key register in Multiplexed Control Mode (C/$\overline{K}$ LOW). This input is ignored unless $\overline{AFLG}$ and C/$\overline{K}$ are both LOW. One byte of key data is entered on each $\overline{ASTB}$, the most significant byte first.
10	$\overline{AFLG}$	O	(Auxiliary Port Flag). This active LOW output signal indicates that the DCP is expecting key data to be entered on pins AUX_0-AUX_7. This can occur only when C/$\overline{K}$ is LOW and a Load Key Through AUX Port command has been entered. $\overline{AFLG}$ will remain active (LOW) during input of all eight bytes, and will go inactive with the leading edge of the eighth strobe ($\overline{ASTB}$).
12	$\overline{PAR}$	O	(Parity). The DCP checks all key bytes for correct (odd) parity as they are entered through either the Master Port (Multiplexed or Direct Control Mode) or the Auxiliary Port (Multiplexed Control Mode only). If any key byte contains even parity, the PAR bit in the Status Register is set to "1," and $\overline{PAR}$ goes LOW. (See Parity Checking of Keys.) Least significant bit of key data is the parity.

BLOCK DIAGRAM



BD003290

DETAILED DESCRIPTION

The overall design of the DCP, as shown in the block diagram, is optimized for high data throughput. Data bytes can be transferred through both the Master and Slave ports, and key bytes can be written through both the Auxiliary and Master ports. Three 8-bit buses, Input, Output and C Bus, carry data and key bytes between the ports and the internal registers. Three 56-bit, write-only key registers are provided for the Master (M) Key, the Encryption (E) Key and the Decryption (D) Key. Parity checking is provided on incoming key bytes. Two 64-bit registers are provided for Initializing Vectors (IVE and IVD) required for chained (feedback) ciphering modes. Three 8-bit registers (Mode, Command and Status) are accessible through the Master Port for interfacing to a host microprocessor, such as the AmZ8000.

Algorithm Processing

The DCP's algorithm processing unit (see the block diagram) is designed to encrypt and decrypt data according to the National Bureau of Standards Data Encryption Standard (DES), as specified in Federal Information Processing Standards Publication 46.

The DES specifies a method for encrypting 64-bit blocks of clear data ("plain text") into corresponding 64-bit blocks of "cipher text." The DCP offers three ciphering methods selected by the Cipher Type field of the Mode Register: Electronic Code Book (ECB), Cipher Block Chain (CBC) and Cipher Feedback (CFB). These methods are implemented in accordance with Federal Information Processing Standards Publication 46. Electronic Code Book (ECB) is a straightforward implementation of the DES: 64 bits of clear data in, 64 bits of cipher text out, with no cryptographic dependence between blocks. Cipher Block Chain (CBC) also operates on blocks of 64 bits, but includes a feedback step which chains consecutive blocks so that repetitive data in the plain text (such as ASCII blanks) does not yield repetitive cipher text; CBC also provides an error extension characteristic valuable in protecting against fraudulent data insertions and deletions.

Cipher Feedback (CFB) is an additive stream cipher method in which the DES generates a pseudorandom binary stream which is then exclusive-OR'd with the clear data to form the cipher text. The cipher text is then fed back to form a portion of the next DES input block. The DCP implements 8-bit cipher feedback, with data input, output, and feedback paths being one byte wide. This method is useful for low speed, character-at-a-time serial communications.

Multiple Key Registers

The DCP provides the necessary registers to implement a multiple-key or Master-Key system. In such an arrangement, a single Master Key, stored in the DCP M Key Register, is used only to encrypt session keys for transmission to remote DES equipment and to decrypt session keys received from such equipment. The M Key Register may be loaded (with plain text) only through the Auxiliary Port, using the Load Clear Master Key command. (See Commands.)

In addition to the M Key Register, the DCP contains two session key registers: the E Key Register, used to encrypt clear text, and the D Key Register, used to decrypt cipher text.

All three registers are loaded by writing commands like Load Clear E Key through Master Port into the Command Register, and then writing the eight bytes of key data to the port when the Command Pending = "1" in the Status Register. (See Commands.)

Operating Modes: Multiplexed Control vs. Direct Control

The DCP can be operated in either of two basic interfacing modes determined by the logic level on the C/$\overline{K}$ input pin. In Multiplexed Control Mode (C/$\overline{K}$ LOW), the DCP is internally configured to allow a host CPU to directly address five of the internal control/status/data registers and thereby control the

device via mode and command values written to these registers. Also, in Multiplexed Control Mode, the Auxiliary Port is enabled for key-byte input.

If the logic level on C/$\overline{K}$ is brought HIGH, the DCP enters Direct Control Mode, and the Auxiliary Port pins are converted into direct hardware status or control signals that are capable of instructing the DCP to perform a functionally complete subset of its cipher processing at very high throughputs. This operating mode is particularly well-suited for ciphering data for high-speed peripheral devices such as magnetic disk or tape.

Data Flow

Bits M_2, M_3 of the Mode Register control the flow of data into and out of the DCP through the Master and Slave Ports. Three basic configurations are provided: Single Port and two Dual Port configurations.

Single Port Configuration

The simplest configuration occurs when the Mode Register configuration bits are set to Master Port only. Under this operating configuration, the Encrypt/Decrypt bit (M_4) controls the processing of data. Data to be encrypted or decrypted is written to the Master Port Input Register address. To facilitate monitoring of the Input Register status, the $\overline{MFLG}$ signal goes LOW when the Input Register is not full. Data is read by the host CPU through the Master Port Output Register address. Pin $\overline{SFLG}$ goes LOW when the Output Register is not empty. Thus, $\overline{MFLG}$ is redefined as a Master Input Flag and $\overline{SFLG}$ is redefined as a Master Output Flag.

Dual Port, Master Port Clear Configuration

In the dual port configurations, both the Master and Slave Ports are used for data entry and removal. In the Master Port Clear configuration, clear text for encryption can be entered only through the Master Port, and clear text resulting from decryption can be read out only through the Master Port. Cipher text can be handled only through the Slave Port. The actual direction of data flow is controlled either by the Encrypt/Decrypt bit (M_4) in the Mode Register, or by the Start Encryption or Start Decryption commands. If encryption is specified, clear data will flow through the Master Port to the Input Register, and cipher data will be available at the Slave Port when it is ready to be read out of the Output Register. For decryption, the process is reversed, cipher data being written to the Input Register through the Slave Port and Clear data being read from the Output Register through the Master Port.

Dual Port, Slave Port Clear Configuration

This configuration is identical to the previously described Dual Port, Master Port Clear configuration, except that the direction of ciphering is reversed. That is, all data flowing in or out of the Master Port is cipher text, and all data at the Slave Port is clear text.

Master Port Read/Write Timing

The DCP's Master Port is designed to operate directly with the multiplexed address-data bus of the AmZ8000 processor. Several features of the Master Port logic should be stressed.

- The level on Master Port Chip Select ($\overline{MCS}$) is latched internally on the rising (trailing) edge of Master Port Address Strobe ($\overline{MAS}$), thus relieving external address decode circuitry of the responsibility for latching chip select at address time.
- The levels on MP_1, MP_2 are also latched internally on the rising edge of $\overline{MAS}$, and are subsequently decoded to enable reading and writing of the DCP's internal registers (Mode, Command, Status, Input and Output). Again, this eliminates the need for external address latching and decoding.

Data transfers through the Master Port are controlled by the levels and transitions on Master Port Data Strobe ($\overline{MDS}$) and Master Port Read/$\overline{Write}$ (MR/$\overline{W}$), the former controlling the timing and the latter controlling the transfer direction. Note that data transfers do not disturb either the chip-select or address latches, so that once the DCP and a particular register have been selected, any number of reads or writes of that register can be accomplished without intervening address cycles. This feature could greatly speed up loading keys and data, given the necessary transfer control external to the DCP.

Loading Keys and Initializing Vector (IV) Registers

Because the key and initializing vector registers are not directly addressable through any of the DCP's ports, keys and vector data must be loaded (and, in the case of vectors, read out) via "command data sequences" (see Commands). Most of the commands recognized by the DCP are of this type: a Load or Read command is written to the Command Register through the Master Port; the command processor responds by asserting the Command Pending output; the user then either writes eight bytes of key or vector data through the Master or Auxiliary Port, as appropriate to the specific command, or reads eight bytes of vector data from the Master Port.

In Direct Control Mode, only the E Key and D Key registers can be loaded; the M Key and IV Registers are inaccessible. Loading the E and D Key registers is accomplished by asserting the proper state on the AUX_6-E/$\overline{D}$ input (HIGH for E Key, LOW for D Key) and then raising the AUX_7-K/$\overline{D}$ input, indicating that key loading is required. The command processor will attach the proper key register to the Master Port and assert the AUX_3-$\overline{CP}$ (Command Pending) signal (active LOW). The eight key bytes may then be written to the Master Port. In Multiplexed Control Mode, all key and vector registers are writable, and all but the Master (M) Key Register may be loaded with encrypted, as well as clear, data. If the operation is a Load Encrypted command, the subsequent data written to the Master or Auxiliary Port (as appropriate) is routed first to the Input Register and decrypted before being written into the specified key or vector register.

Parity Checking of Keys

Key bytes are considered to contain seven bits of key information and one parity bit. By DES designation, the low-order bit is the parity bit. The parity checking circuit is enabled whenever a byte is written to one of three key registers. The output of the parity detection circuit is connected to pin $\overline{PAR}$, and the state of this pin is reflected in Status Register bit PAR (S_3). Status Register bit PAR goes to "1" whenever a byte with even parity (an even number of "1s") is detected. In addition to the PAR bit, the Status Register has a Latched Parity Bit (LPAR, S_4) which is set to "1" whenever the Status Register PAR bit goes to "1." Once set, the LPAR bit is not cleared until a reset occurs or a new Load Key command is issued.

When an encrypted key is entered, the parity detect logic operates only after the decrypted key is available. The encrypted data is not checked for parity. The $\overline{PAR}$ signal will reflect the state of the decrypted bytes on a byte-to-byte basis, as they are clocked through the parity check logic on their way to the Key Register. Thus, the time $\overline{PAR}$ indicates the status of a byte of decrypted key data may be as short as four clock cycles. The LPAR bit in the Status Register will indicate if any erroneous bytes of data were entered.

Initialization

The DCP can be reset in several ways:
1. By the "Software Reset" command.

2. By a hardware reset, which occurs whenever both $\overline{MAS}$ and $\overline{MDS}$ go LOW simultaneously.
3. By writing to the Mode Register.
4. By aborting any command.

All these sequences are the same internally, except that loading the Mode Register does not subsequently reset the Mode Register.

Once a reset process starts, the DCP is unable to respond to further commands for approximately five clock cycles.

If a power-up hardware reset is used, the leading edge of the reset signal should not occur until approximately 1 ms after V_{CC} has reached normal operating voltage. This delay time is needed for internal signals to stabilize.

Register Description

The registers in the DCP which can be directly addressed through the Master Port are shown with their addresses in Figure 1. A brief description of these registers and others not directly accessible is given below.

Command Register

Data written to the 8-bit, write-only Command Register through the Master Port is interpreted as an instruction. A detailed description of each command is given under Detailed Description, and the commands and their binary representations are summarized in Figure 2.

Status Register

The bit assignments in the read-only Status Register are shown in Figure 4. The PAR, AFLG, SFLG and MFLG bits indicate the status of the like-named output pins, as do the bits Busy and Command Pending when the DCP is in Direct Control Mode (C/$\overline{K}$ HIGH). In each case, the output signal will be active LOW when the corresponding status bit is a "1." The Parity bit indicates the parity of the most recently entered key byte. The LPAR bit, on the other hand, indicates whether any key byte with even parity has been encountered since the last Reset or Load Key command.

The Busy bit will be a "1" whenever the ciphering algorithm unit is actively encrypting or decrypting data, either as a response to a command, such as Load Encrypted Key (in which case the Command Pending bit will be a "1"), or in the ciphering of regular text (indicated by the Start/Stop bit being a "1"). The Busy bit will remain a "1" even after ciphering is complete if the ciphered data cannot be transferred to the Output Register because that register still contains output from a previous ciphering cycle. Busy will be "0" at all other times, including if no ciphering is possible because no data has been written to the Input Register.

The Command Pending bit will be set to "1" by any command whose execution requires the transfer of data to or from a non-addressable internal register, such as when writing key bytes to the E Key Register or reading bytes from the IVE Register. Thus, Command Pending will be set following all commands except the three Start commands, the Stop command and the Software Reset command. Command Pending will return to "0" after all eight bytes have been transferred following Load Clear, Read Clear or Read Encrypted commands and after data has been transferred, decrypted and loaded into the desired register following Load Encrypted commands.

C/$\overline{K}$	MP2	MP1	MR/$\overline{W}$	$\overline{MCS}$	Register Addressed
0	X	0	0	0	Input Register
0	X	0	1	0	Output Register
0	0	1	0	0	Command Register
0	0	1	1	0	Status Register
0	1	1	X	0	Mode Register
X	X	X	X	1	No Register Accessed
1	X	X	0	0	Input Register
1	X	X	1	0	Output Register

Figure 1. Master Port Register Addresses

Hex Code	Command
90	Load Clear M Key through Auxiliary Port
91	Load Clear E Key through Auxiliary Port
92	Load Clear D Key through Auxiliary Port
11	Load Clear E Key through Master Port
12	Load Clear D Key through Master Port
B1	Load Encrypted E Key through Auxiliary Port
B2	Load Encrypted D Key through Auxiliary Port
31	Load Encrypted E Key through Master Port
32	Load Encrypted D Key through Master Port
85	Load Clear IVE through Master Port
84	Load Clear IVD through Master Port
A5	Load Encrypted IVE through Master Port
A4	Load Encrypted IVD through Master Port
8D	Read Clear IVE through Master Port
8C	Read Clear IVD through Master Port
A9	Read Encrypted IVE through Master Port
A8	Read Encrypted IVD through Master Port
39	Encrypt with Master Key
41	Start Encryption
40	Start Decryption
C0	Start
E0	Stop
00	Software Reset

Figure 2. Command Codes in Multiplexed Control Mode

Pins				Command Initiated
C/$\overline{K}$	AUX7-K/$\overline{D}$	AUX6-E/$\overline{D}$	AUX5-S/$\overline{S}$	
H	L	L	↑	Start Decryption
H	L	H	↑	Start Encryption
H	L	X	↓	Stop
H	↑	L	L	Load D Key Clear through Master Port
H	↑	H	L	Load E Key Clear through Master Port
H	↓	X	L	End Load Key Command
H	H	X	H	Not Allowed
L	Data	Data	Data	AUX Pins Become Key-Byte Inputs

Figure 3. Implicit Command Sequences in Direct Control Mode

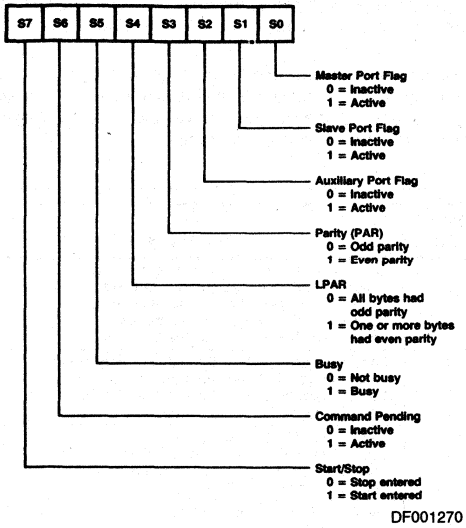


Figure 4. Status Register Bit Assignments

The Start/Stop bit is set to "1" when one of the Start commands is entered, and is reset to "0" whenever a reset occurs or when a new command other than a Start is entered.

Mode Register

Bit assignments in this 5-bit read/write register are shown in Figure 5. The Cipher Type bits (M_1, M_0) indicate to the DCP which ciphering algorithm is to be used. On reset, the Cipher Type defaults to Electronic Code Book.

Configuration bits (M_3, M_2) indicate which data ports are to be associated with the Input and Output Registers and flags. When these bits are set to the Single Port, Master-only configuration (M_3, M_2 = 10) the Slave Port is disabled, and no manipulation of Slave Port Chip Select ($\overline{SCS}$) or Data Strobe ($\overline{SDS}$) can result in data movement through the Slave Port; all data transfers are accomplished through the Master Port, as described more fully in Detailed Description. Both $\overline{MFLG}$ and $\overline{SFLG}$ are used in this configuration; $\overline{MFLG}$ gives the status of the Input Register and $\overline{SFLG}$ the Output Register.

When the Configuration Bits are set to one of the Dual Port configurations (M_3, M_2 = 00 or 01), both the Master and Slave Ports are available for input and output. When M_3, M_2 = 01 (the default configuration), the Master Port handles clear data while the Slave Port handles encrypted data. Configuration M_3, M_2 = 00 reverses this assignment. Actual data direction at any particular moment is controlled by the Encrypt/Decrypt bit.

The Encrypt/Decrypt bit (M_4) instructs the DCP algorithm processor to encrypt or decrypt the data from the Input Register using the ciphering method specified by the Cipher Type bits. The Encrypt/Decrypt bit also controls data flow within the DCP. For example, when the configuration bits are "01" (Dual Port, Master Clear, Slave encrypted) and the Encrypt/Decrypt bit is "1" (encrypt), clear data will flow into the DCP through the Master Port, and encrypted data will flow out through the Slave Port. When the Encrypt/Decrypt bit is set to "0" (decrypt), data flow reverses.

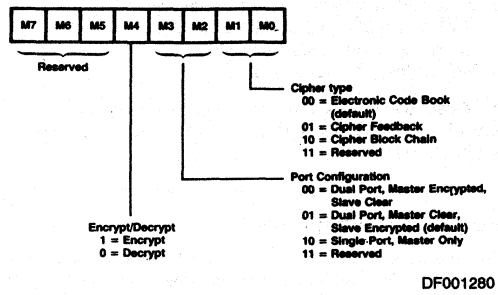


Figure 5. Mode Register Bit Assignments

Input Register

The 64-bit, write-only Input Register is organized to appear to the user as eight bytes of push down storage. A status circuit monitors the number of bytes that have been stored. The register is considered empty when the data stored in it has been or is being processed; it is considered full when one byte of data has been entered in Cipher Feedback or when eight bytes of data have been entered in the Electronic Code Book or Cipher Block Chain. If the user attempts to write data into the Input Register when it is full, the Input Register will disregard the attempt; no data in the register will be destroyed.

Output Register

The 64-bit, read-only Output Register is organized to appear to the user as eight bytes of pop-up storage. A status circuit detects the number of bytes stored in the Output Register. The register is considered empty when all the data stored in it has been read out by the host CPU, and is considered full if it still contains one or more bytes of output data. If a user attempts to read data from the Output Register when it is empty, the buffers driving the output bus will remain in a three-state condition.

The following multibyte registers cannot be directly addressed, but are loaded or read in response to commands written to the Command Register. (See Commands.)

M, E, D Key Registers

There are three 64-bit, write-only key registers in the DCP: the Master (M) Key Register; the Encrypt (E) Key Register; and the Decrypt (D) Key Register. The Master Key can be loaded only with clear data through the Auxiliary Port. The Encrypt and Decrypt Keys can be loaded in any of four ways: (1) as clear data through the Auxiliary Port; (2) as clear data through the Master Port; (3) as encrypted data through the Auxiliary Port; or (4) as encrypted data through the Master Port. In the last two cases, the encrypted data is first routed to the Input Register, decrypted using the M Key, and finally written to the target key register from the Output Register.

Initializing Vector Registers

Two 64-bit registers are provided to store feedback values for Cipher Feedback and Chained Block ciphering methods. One Initializing Vector (IVE) register is used during encryption; the other (IVD) during decryption. Both registers can be loaded with either clear or encrypted data through the Master Port (in the latter case, the data is decrypted before being loaded into the IV register), and both may be read out either clear or encrypted through the Master Port. (See Commands.)

Encrypt/ Decrypt M4	Port Configuration M3	M2	Input Register Flag	Output Register Flag
0	0	0	MFLG	SFLG
0	0	1	SFLG	MFLG
0	1	0	MFLG	SFLG
1	0	0	SFLG	MFLG
1	0	1	MFLG	SFLG
1	1	0	MFLG	SFLG

Figure 6. Association of Master Port Flag (MFLG) and Slave Port Flag (SFLG) with Input and Output Registers

Commands

All operations of the DCP result from command inputs, which are entered in Multiplexed Control Mode by writing a command byte to the Command Register. Command inputs are entered in Direct Control Mode by raising and lowering the logic levels on the AUX_7-K/$\overline{D}$, AUX_6-E/$\overline{D}$ and AUX_5-S/$\overline{S}$ pins. Figure 2 shows all commands that may be given in Multiplexed Control Mode. Figure 3 shows that subset executable in Direct Control Mode.

Load Clear M Key Through Auxiliary Port (90 Hex)

Load Clear E Key Through Auxiliary Port (91)

Load Clear D Key Through Auxiliary Port (92)

These commands override the data flow specifications set in the Mode Register and cause the Master (M), Encrypt (E), or Decrypt (D) Key Register to be loaded with eight bytes written to the Auxiliary Port. After the Load command is written to the Command Register, the Auxiliary Port Flag ($\overline{AFLG}$) will go active (LOW) and the corresponding bit in the Status Register (S_2) will go to "1," indicating that the device is able to accept key bytes at the Auxiliary Port pins. Additionally, the Command Pending bit (S_6) will go to "1" during the entire loading process.

Each byte is written by placing an active LOW signal on the Auxiliary Port Strobe ($\overline{ASTB}$) once data has been set up on the Auxiliary Port pins. The actual write process occurs on the rising (trailing) edge of $\overline{ASTB}$. (See Switching Characteristics for exact set up, strobe width, and hold times.)

The Auxiliary Port Flag ($\overline{AFLG}$) will go inactive immediately after the eighth strobe goes active (LOW). However, the Command Pending bit (S_6) will remain "1" for several more clock cycles, until the key loading process is completed. All key bytes are checked for correct (odd) parity as they are entered (see Parity Checking).

Load Clear E Key Through Master Port (11 Hex)

Load Clear D Key Through Master Port (12)

These commands are available in both multiplexed control and direct control modes. They override the data flow specifications set in the Mode Register and attach the Master Port inputs to the Encrypt (E) or Decrypt (D) Key Register, as appropriate, until eight key bytes have been written. In Multiplexed Control Mode, the command is initiated by writing the Load command to the Command Register. In Direct Control Mode, the command is initiated by raising the AUX_7-K/$\overline{D}$ control input while the AUX_5-S/$\overline{S}$ input is LOW. In this later case, the level on AUX_6-E/$\overline{D}$ determines which key register is written (HIGH = E Register).

Once the command has been recognized, the Command Pending bit (S_6 in the Status Register) will go to "1" and in Direct Control Mode AUX_3-$\overline{CP}$ will go active (LOW), indicating that key entry may proceed. The host system then writes exactly eight bytes to the Master Port (at the Input Register address in Multiplexed Control Mode). When the key register has been loaded, Command Pending will return to "0," and in Direct Control Mode, the AUX_3-$\overline{CP}$ output will go inactive, indicating that the DCP can accept the next command.

Load Encrypted E Key Through Auxiliary Port (B1 Hex)

Load Encrypted D Key Through Auxiliary Port (B2)

Execution of these commands (in Multiplexed Control Mode only) is similar to the Load Clear E (D) Key Through Auxiliary Port, except that key bytes are first decrypted using the Electronic Code Book algorithm and the Master (M) key, and then loaded into the appropriate key register, after having passed through the parity check logic (see Parity Checking).

The Command Pending bit (S_6) will be "1" during the entire decrypt-and-load operation. In addition, the Busy bit (S_5) will be "1" during the actual decryption process.

Load Encrypted E Key Through Master Port (31 Hex)

Load Encrypted D Key Through Master Port (32)

These commands (in Multiplexed Control Mode only) are similar in effect to Load Clear E (D) Key Through Master Port, except that key bytes are initially decrypted using the Electronic Code Book algorithm and the Master (M) Key, and then loaded byte-by-byte into the target key register, after having passed through the parity check logic (see Parity Checking).

The Command Pending bit (S_6) will be "1" during the entire decrypt-and-load operation. In addition, the Busy bit (S_5) will be "1" during the actual decryption process.

Load Clear IVE Register Through Master Port (85 hex)

Load Clear IVD Register Through Master Port (84)

These commands (in Multiplexed Control Mode only) are virtually identical to Load Clear E (or D) Key Through Master Port except that the data written to the Input Register address is routed to the Encryption Initializing Vector (IVE) or Decryption Initializing Vector (IVD) Register instead of a key register, and no parity checking occurs. Command Pending (S_6) is a "1" during the entire loading process.

Load Encrypted IVE Register Through Master Port (A5 Hex)

Load Encrypted IVD Register Through Master Port (A4)

These commands are analogous to the Load Encrypted E (or D) Key Through Master Port commands. The data flow specifications set in the Mode Register are overridden, and the eight vector bytes are decrypted using the Decryption (D) Key and the Electronic Code Book algorithm. The resulting clear vector bytes are loaded into the target Initializing Vector register, and no parity checking occurs. The Busy bit (S_5) does not go to "1" during the decryption process, but Command

Pending (S_6) will be "1" during the entire decryption-and-load operation.

Read Clear IVE Register Through Master Port (8D Hex)

Read Clear IVD Register Through Master Port (8C)

The effect of these commands (in Multiplexed Control Mode only) is to override the data flow specifications set in the Mode Register and to connect the appropriate Initializing Vector Register to the Master Port at the Output Register address. In this state, each IV register appears as eight bytes of FIFO storage. The first byte of data will be available 6 clocks after loading the command register. The command pending bit will be set to "1" and will remain a "1" until sometime after the eighth byte is read out. The host system has the responsibility to read out exactly eight bytes.

Read Encrypted IVE Register Through Master Port (A9 Hex)

Read Encrypted IVD Register Through Master Port (A8)

The effect of these commands (in Multiplexed Control Mode only) is to override the specifications set in the Mode Register and to encrypt the contents of the specified Initializing Vector Register using the Electronic Code Book algorithm and the Encrypt (E) Key. The resulting cipher text is placed in the Output Register, from which it can be read out as eight bytes, through the Master Port. During the actual encryption process, the Busy bit (S_5) will be "1." When Busy goes to "0," the encrypted vector bytes are ready to be read out. Command Pending (S_6) will be "1" during the entire encryption-and-output process, and will go to "0" when the eighth byte is read out. The host system is responsible for reading out exactly eight bytes.

Encrypt with Master (M) Key (39 Hex)

This command, in Multiplexed Control Mode only, overrides the data flow specifications set in the Mode Register and causes the DCP to accept eight bytes from the Master Port written to the Input Register. When eight bytes have been received, the DCP encrypts the input using the Master (M) Key. The encrypted data is loaded into the Output Register where it may be read out through the Master Port. The Command Pending (S_6) and Busy (S_5) bits are used to sense the three phases of this operation. Command Pending goes to "1" as soon as the Input Register can accept data. When exactly eight bytes have been entered, the Busy bit will go to "1" until the encryption process is complete.

When Busy goes to "0," the encrypted data is available to be read out. Command Pending will return to "0" when the eighth byte has been read.

Start Encryption (41 Hex)

Start Decryption (40)

Start (C0)

The three "Start" commands begin normal data ciphering by setting the Start/Stop bit (S_7) in the Status Register to "1." The Start Encryption and Start Decryption commands explicitly specify the ciphering direction by forcing the Encrypt/Decrypt bit (M_4) in the Mode Register to "1" of "0," respectively; whereas, Start uses the current state of the Encrypt/Decrypt bit, as specified in a previous Mode Register load.

When a Start command has been entered, the Port Status Flag ($\overline{MFLG}$ or $\overline{SFLG}$) associated with the Input Register will become active (LOW), indicating that data may be written to the Input Register to begin ciphering.

In Direct Control Mode, the Start command is issued by raising the level on the AUX_5-S/$\overline{S}$ input (see Figure 3). The ciphering direction is specified by the level on AUX_6-E/$\overline{D}$. If AUX_6-E/$\overline{D}$ is high when AUX_5-S/$\overline{S}$ goes HIGH, the command is Start Encryption; if AUX_6-E/$\overline{D}$ is low, it is Start Decryption.

Stop (E0 Hex)

The Stop command clears the Start/Stop bit (S_7) in the Status Register to "0." This causes the input flag ($\overline{MFLG}$ or $\overline{SFLG}$) to become inactive and inhibits the loading of any further input into the algorithm unit. If ciphering is in progress (Busy bit (S_5) is "1" or AUX_2-$\overline{BSY}$ is active), it will finish and any data in the Output Register will remain accessible:

In Direct Control Mode, the Stop command is implied when the signal level on the AUX_5-S/$\overline{S}$ input goes from HIGH to LOW (see Figure 3).

Software Reset (00)

This command has the same effect as a hardware reset ($\overline{MAS}$ and $\overline{MDS}$ low): it forces the DCP back to its default configuration, and all processing flags go into inactive mode. The default configuration includes setting the Mode Register to Electronic Code Book cipher type, and Dual Port Configuration with Master Port clear, Slave Port encrypted.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5 to +7.0V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5V ±5%	0V
Industrial	−40°C to 85°C	5V ±5%	0V

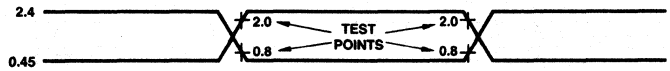
Operating ranges define those limits over which the functionality of the device is guaranteed.

Am9518/AmZ8068
DC CHARACTERISTICS over operating range unless otherwise specified

T_A = 0 to 70°C, V_{CC} = +5.0V ±5%, V_{SS} = 0V

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		−0.5		.8	Volts
V_{IH}	Input High Voltage		2.2		V_{CC}	Volts
V_{OL}	Output Low Voltage	I_{OL} = 3.2mA			.40	Volts
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4			Volts
I_I	Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{CC}$			±10	μA
I_{OZ}	Output Leakage Current	V_{SS} + .40 ≤ $V_{IN} \leq V_{CC}$			±10	μA
I_{CC}	Supply Current (AVER.)			150	250	mA

SWITCHING TEST INPUT WAVEFORM



WF003570

00618B

Refer to page 7-1 for Essential Information on Military Devices

Am9518/AmZ8068 SWITCHING CHARACTERISTICS

The table below specifies the guaranteed performance of this device over the Commercial Operating Range of 0 to +70°C with V_{CC} from 4.75 to 5.25V. All data are in nanoseconds. Switching tests are made with inputs and outputs measured at 0.8V for a LOW and 2.0V for a HIGH. Outputs are fully loaded, with $C_L \geq$ =50pF. See Switching Waveform figures for graphic illustration of timing parameters.

SWITCHING CHARACTERISTICS over operating range (Note 1)

Number	Parameters	Description	Am9518			AmZ8068			Units
			Min	Typ	Max	Min	Typ	Max	
Clock									
1	TWH	Clock Width (HIGH)	150			115			ns
2	TWL	Clock Width (LOW)	150			115			ns
3	TC	Clock HIGH to Next Clock HIGH (Clock Cycle)	320		1000	250		1000	ns
Reset									
5	TG1LG1H	$\overline{MDS} \cdot \overline{MAS}$ LOW to $\overline{MDS} \cdot \overline{MAS}$ HIGH (Reset Pulse Width)	TC			TC			ns
6	TCHG1H	Clock HIGH to $\overline{MDS} \cdot \overline{MAS}$ HIGH	0		50	0		50	ns
Direct Control Mode									
9	TNLMH	S/$\overline{S}$ LOW to C/$\overline{K}$ HIGH (Setup)	3TC			3TC			ns
10	TKLMH	K/$\overline{D}$ LOW to C/$\overline{K}$ HIGH (Setup)	3TC			3TC			ns
11	TMHNH	C/$\overline{K}$ HIGH to S/$\overline{S}$ HIGH	6TC			6TC			ns
12	TMHKH	C/$\overline{K}$ HIGH to K/$\overline{D}$ HIGH	6TC			6TC			ns
14	TEVKH	E/$\overline{D}$ VALID to K/$\overline{D}$ HIGH (Setup)	3TC			3TC			ns
15	TKHRL	K/$\overline{D}$ HIGH to $\overline{CP}$ LOW			300			300	ns
17	TKLEX	K/$\overline{D}$ LOW to E/$\overline{D}$ INVALID (Hold)	TC			TC			ns
19	TCLNV	Clock LOW to S/$\overline{S}$ VALID	20		80	20		80	ns
20	TEVNH	E/$\overline{D}$ VALID to S/$\overline{S}$ HIGH (Setup)	3TC			3TC			ns
21	TNHF1L	S/$\overline{S}$ HIGH to $\overline{MFLG}$ ($\overline{SFLG}$) LOW (Port Input Flag)			300			230	ns
22	TCHF1L	Clock HIGH to $\overline{MFLG}$ ($\overline{SFLG}$) LOW (Port Input Flag) (Note 2)			300			230	ns
24	TCHBL	Clock HIGH to $\overline{BSY}$ LOW			400			300	ns
25	TCLBH	Clock LOW to $\overline{BSY}$ HIGH			300			230	ns
27	TCHF1L	Clock HIGH to $\overline{MFLG}$ ($\overline{SFLG}$) LOW (Port Output Flag)			300			230	ns
28	TNLF1H	S/$\overline{S}$ LOW to $\overline{MFLG}$ ($\overline{SFLG}$) HIGH (Port Input Flag) (Note 3)			300			230	ns
Multiplexed Control Mode — Master Port									
32	TWA	$\overline{MAS}$ Width (LOW)	115			80			ns
34	TS1LAH	$\overline{MCS}$ LOW to $\overline{MAS}$ HIGH (Setup)	0			0			ns
35	TAHS1H	$\overline{MAS}$ HIGH to $\overline{MCS}$ HIGH (Hold)	60			60			ns
36	TD1VAH	Address-in VALID to $\overline{MAS}$ HIGH (Address Setup Time)	90			55			ns
37	TAHD1X	$\overline{MAS}$ HIGH to Address-in INVALID (Address Hold Time)	60			60			ns

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Parameters	Description		Am9518 Min	Am9518 Typ	Am9518 Max	AmZ8068 Min	AmZ8068 Typ	AmZ8068 Max	Units
Master (Slave) Port Read/Write										
40	TS1LG1L	$\overline{MCS}$ ($\overline{SCS}$) LOW to $\overline{MDS}$ ($\overline{SDS}$) LOW (Select Setup) (Note 4)		100			100			ns
41	TG1HS1H	$\overline{MDS}$ ($\overline{SDS}$) HIGH to $\overline{MCS}$ ($\overline{SCS}$) HIGH (Select Hold Time) (Note 4)		25			25			ns
42	TWVG1L	MR/$\overline{W}$ VALID to $\overline{MDS}$ LOW (Setup)		100			100			ns
43	TG1HWX	$\overline{MDS}$ HIGH to MR/$\overline{W}$ INVALID (Hold)		25			25			ns
44	TG1LG1H	$\overline{MDS}$ ($\overline{SDS}$) LOW to $\overline{MDS}$ ($\overline{SDS}$) HIGH	Width – Write, Data Read	160		1000	125		1000	ns
			Width – Status Register Read	300		1000	200		1000	
45	TCLG1H	Clock LOW to $\overline{MDS}$ ($\overline{SDS}$) HIGH (Note 11)		0		TWL – 100	0		TWL – 65	
46	TGIHG1L	$\overline{MDS}$ ($\overline{SDS}$) HIGH to $\overline{MDS}$ ($\overline{SDS}$) LOW (Data Strobe Recovery Time)		160			125			ns
47	TD1VG1H	Write-Data VALID $\overline{MDS}$ ($\overline{SDS}$) HIGH	Setup Time – Key Load (Note 8)	160			125			ns
			Setup Time – Data Write	160			125			
			Setup Time – Command/ Mode Register Write	160			125			
48	TG1HD1X	$\overline{MDS}$ ($\overline{SDS}$) HIGH to Write-Data INVALID (Hold Time – All Writes)		25			25			ns
49	TG1LQ1V	$\overline{MDS}$ ($\overline{SDS}$) LOW to Read-Data VALID	Read Access Time – Status Register			300			200	ns
			Read Access Time – Data			150			120	
50	TG1HQ1V	$\overline{MDS}$ ($\overline{SDS}$) HIGH to Read-Data INVALID (Read Hold Time)		5			5			ns
51	TG1LF1H	$\overline{MDS}$ ($\overline{SDS}$) LOW to $\overline{MFLG}$ ($\overline{SFLG}$) HIGH (Last Strobe) (Note 5)				160			125	ns
52	TG1LRH	$\overline{MDS}$ HIGH to CP HIGH	Last Strobe, Key Load			TC + 500			TC + 500	ns
53	TG1HNL	$\overline{MDS}$ ($\overline{SDS}$) HIGH to S/$\overline{S}$ LOW (Hold Time) (Note 9)		4TC			4TC			ns
54	TG1HPV	$\overline{MDS}$ HIGH to $\overline{PAR}$ VALID (Key Write)				250			200	ns
Auxiliary Port Key Entry										
61	TG3LG3H	$\overline{ASTB}$ LOW to $\overline{ASTB}$ HIGH (Width)		160			160			ns
62	TCLG3H	Clock LOW to $\overline{ASTB}$ HIGH		0		50	0		50	ns
63	TG3HG3L	$\overline{ASTB}$ HIGH to Next $\overline{ASTB}$ LOW (Recovery Time)		320			250			ns
64	TD3VG3H	Write-Data VALID to $\overline{ASTB}$ HIGH (Data Setup Time)		300			200			ns
65	TG3HD3X	$\overline{ASTB}$ HIGH to Write-Data INVALID (Data Hold Time)		80			80			ns
66	TG3HPV	$\overline{ASTB}$ HIGH to $\overline{PAR}$ VALID				300			200	ns
67	TG3LF3H	$\overline{ASTB}$ LOW to $\overline{AFLG}$ HIGH (Last Strobe)				300			230	ns

Notes:
1. All input transition times assumed ⩽ 20ns.
2. Parameter TCHF1L applies to all input blocks except the first (when S/$\overline{S}$ first goes HIGH).
3. When S/$\overline{S}$ goes inactive (LOW) in direct control mode, the flag associated with the input port will turn off.
4. Direct control mode only.
5. In Cipher Feedback, the port flag ($\overline{MFLG}$ or $\overline{SFLG}$) will go inactive following the leading edge of the first data strobe ($\overline{MDS}$ or $\overline{SDS}$); in all other modes and operations, the flags go inactive on the eighth data strobe.
6. Do not remove K/$\overline{D}$ until CP is inactive (HIGH).
7. Do not change E/$\overline{D}$ until $\overline{MFLG}$ ($\overline{SFLG}$) is inactive (HIGH).
8. 300ns Min if parity check is needed.
9. In Cipher Feedback mode, $\overline{BSY}$ must be inactive before S/$\overline{S}$ goes LOW.
10. $\overline{AFLG}$ must go active (LOW) before $\overline{ASTB}$ goes active (LOW).
11. This limit is valid when the clock frequency is 4MHz. At slower clock rates, the range is wider.

Parameter Naming Convention for DCP

Name: **T A [N] B C [N] D**

A C Signal names (see below)

B D Signal States:

- H High
- L Low
- V Valid
- X Not Valid
- Z High Impedance

(N) Optional Port number (modifies signal name):

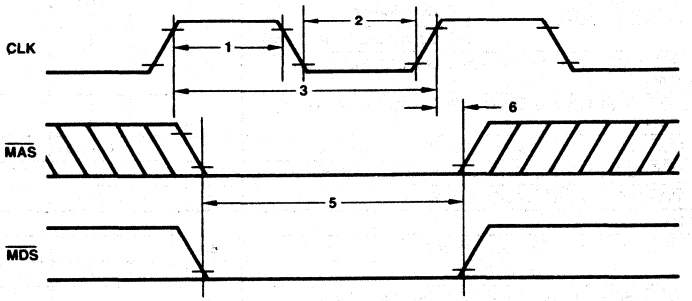
- 1 = Master Port
- 2 = Slave Port
- 3 = AUX (Key) Port

Signal Name Characters

A Address Strobe
B $\overline{BSY}$
C Clock
D* Data In (or address at Master Port) D1, D2, D3
E E/$\overline{D}$
F* Flag ($\overline{MFLG}$, $\overline{SFLG}$, $\overline{AFLG}$)
G* Data Strobe ($\overline{MDS}$, $\overline{SDS}$, $\overline{ASTB}$)
K K/$\overline{D}$
M C/$\overline{K}$ (Mode)
N S/$\overline{S}$ (Start)
P $\overline{PAR}$
Q* Data Out (Master or Slave Port)
R $\overline{CP}$
S* Chip Select (Master or Slave Port)
W MR/$\overline{W}$

*Modified by Port number. Example: D1 = Data In, Master Port; F2 = SFLG; G3 = ASTB; Q2 = Data Out, Slave Port; S1 = MCS.

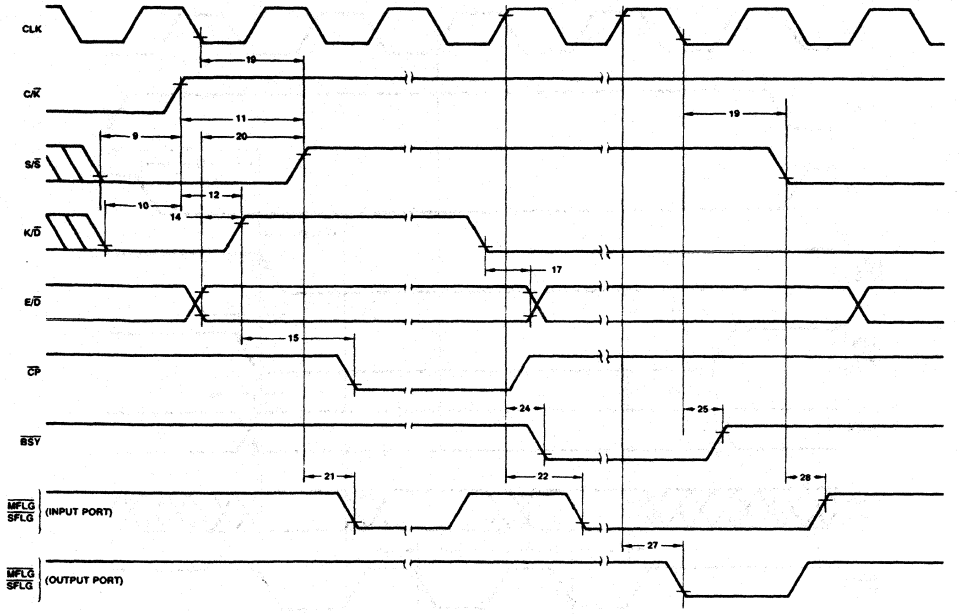
SWITCHING WAVEFORMS



WF003580

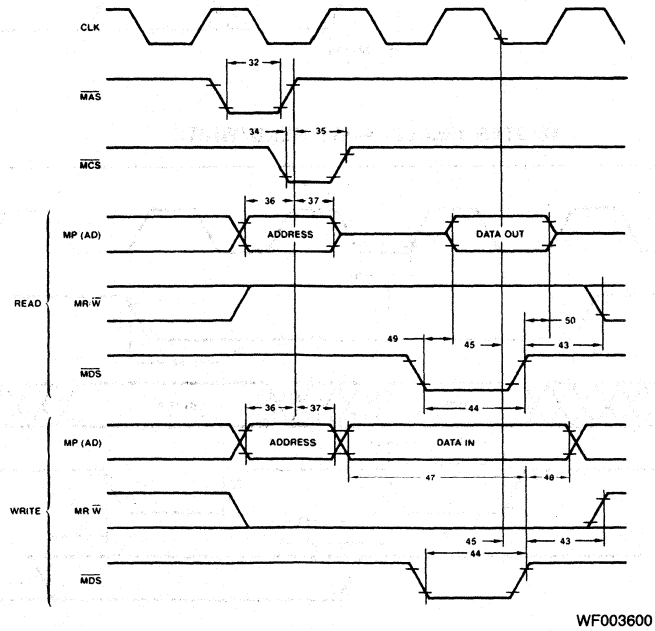
CLOCK AND RESET

SWITCHING WAVEFORMS (Cont.)



WF003590

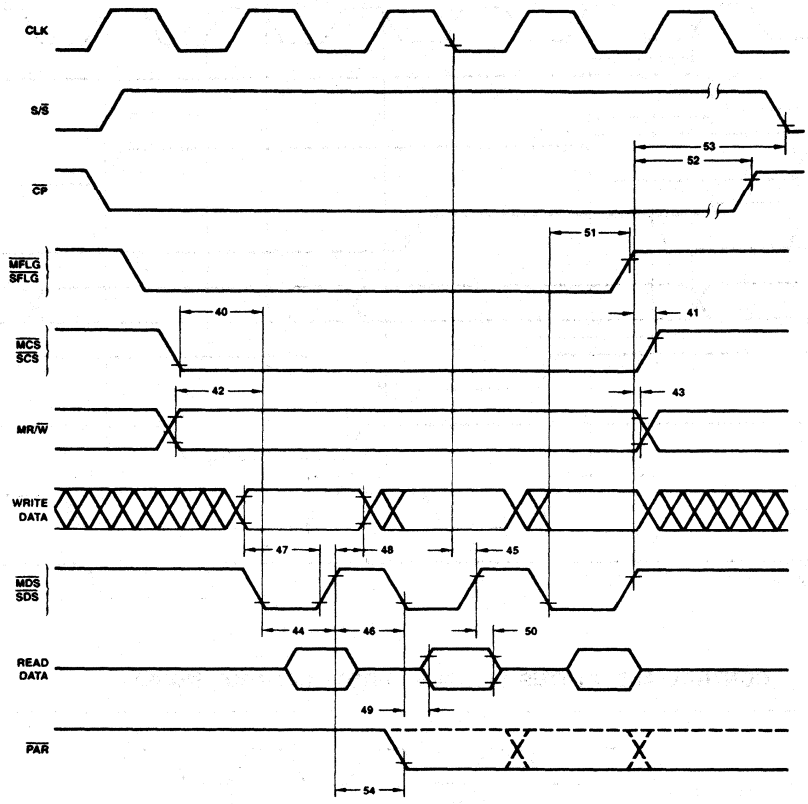
CONTROL AND STATUS SIGNALS (DIRECT CONTROL MODE)



WF003600

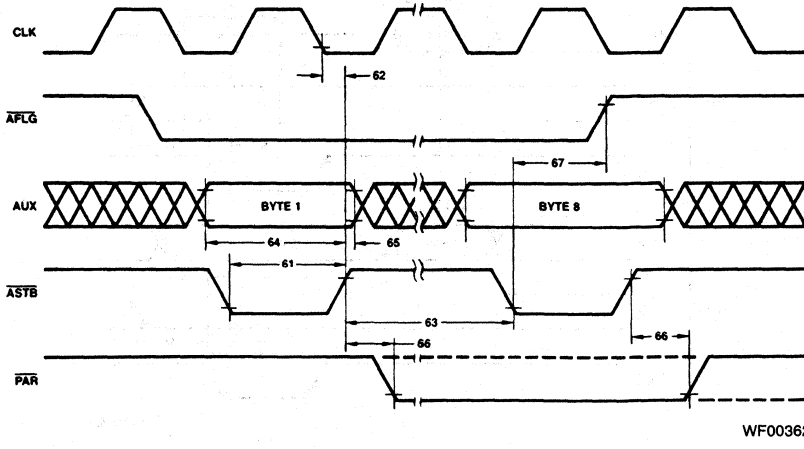
MASTER PORT, MULTIPLEXED CONTROL MODE READ/WRITE

SWITCHING WAVEFORMS (Cont.)



WF003610

MASTER (SLAVE) PORT READ/WRITE

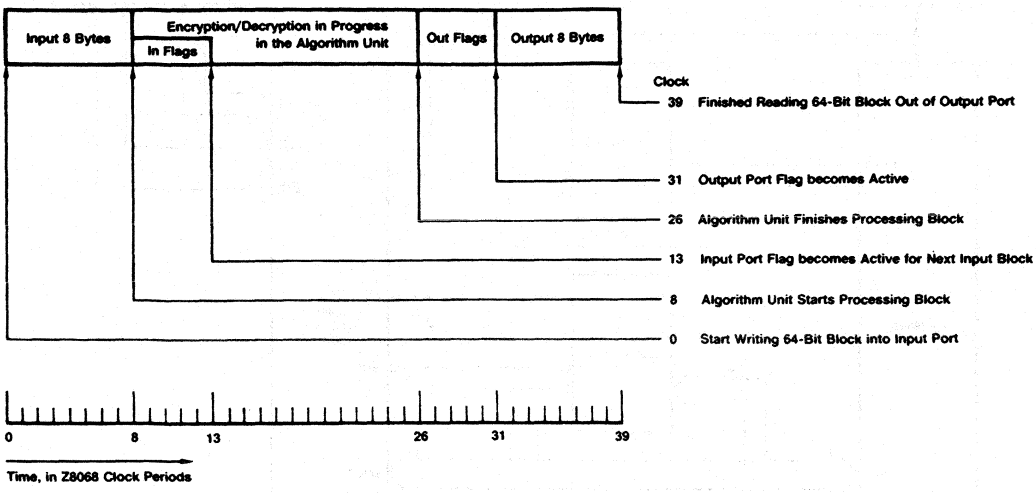


WF003620

AUXILIARY-PORT KEY ENTRY

TIMING FOR PIPELINED, DUAL-PORT OPERATION

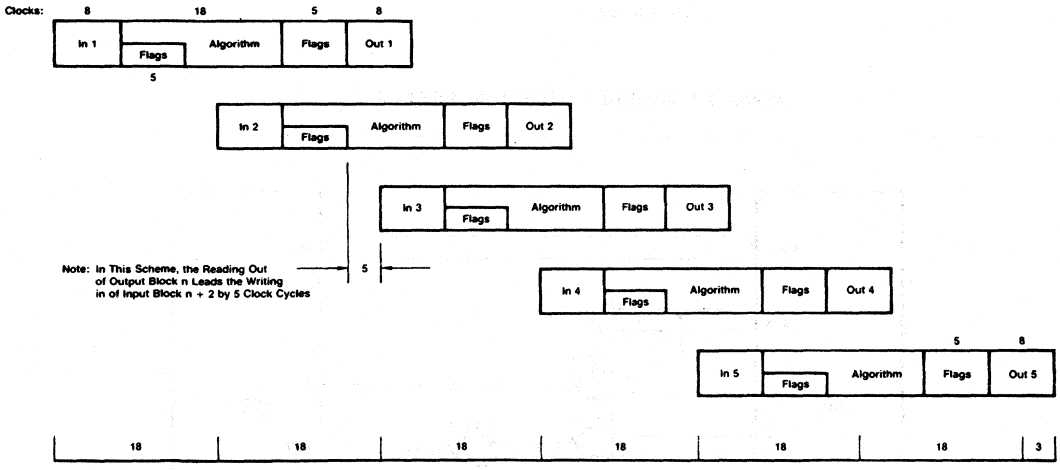
Detailed Timing of 1 Block



DF001290

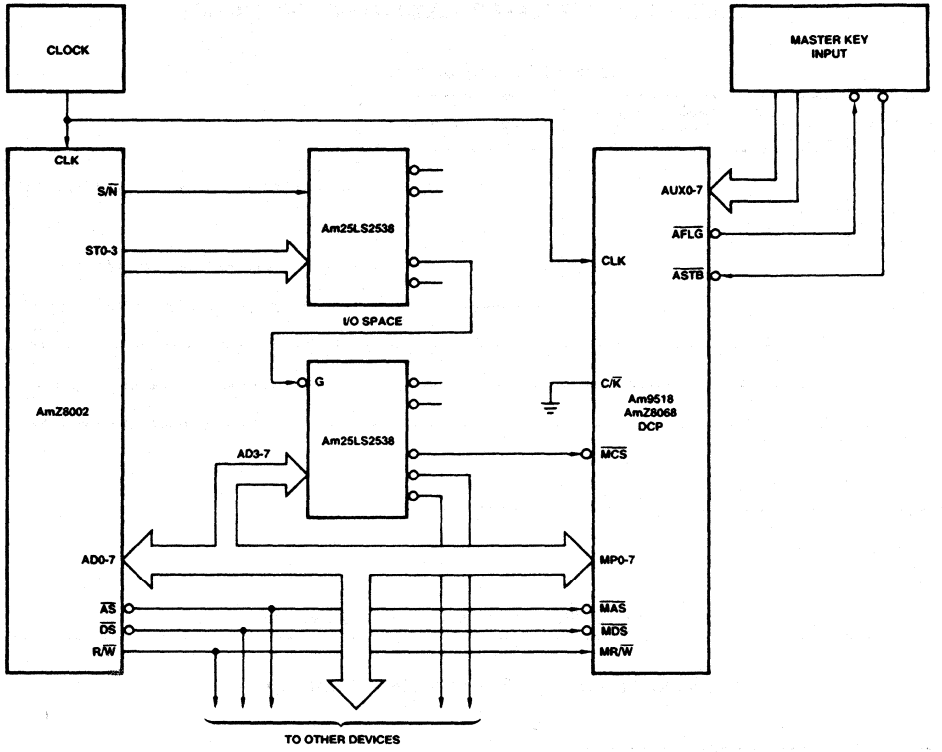
Note: AmZ8068 clock period = 250 nanoseconds

Pipelining Scheme A: Minimum Timing Operation



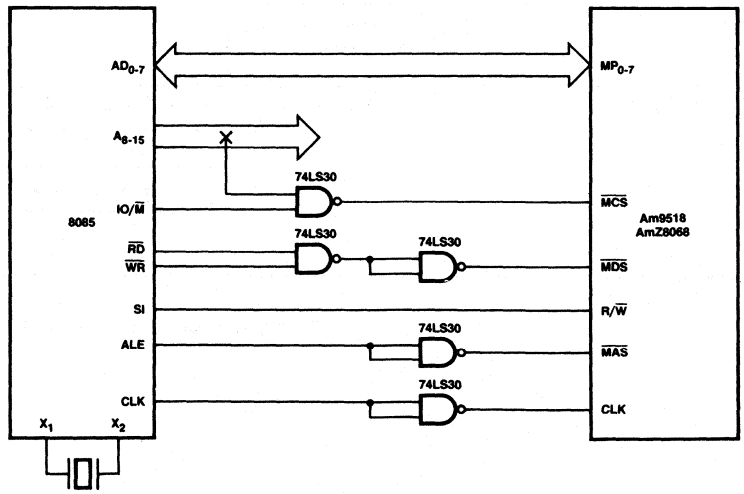
DF001300

For n blocks, total time = (n + 1) x 18 + 3



AF002250

MINIMUM Am9518/AmZ8068 INTERFACE



AF002260

MINIMUM 8085 TO Am9518/AmZ8068 INTERFACE

Am9519A

Universal Interrupt Controller

DISTINCTIVE CHARACTERISTICS

- Eight individually maskable interrupt inputs reduce CPU overhead
- Unlimited interrupt channel expansion with no extra hardware
- Programmable 1-byte to 4-byte response provides vector address and message protocol for 8-bit CPUs
- Rotating and fixed priority resolution logic
- Software interrupt request capability
- Common vector and polled mode options
- Automatic hardware clear of in-service interrupts reduces software overhead
- Polarity control of interrupt inputs and outputs
- Reset minimizes software initialization by automatically generating CALL to location zero

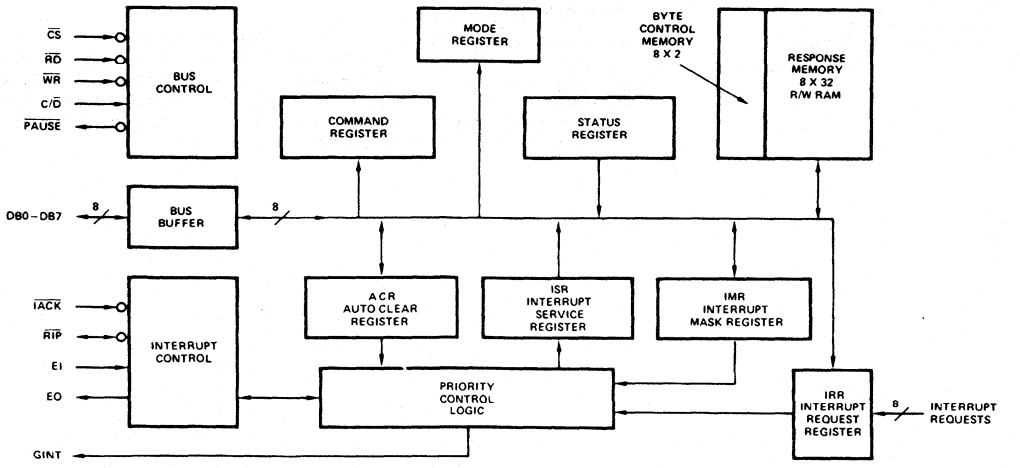
GENERAL DESCRIPTION

The Am9519A Universal Interrupt Controller is a processor support circuit that provides a powerful interrupt structure to increase the efficiency and versatility of microcomputer-based systems. A single Am9519A manages up to eight maskable interrupt request inputs, resolves priorities and supplies up to four bytes of fully programmable response for each interrupt. It uses a simple expansion structure that allows many units to be cascaded for control of large numbers of interrupts. Several programmable control features are provided to enhance system flexibility and optimization.

The Universal Interrupt Controller is designed with a general purpose interface to facilitate its use with a wide range of digital systems, including most popular 8-bit microprocessors. Since the response bytes are fully programmable, any instruction or vectoning protocol appropriate for the host processor may be used.

When the Am9519A controller receives an unmasked interrupt request, it issues a Group Interrupt output to the CPU. When the interrupt is acknowledged, the controller outputs the one-to-four byte response associated with the highest priority unmasked interrupt request. The ability of the CPU to set interrupt requests under software control permits hardware prioritization of software tasks and aids system diagnostic and maintenance procedures.

BLOCK DIAGRAM

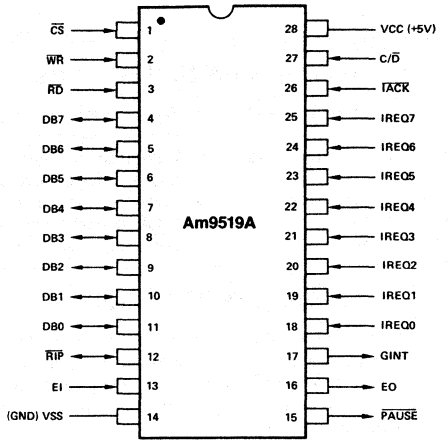


BD003280

00147B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-28, P-28

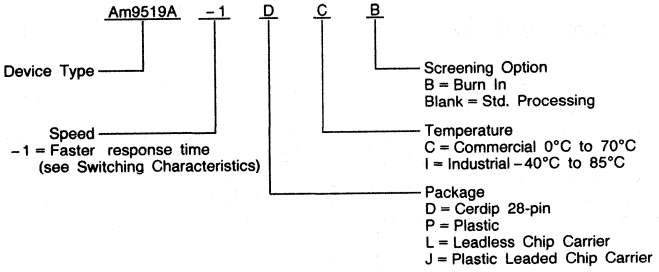


CD005100

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9519A	DC, DCB, DI, DIB, PC, PCB,
Am9519A-1	PI, PIB, LC, LCB, LI, LIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}		+5 Volt Power Supply.
14	V_{SS}		Ground.
11-4	DB0-DB7	I/O	(Data Bus). The eight bidirectional data bus signals are used to transfer information between the Am9519A and the system data bus. The direction of transfer is controlled by the $\overline{IACK}$, $\overline{WR}$ and $\overline{RD}$ input signals. Programming and control information are written into the device; status and response data are output by it.
1	$\overline{CS}$	I	(Chip Select). The active low Chip Select input enables read and write operations on the data bus. Interrupt acknowledge responses are not conditioned by $\overline{CS}$.
3	$\overline{RD}$	I	(Read). The active low Read signal is conditioned by $\overline{CS}$ and indicates that information is to be transferred from the Am9519A to the data bus.
2	$\overline{WR}$	I	(Write). The active low Write signal is conditioned by $\overline{CS}$ and indicates that data bus information is to be transferred from the data bus to a location within the Am9519A.
27	C/$\overline{D}$	I	(Control/Data). The C/$\overline{D}$ control signal selects source and destination locations for data bus read and write operations. Data read or write transfers are made to or from preselected internal registers or memory locations. Control write operations load the command register and control read operations output the status register.
18-25	IREQ0-IREQ7	I	(Interrupt Request). The Interrupt Request signals are used by external devices to indicate that service by the host CPU is desired. IREQ inputs are accepted asynchronously and they may be programmed for either a HIGH-to-LOW or LOW-to-HIGH edge transition. Active inputs are latched internally in the Interrupt Request Register. After the IRR bit is cleared, an IREQ transition of the programmed polarity must occur to initiate another request.
12	$\overline{RIP}$	I/O	(Response In Process). Response In Process is a bidirectional signal used when two or more Am9519A circuits are cascaded. It permits multibyte response transfers to be completed without interference from higher priority interrupts. An Am9519A that is responding to an acknowledged interrupt will treat $\overline{RIP}$ as an output and hold it LOW until the acknowledge response is finished. An Am9519A without an acknowledged interrupt will treat $\overline{RIP}$ as an input and will ignore $\overline{IACK}$ pulses as long as $\overline{RIP}$ is LOW. The $\overline{RIP}$ output is open drain and requires an external pull-up resistor to VCC.
26	$\overline{IACK}$	I	(Interrupt Acknowledge). The active-low Interrupt Acknowledge line indicates that the external system is asking for interrupt response information. Depending on the programmed state of the Am9519A, it will accept 1, 2, 3 or 4 $\overline{IACK}$ pulses; one response byte is transferred per pulse. The first $\overline{IACK}$ pulse causes selection of the highest priority unmasked pending interrupt request and generates a $\overline{RIP}$ output signal.
15	$\overline{PAUSE}$	O	(Pause). The active-low Pause signal is used to coordinate interrupt responses with data bus and control timing. Pause goes LOW when the first $\overline{IACK}$ is received and remains LOW until $\overline{RIP}$ goes LOW. The external system can use Pause to stretch the acknowledge cycle and allow the control timing to automatically adjust to the actual priority resolution delays in the interrupt system. Second, third and fourth response bytes do not cause Pause to go LOW. Pause is an open drain output and requires an external pull-up resistor to VCC.
16	EO	O	(Enable Out). The active-high EO signal is used to implement daisy-chained cascading of several Am9519A circuits. EO is connected to the EI input of the next lower priority chip. On receipt of an interrupt acknowledge, each EO will go inactive until it has been determined that no valid interrupt request is pending on that chip. If an active request is present, EO remains LOW. EO is also held LOW when the master mask bit is active, thus disabling all lower priority chips.
13	EI	I	(Enable in). The active-high EI signal is used to implement daisy-chained cascading of several Am9519A circuits. EI is connected to EO of the next higher priority chip. It may also be used as a hardware disable input for the interrupt system. When EI is LOW, $\overline{IACK}$ inputs will not affect ISR; however, $\overline{PAUSE}$ will go LOW until $\overline{RIP}$ goes LOW. EI is internally pulled up to VCC so that no external pull-up is needed when EI is not used.
17	GINT	O	(Group Interrupt). The Group Interrupt output signal indicates that at least one unmasked interrupt request is pending. It may be programmed for active-high or active-low polarity. When active-low, the output is open drain and requires an external pull-up resistor to VCC. Since a glitch on GINT occurs approximately 100nsec after the last IACK pulse, this pin should not be connected to edge sensitive devices.

PRODUCT OVERVIEW

Register Description

Interrupt Request Register (IRR): The 8-bit IRR is used to store pending interrupt requests. A bit in the IRR is set whenever the corresponding IREQ input goes active. Bits may also be set under program control from the CPU, thus permitting software generated interrupts. IRR bits may be cleared under program control. An IRR bit is automatically cleared when its interrupt is acknowledged. All IRR bits are cleared by a reset function.

Interrupt Service Register (ISR): The 8-bit ISR contains one bit for each IREQ input. It is used to indicate that a pending interrupt has been acknowledged and to mask all lower priority interrupts. When a bit is set by the acknowledge logic in the ISR, the corresponding IRR bit is cleared. If an acknowledged interrupt is not programmed to be automatically cleared, its ISR bit must be cleared by the CPU under program control when it is desired to permit interrupts from lower priority devices. When the interrupt is programmed for automatic clearing, the ISR bit is automatically reset during the acknowledge sequence. All ISR bits are cleared by a reset function.

Interrupt Mask Register (IMR): The 8-bit IMR is used to enable or disable the individual interrupt inputs. The IMR bits correspond to the IREQ inputs, and all eight may be loaded, set or cleared in parallel under program control. In addition, individual IMR bits may be set or cleared by the CPU. Care must be taken therefore when disabling a specific channel by setting its IMR bit. If that bit is causing the GINT pin to be active, a lock-up condition can occur if the CPU recognizes the interrupt and then the Am9519A removes the request. During the $\overline{IACK}$ cycle, $\overline{PAUSE}$ will go LOW and stay LOW. The solution is to disable CPU interrupts prior to writing to the IMR and then re-enable them. A reset function will set all eight

mask bits, disabling all requests. A mask bit that is set does not disable the IRR, and an IREQ that arrives while a corresponding mask bit is set will cause an interrupt later when the mask bit is cleared. Only unmasked interrupt inputs can generate a Group Interrupt output.

Response Memory: An 8 x 32 read/write response memory is included in the Am9519A. It is used to store up to four bytes of response information for each of the eight interrupt request inputs. All bits in the memory are programmable, allowing any desired vector, opcode, instruction or other data to be entered. The Am9519A transfers the interrupt response information for the highest priority unmasked interrupt from the memory to the data bus when the $\overline{IACK}$ input is active.

Auto Clear Register: The 8-bit Auto Clear register contains one bit for each IREQ input and specifies the operating mode for each of the ISR bits. When an auto clear bit is off, the corresponding ISR bit is set when that interrupt is acknowledged and is cleared by software command. When an auto clear bit is on, the corresponding ISR bit is cleared by the hardware by the rising edge of the last acknowledge pulse. A reset function clears all auto clear bits.

Status Register: The 8-bit Status register contains information concerning the internal state of the chip. It is especially useful when operating in the polled mode to identify interrupting devices. Figure 1 shows the status register bit assignments. The polarity of the GINT bit 7 is not affected by the GINT polarity control. Bits S0-S2 are set asynchronously to a status register read operation. It is recommended to read the register twice and to compare the binary vectors for equality prior to proceeding with the device service in polled mode. The polarity of the GINT bit 7 is not affected by the GINT polarity control (Mode bit 3). The Status register is read by executing a read operation ($\overline{CS} = 0$, $RS = 0$) with the control location selected ($C/\overline{D} = 1$).

Mode Register: The 8-bit Mode register controls the operating options of the Am9519A. Figure 2 shows the bit assignments for the Mode register. The five low order mode bits (0 through 4) are loaded in parallel by command. Bits 5, 6 and 7 are controlled by separate commands. (See Figure 4.) The Mode register cannot be read out directly to the data bus, but Mode bits 0, 2 and 7 are available as part of the Status register.

Command Register: The 8-bit Command register stores the last command entered. Depending upon the command opcode, it may initiate internal actions or precondition the part for subsequent data bus transfers. The Command register is loaded by executing a write operation ($\overline{WR} = 0$) with the control location selected ($C/\overline{D} = 1$), as shown in Figure 3.

Byte Count Register: The length in bytes of the response associated with each interrupt is independently programmed so that different interrupts may have different length responses. The byte count for each response is stored in eight 2-bit Byte Count registers. For a given interrupt, the Am9519A will expect to receive a number of $\overline{IACK}$ pulses that equal the corresponding byte count and will hold $\overline{RIP}$ LOW until the count is satisfied.

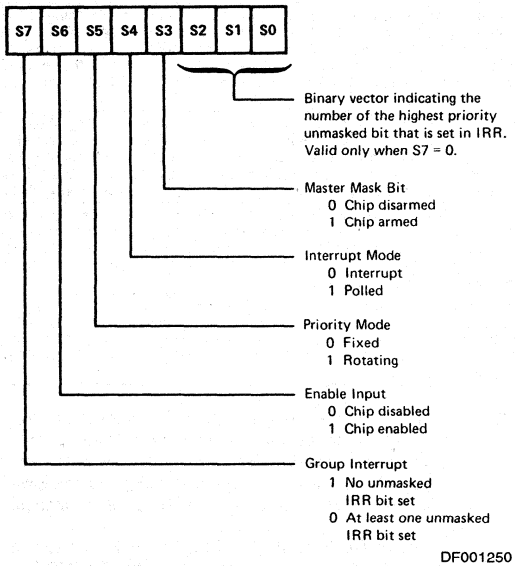


Figure 1. Status Register Bit Assignments

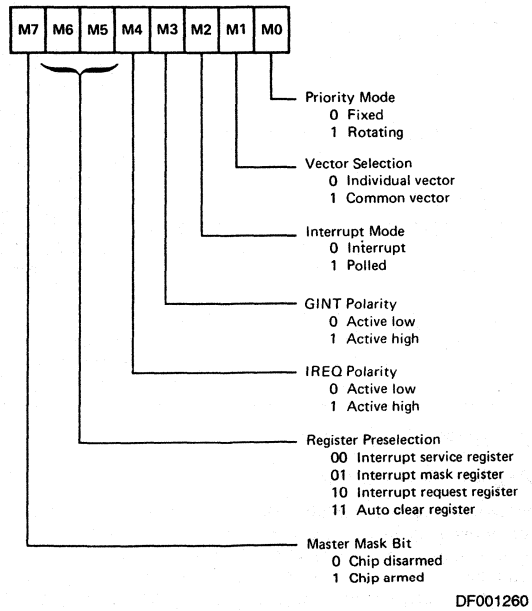


Figure 2. Mode Register Bit Assignments

DETAILED DESCRIPTION

Interrupts are used to improve system throughput and response time by eliminating heavy dependence on software polling procedures. Interrupts allow external devices to asynchronously modify the instruction sequence of a program being executed. In systems with multiple interrupts, vectoring can further improve performance by allowing direct identification of the interrupting device and its associated service routine. The Am9519A Universal Interrupt Controller contains, on one chip, all of the circuitry necessary to detect, prioritize and manage eight vectored interrupts. It includes many

options and operating modes that permit the design of sophisticated interrupt systems.

Reset

The reset function is accomplished by software command or automatically during power-up. The reset command may be issued by the CPU at any time. Internal power-up circuitry is triggered when VCC reaches a predetermined threshold, causing a brief internal reset pulse. In both cases, the resulting internal state of the machine is that all registers are cleared except the Mask register which is set. Thus, no Group Interrupt will be generated, and no interrupt requests will be recognized. The response memory and Byte Count registers are not affected by reset. Their contents after power-up are unpredictable and must be established by the host CPU during initialization.

Operating Sequence

A brief description of a typical sequence of events in an operating interrupt system will illustrate the general interactions among the host CPU, the interrupt controller and the interrupting peripheral.

1. The Am9519A controller is initialized by the CPU to customize its configuration and operation for the application at hand. Both the controller and the CPU are then enabled to accept interrupts.
2. One (or more) of the interrupt request inputs to the controller becomes active indicating that peripheral equipment is asking for service. The controller asynchronously accepts and latches the request(s).
3. If the request is masked, no further action takes place. If the request is not masked, a Group Interrupt output is generated by the controller.
4. The GINT signal is recognized by the CPU which normally will complete the execution of the current instruction, insert an interrupt acknowledge sequence into its instruction execution stream, and disable its internal interrupt structure. The controller expects to receive one or more $\overline{IACK}$ signals from the CPU during the acknowledge sequence.
5. When the controller receives the $\overline{IACK}$ signal, it brings $\overline{PAUSE}$ low and selects the highest priority unmasked pending request. When selection is complete, the $\overline{RIP}$ output is brought low and the first byte in the response memory associated with the selected request is output on the data bus. $\overline{PAUSE}$ stays low until RIP goes low. $\overline{RIP}$ stays low until the last byte of the response has been transferred.
6. During the acknowledge sequence, the IRR bit corresponding to the selected request is automatically cleared, and the corresponding ISR bit is set by the falling edge of $\overline{IACK}$. When the ISR bit is set, the Group Interrupt output is disabled until a higher priority request arrives or the ISR bit is cleared. The ISR bit will be cleared by either hardware or software.
7. If a higher priority request arrives while the current request is being serviced, GINT will be output by the controller, but will be recognized and acknowledged only if the CPU has its interrupt input enabled. If acknowledged, the corresponding higher priority ISR bit will be set and the requests nested.

Information Transfers

Figure 3 shows the control signal configurations for all information transfer operations between the Am9519A and the data bus. The following conventions are assumed: $\overline{RD}$ and $\overline{WR}$ active are mutually exclusive; $\overline{RD}$, $\overline{WR}$ and C/$\overline{D}$ have no meaning unless $\overline{CS}$ is LOW; active $\overline{IACK}$ pulses occur only when $\overline{CS}$ is HIGH.

For reading, the Status register is selected directly by the C/$\overline{D}$ control input. Other internal registers are read by preselecting the desired register with mode bits 5 and 6, and then executing a data read. The response memory can be read only with $\overline{IACK}$ pulses. For writing, the Command register is selected directly by the C/$\overline{D}$ control input. The Mask and Auto Clear registers are loaded following specific commands to that effect. To load each level of the response memory, the response preselect command is issued to select the desired level. An appropriate number of data write operations are then executed to load that level.

CONTROL INPUT					DATA BUS OPERATION
$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{IACK}$	
0	0	0	1	1	Transfer contents of preselected data register to data bus
0	0	1	0	1	Transfer contents of data bus to preselected data register
0	1	1	0	1	Transfer contents of status register to data bus
0	1	1	0	1	Transfer contents of data bus to command register
1	X	X	X	0	Transfer contents of selected response memory location to data bus
1	X	X	X	1	No information transferred

Figure 3. Summary of Data Bus Transfers

The Pause output may be used by the host CPU to ensure that propertiming relationships are maintained with the Am9519A when $\overline{IACK}$ is active. The $\overline{IACK}$ pulse width required depends on several variables, including: operating temperature, internal logic delays, number of interrupt controllers chained together, and the priority level of the interrupt being acknowledged. When delays in these variables combine to delay selection of a request following the falling edge of the first $\overline{IACK}$, the Pause output may be used to extend the $\overline{IACK}$ pulse, if necessary. Pause will remain LOW until a request has been selected, as indicated by the falling edge of $\overline{RIP}$. Typically, the internal interrupt selection process is quite fast, especially for systems with a single Am9519A, and Pause will consequently remain LOW for only a very brief interval and will not cause extension of the $\overline{IACK}$ timing.

Operating Options

The Mode register specifies the various combinations of operating options that may be selected by the CPU. It is cleared by power-up or by a reset command. Mode bit 0 specifies the rotating/fixed priority mode (see Figure 2). In the fixed mode, priority is assigned to the request inputs based upon their physical location at the chip interface, with IREQ0 the highest and IREQ7 the lowest. In the rotating mode, relative priority is the same as for the fixed mode and the most recently serviced request is assigned the lowest priority. In the fixed mode, a lower priority request might never receive service if enough higher priority requests are active. In the rotating mode, any request will receive service within a maximum of seven other service cycles no matter what pattern the request inputs follow.

Mode bit 1 selects the individual/common vector option. Individual vectoring provides a unique location in the response memory for each interrupt request. The common vector option always supplies the response associated with IREQ0 no matter which request is being acknowledged.

Mode bit 2 specifies interrupt or polled operation. In the polled mode, the Group Interrupt output is disabled. The CPU may

read the Status register to determine if a request is pending. Since $\overline{IACK}$ pulses are not normally supplied in polled mode, the IRR bit is not automatically cleared, but may be cleared by command. With no $\overline{IACK}$ input, the ISR and the response memory are not used. An Am9519A in the polled mode has EI connected to EO so that in multichip interrupt systems the polled chip is functionally removed from the priority hierarchy.

Mode bit 3 specifies the sense of the GINT output. When active high polarity is selected, the output is a two-state configuration. For active low polarity, the output is open drain and requires an external pull-up resistor to provide the high logic level. The open drain output allows wired-or configurations with other similar output signals.

Mode bit 4 specifies the sense of the IREQ inputs. When active low polarity is selected, the IRR responds to falling edges on the request inputs. When active high is selected, the IRR responds to rising edges.

Mode bits 5 and 6 specify the register that will read on subsequent data read operations ($C/\overline{D}$ = 0, $\overline{RD}$ = 0). This preselection remains valid until changed by a reset or a command.

Mode bit 7 is the master mask bit that disables all request inputs. It is used to disable all interrupts without modifying the IMR so that the previous IMR contents are valid when interrupts are re-enabled. When the master mask bit is LOW, it causes the EO line to remain disabled (LOW). Thus, for multiple-chip interrupt systems, one master mask bit can disable the whole interrupt structure. Alternatively, portions of the structure may be disabled. The state of the master mask bit is available as bit S3 of the Status register.

Programming

After reset, the Am9519A must be initialized by the CPU to perform useful work. At a minimum, the master mask bit and at least one of the IMR bits should be enabled. If vectoring is to be used, the response memory must be loaded; if not, the mode must be changed to a non-vectored configuration. Normally, the first step will be to modify the Mode register and the Auto clear register to establish the configuration desired for the application. Then the response memory and byte count will be loaded for those request levels that will be in use. The response memory for every channel must be written even if the channel is not used. Every byte need not be written, only those specified by the byte count. Finally, the master mask bit and at least portions of the IMR will be enabled to allow interrupt processing to proceed.

Commands

The host CPU configures, changes and inspects the internal condition of the Am9519A using the set of commands shown in Figure 4. An "X" entry in the table indicates a "don't care" state. All commands are entered by directly loading the Command register as shown in Figure 3 ($C/\overline{D}$ = 1, $\overline{WR}$ = 0). Figure 5 shows the coding assignments for the Byte Count registers. (A detailed description of each command is contained in the Am9519A Application Note AMPUB-071.)

BY1	BY0	COUNT
0	0	1
0	1	2
1	0	3
1	1	4

Figure 5. Byte Count Coding

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Reset
0	0	0	1	0	X	X	X	Clear all IRR and all IMR bits
0	0	0	1	1	B2	B1	B0	Clear IRR and IMR bit specified by B2, B1, B0
0	0	1	0	0	X	X	X	Clear all IMR bits
0	0	1	0	1	B2	B1	B0	Clear IMR bit specified by B2, B1, B0
0	0	1	1	0	X	X	X	Set all IMR bits
0	0	1	1	1	B2	B1	B0	Set IMR bit specified by B2, B1, B0
0	1	0	0	0	X	X	X	Clear all IRR bits
0	1	0	0	1	B2	B1	B0	Clear IRR bit specified by B2, B1, B0
0	1	0	1	0	X	X	X	Set all IRR bits
0	1	0	1	1	B2	B1	B0	Set IRR bit specified by B2, B1, B0
0	1	1	0	X	X	X	X	Clear highest priority ISR bit
0	1	1	1	0	X	X	X	Clear all ISR bits
0	1	1	1	1	B2	B1	B0	Clear ISR bit specified by B2, B1, B0
1	0	0	M4	M3	M2	M1	M0	Load Mode register bits 0 – 4 with specified pattern
1	0	1	0	M6	M5	0	0	Load Mode register bits 5, 6 with specified pattern
1	0	1	0	M6	M5	0	1	Load Mode register bits 5, 6 and set mode bit 7
1	0	1	0	M6	M5	1	0	Load Mode register bits 5, 6 and clear mode bit 7
1	0	1	1	X	X	X	X	Preselected IMR for subsequent loading from data bus
1	1	0	0	X	X	X	X	Preselected Auto Clear register for subsequent loading from data bus
1	1	1	BY1	BY0	L2	L1	L0	Load BY1, BY0 into byte count register and preselect response memory level specified by L2, L1, L0 for subsequent loading from data bus

Figure 4. Am9519A Command Summary

APPLICATIONS

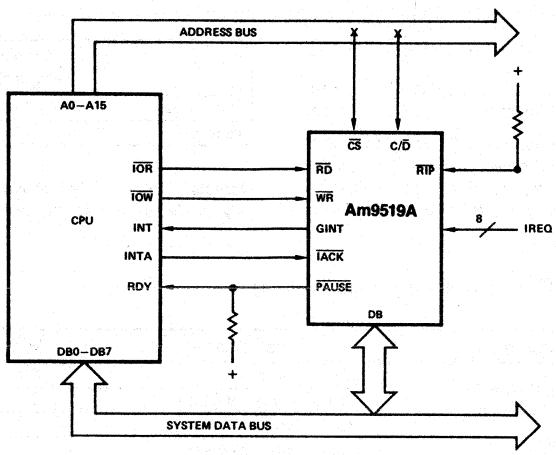


Figure 6. Base Interrupt System Configuration

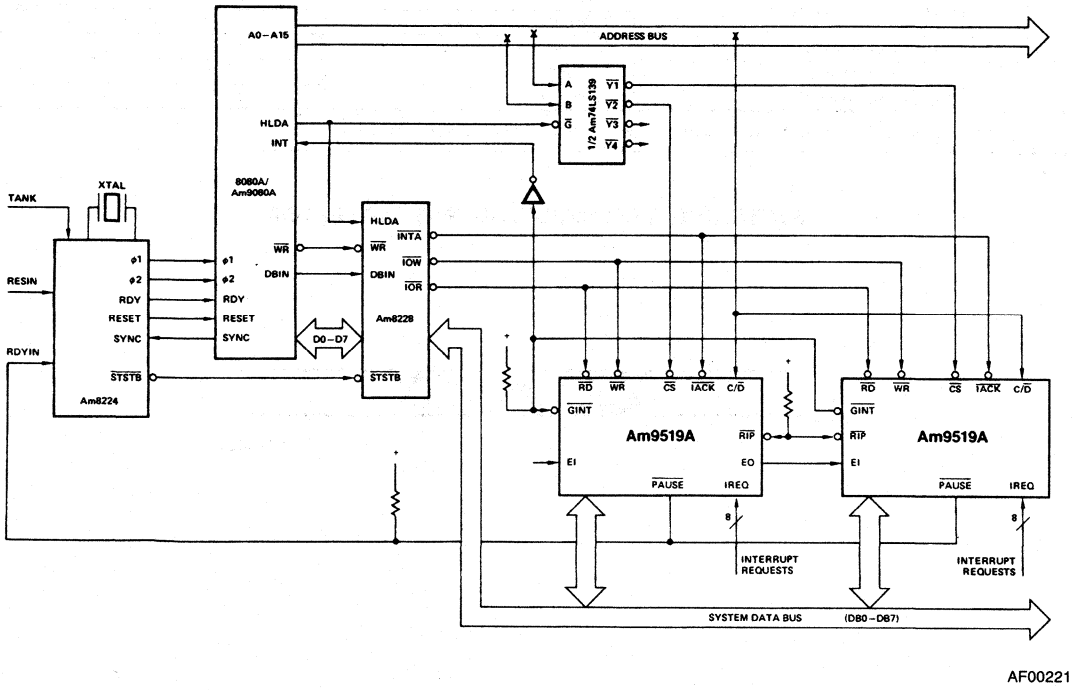


Figure 7. Expanded Interrupt System Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
VCC with Respect to VSS ..... −0.5V to +7.0V
All Signal Voltages
with Respect to VSS ..... −0.5V to +7.0V
Power Dissipitation (Package Limitation) 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	Ambient Temperature	V_{CC}	V_{SS}
Commercial	$0°C \leq T_A \leq 70°C$	5.0V ±5%	0V
Industrial	$-40°C \leq T_A \leq 85°C$	5.0V ±10%	0V

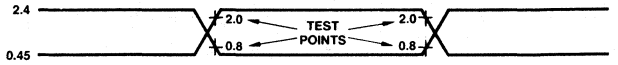
Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

Parameters	Description	Test Conditions		Min	Typ	Max	Units
VOH	Output High Voltage (Note 12)	IOH = −200μA		2.4			Volts
		IOH = −100μA (EO only)		2.4			
VOL	Output Low Voltage	IOL = 3.2mA				0.4	Volts
		IOL = 1.0mA (EO only)				0.4	
VIH	Input High Voltage			2.0		VCC	Volts
VIL	Input Low Voltage			−0.5		0.8	Volts
IIX	Input Load Current	VSS ≤ VIN ≤ VCC	EI Input	−60		10	μA
			Other Inputs	−10		10	
IOZ	Output Leakage Current	VSS ≤ VOUT ≤ VCC, Output Off		−10		10	μA
ICC	VCC Supply Current	T_A = +25°C			80	125	mA
		T_A = 0°C			100	145	
CO	Output Capacitance	fc = 1.0MHz				15	pF
CI	Input Capacitance	T_A = 25°C				10	
CIO	I/O Capacitance	All pins at 0V				20	

Note: 1. Typical values for T_A = 25°C, nominal supply voltage and nominal processing parameters.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007820

00147B

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING CHARACTERISTICS over operating range (Notes 2, 3, 4, 5)

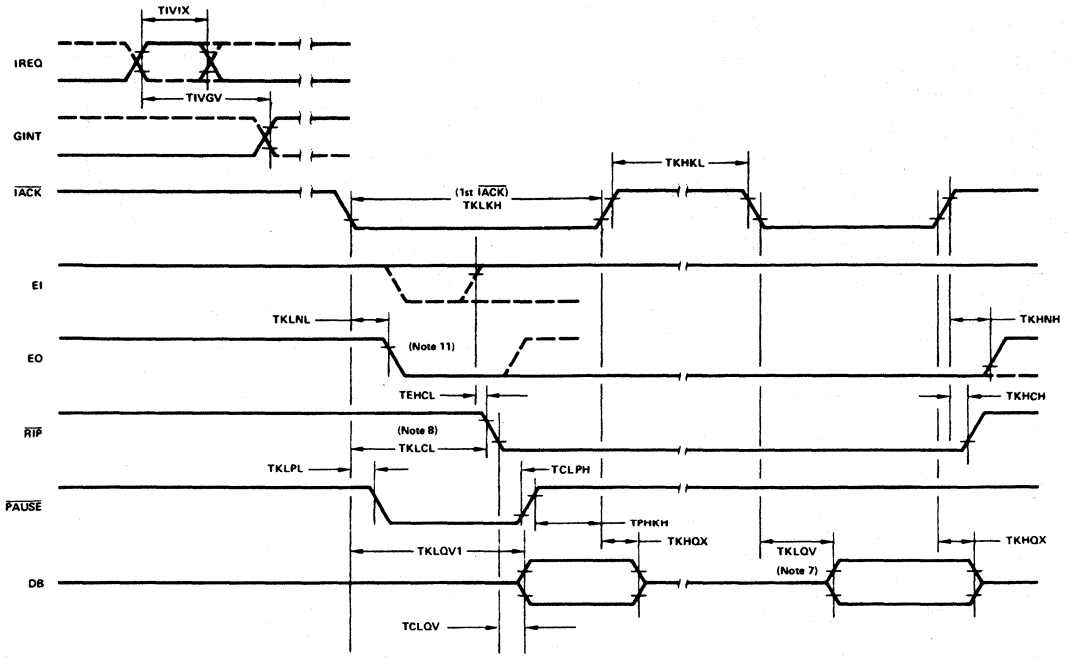
Parameters	Description		Am9519A		Am9519A-1		Units
			Min	Max	Min	Max	
TAVRL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Read LOW		0		0		ns
TAVWL	C/$\overline{D}$ Valid and $\overline{CS}$ LOW to Write LOW		0		0		ns
TCLPH	$\overline{RIP}$ LOW to $\overline{PAUSE}$ HIGH (Note 6)		75	375	75	375	ns
TCLQV	$\overline{RIP}$ LOW to Data Out Valid (Note 7)			50		40	ns
TDVWH	Data In Valid to Write HIGH		250		200		ns
TEHCL	Enable In HIGH to $\overline{RIP}$ LOW (Notes 8, 9)		30	300	30	300	ns
TIVGV	Interrupt Request Valid to Group Interrupt Valid		100	800		650	ns
TIVIX	Interrupt Request Valid to Interrupt Request Don't Care (IREQ Pulse Duration)		250		250		ns
TKHCH	$\overline{IACK}$ HIGH to $\overline{RIP}$ HIGH (Note 8)			450		350	ns
TKHKL	$\overline{IACK}$ HIGH to $\overline{IACK}$ LOW ($\overline{IACK}$ Recovery)		120		100		ns
TKHNH	$\overline{IACK}$ HIGH to EO HIGH (Notes 10, 11)			975		750	ns
TKHQX	$\overline{IACK}$ HIGH to Data Out Invalid		20	200	20	100	ns
TKLCL	$\overline{IACK}$ LOW to $\overline{RIP}$ LOW (Notes 8, 13)	COM'L	75	600	75	450	ns
		IND	75	600			ns
TKLKH	$\overline{IACK}$ LOW to $\overline{IACK}$ HIGH (1st $\overline{IACK}$) (Note 13)		975		800		ns
TKLNL	$\overline{IACK}$ LOW to EO LOW (Notes 10, 11, 13)			125		100	ns
TKLPL	$\overline{IACK}$ LOW to $\overline{PAUSE}$ LOW (Note 13)		25	175	25	125	ns
TKLQV	$\overline{IACK}$ LOW to Data Out Valid (Notes 7, 13)		25	300	25	200	ns
TKLQV1	1st $\overline{IACK}$ LOW to Data Out Valid (Note 13)		75	650	75	490	ns
TPHKH	$\overline{PAUSE}$ HIGH to $\overline{IACK}$ HIGH		0		0		ns
TRHAX	Read HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care		0		0		ns
TRHQX	Read HIGH to Data Out Invalid		20	200	20	100	ns
TRLQV	Read LOW to Data Out Valid			300		200	ns
TRLQX	Read LOW to Data Out Unknown		50		50		ns
TRLRH	Read LOW to Read HIGH ($\overline{RD}$ Pulse Duration)		300		250		ns
TWHAX	Write HIGH to C/$\overline{D}$ and $\overline{CS}$ Don't Care		25		25		ns
TWHDX	Write HIGH to Data in Don't Care		25		25		ns
TWHRW	Write HIGH to Read or Write LOW (Write Recovery)		600		400		ns
TWLWH	Write LOW to Write HIGH ($\overline{WR}$ Pulse Duration)		300		250		ns
TKHIH	$\overline{IACK}$ HIGH to GINT inactive			1000		800	ns

Notes:
1. Typical values for $T_A = 25°C$, nominal supply voltage and nominal processing parameters.
2. Test conditions assume transition times of 20ns or less, timing reference levels of 0.8V and 2.0V and output loading of one TTL gate plus 100pF, unless otherwise noted.
3. Transition abbreviations used for the switching parameter symbols include: H = HIGH, L = LOW, V = Valid, X = unknown or don't care, Z = high-impedance.
4. Signal abbreviations used for the switching parameter symbols include: R = Read, W = Write, Q = Data Out, D = Data In, A = Address (CS and C/D), K = Interrupt Acknowledge, N = Enable Out, E = Enable In, P = Pause, C = $\overline{RIP}$.
5. Switching parameters are listed in alphabetical order.
6. During the first $\overline{IACK}$ pulse, $\overline{PAUSE}$ will be LOW long enough to allow for priority resolution and will not go HIGH until after RIP goes LOW (TCLPH).
7. TKLQV applies only to second, third and fourth $\overline{IACK}$ pulses while $\overline{RIP}$ is LOW. During the first $\overline{IACK}$ pulse, Data Out will be valid following the falling edge of $\overline{RIP}$ (TCLQV).
8. $\overline{RIP}$ is pulled LOW to indicate that an interrupt request has been selected. $\overline{RIP}$ cannot be pulled LOW until EI is HIGH following an internal delay. TKLCL will govern the falling edge of $\overline{RIP}$ when EI is always HIGH or is HIGH early in the acknowledge cycle. The TEHCL will govern when EI goes HIGH later in the cycle. The rising edge of EI will be determined by the length of the preceding priority resolution chain.
   $\overline{RIP}$ remains LOW until after the rising edge of the $\overline{IACK}$ pulse that transfers the last response byte for the selected IREQ.
9. Test conditions for the EI line assume timing reference levels of 0.8V and 2.0V with transition times of 10ns or less.
10. Test conditions for the EO line assume output loading of two LS TTL gates plus 30pF and timing reference levels of 0.8V or 2.0V. Since EO normally only drives EI of another Am9519A, higher speed operation can be specified with this more realistic test condition.
11. The arrival of $\overline{IACK}$ will cause EO to go LOW, disabling additional circuits that may be connected to EO. If no valid interrupt is pending, EO will return HIGH when EI is HIGH. If a pending request is selected, EO will stay LOW until after the last $\overline{IACK}$ pulse for that interrupt is complete and $\overline{RIP}$ goes HIGH.
12. VOH specifications do not apply to $\overline{RIP}$ or to $\overline{GINT}$ when active-low. These outputs are open drain, and VOH levels will be determined by external circuitry.
13. $\overline{CS}$ must be HIGH for at least 100ns prior to $\overline{IACK}$ going LOW.

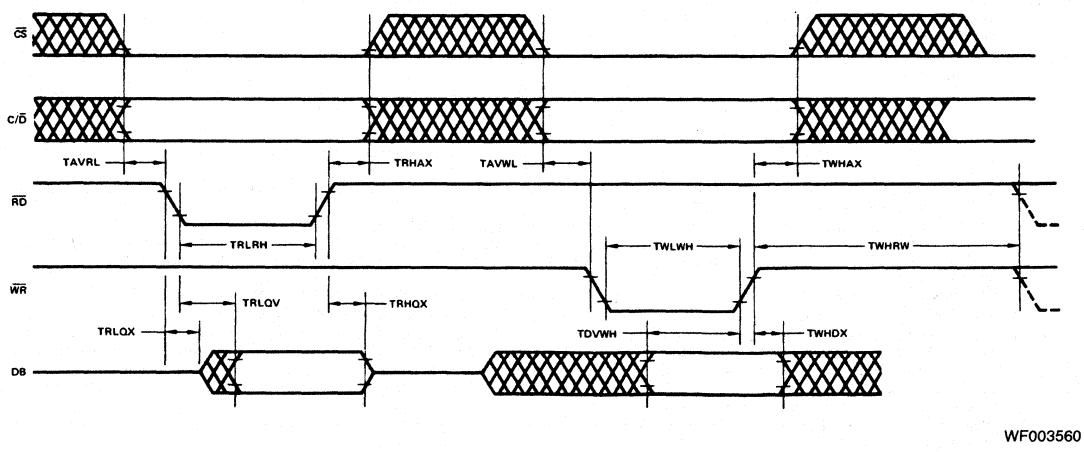
00147B

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING WAVEFORMS



Interrupt Operations



Data Bus Transfers

Am9520/Am9521/AmZ8065

Burst Error Processor

DISTINCTIVE CHARACTERISTICS

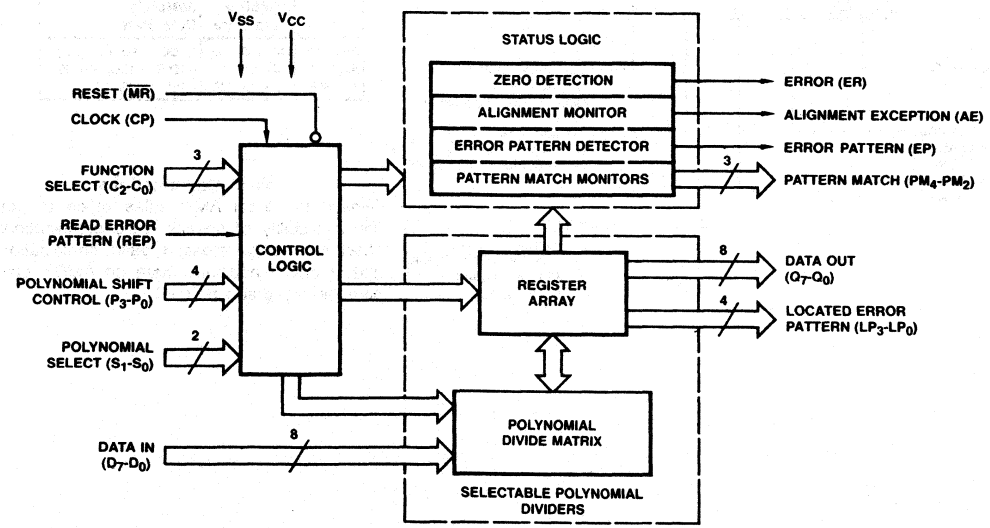
- **Provides for detection and correction of burst errors**
  Detects errors in serial data up to 585K bits long. Allows correction of error bursts of up to 12 bits.
- **High-Speed Operation**
  Effective data rates up to 20 Mbits/second for Am9520/Am9521/AmZ8065 and 30 Mbits/second for -1 versions. Fast enough for high-performance hard and soft disk systems.
- **Selectable Industry-Standard Polynomials**
  35-bit and 32-bit polynomials on Am9521. Am9520/AmZ8065 additionally has popular IBM 56-bit and 48-bit versions.
- **Three correction algorithms provide flexibility**
  !q j Full-period clock-around method for conforming to current practices. Chinese remainder theorem reduces correction time by orders of magnitude. Reciprocal polynomial makes correction possible with 48-bit code.
- **Designed for use in both microprogrammed and microprocessor disk controller systems**
  Device complements both AmZ8000 and Am2900 microprocessor families and can also be used with other microprocessors.

GENERAL DESCRIPTION

The Burst Error Processor (BEP) provides for error detection and correction for high-performance disk systems and other systems in which high-speed serial data transfer takes place. As data density and transfer rates increase in both hard and floppy disks and other storage media, error detection and correction become increasingly important. The BEP is an LSI circuit that facilitates the most common error detection and correction schemes accommodating data streams of up to 585K bits at up to 20M bits/second effective data rate.

The BEP provides a choice of four standard polynomials, including the popular 56-bit and 48-bit versions, to satisfy a broad range of applications. The device divides the data stream by the selected polynomial using the rules of algebra in polynomial fields. The resulting remainder is the check word, which is then appended to the data for writing on the disk as a record. When the record is read back, the BEP computes the syndrome for data validation. If an error is detected, the location and pattern of this burst in the data stream is determined for corrections.

BLOCK DIAGRAM

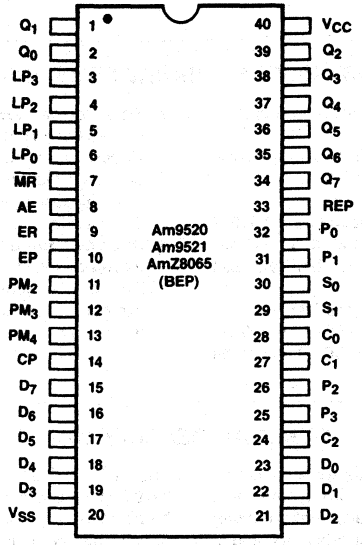


00617B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40

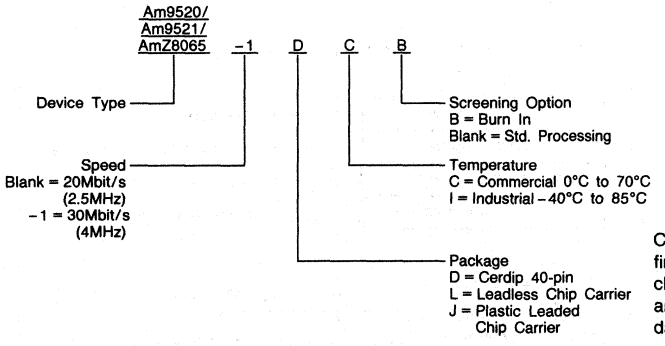


CD005121

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations		
	20Mbit/s Data Rate	30Mbit/s Data Rate
Am9520/ Am9521/ AmZ8065/	DC, DCB, DI, DIB, LC, LCB, LI, LIB	-1DC, -1DCB, -1DI, -1DIB, -1LC, -1LCB, -1LI, -1LIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

00617B

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION

Pin No.	Name	I/O	Description
40	V_{CC}		+5V Power Supply.
20	V_{SS}		Ground.
30,29	S_0-S_1	I	Polynomial Select. Logic levels on these two inputs select one of the four standard polynomials provided in the Am9520. The following table specifies the polynomial select codes.

POLYNOMIAL SELECT CODES

S_1	S_0	Polynomial	Number of Check Bits
L	L	$(X^{22}+1) \bullet (X^{11}+X^7+X^6+X+1) \bullet (X^{12}+X^{11}+X^{10}+ \ldots +X+1) \bullet (X^{11}+X^9+X^7+X^6+X^5+X+1)$	56
L	H	$(X^{21}+1) \bullet (X^{11}+X^2+1)$	32
H	L	$(X^{23}+1) \bullet (X^{12}+X^{11}+X^8+X^7+X^3+X+1)$	35
H	H	$(X^{13}+1) \bullet (X^{35}+X^{23}+X^8+X^2+1)$	48

23,22 21,19 18,17 16,15	D_0-D_7	I	Data In. These eight inputs are used for entering information. D_0 is the least significant bit, and D_7 is the most significant bit position. HIGH on any input corresponds to 1, and LOW represents 0. Data entry occurs on the LOW-to-HIGH transition of the CP input. Any change on the D_0-D_7 inputs must take place only when the CP input is HIGH. See Timing diagram for details on set-up and hold time specifications.
28,27 24	C_0-C_2	I	Function Select. These three inputs specify the desired function according to the following table. Detailed description of each function is found in later sections of this document. Any change on the C_0-C_2 inputs must take place only when the CP input is HIGH. See Timing diagram for set-up and hold time specifications.

FUNCTION SELECT CODES

C_2	C_1	C_0	Function
L	L	L	Compute check bits
L	L	H	Write check bits
L	H	L	Read normal
L	H	H	Read high speed
H	L	L	Load
H	L	H	Reserved
H	H	L	Correct normal (Full period clock around)
H	H	H	Correct high speed (Chinese remainder theorem method)

14	CP	I	Clock. Operations are controlled by this input. Outputs become valid after the LOW-to-HIGH transition on the CP input. The quiescent state of the CP input is HIGH. Any changes on the data and control inputs must take place only when the CP input is HIGH. See Timing diagrams for exact specifications. During operations, it may be required to stop the clock; the clock must be stopped in the HIGH state only. Also, note that requirements on the CP input during correction operations differ from those during other functions. See Timing diagram for details.
7	$\overline{MR}$	I	Master Reset. LOW on this input initializes the device. This input must remain LOW for a specified time to accomplish initialization before returning to the quiescent HIGH state. In general, the devices require initialization prior to performing Compute Check Bits, Read Normal, Read High Speed and Load functions.
2,1,39 38,37 36,35 34	Q_0-Q_7	O	Data Out (3-State). The check bits are made available on these eight outputs one byte at a time. Q_0 is the least significant bit position, and Q_7 is the most significant. HIGH on these outputs represents 1 and LOW 0. The Q_0-Q_7 are active only during the following conditions: a) The C_0-C_2 inputs specify Write Check Bits Function. b) The REP input is HIGH. During all other conditions Q_0-Q_7 are in a high-impedance state.
6,5 4,3	LP_0-LP_3	O	Located Error Pattern (3-state). The LP_0-LP_3 outputs together with the Q_0-Q_7 outputs provide the 12-bit error pattern in which Q_7 is the most significant bit and LP_0 is the least significant bit position. HIGH represents 1, and LOW represents 0. The REP input must be HIGH to read the error pattern. If the REP input is LOW, the LP_0-LP_3 outputs are in the high-impedance state.
33	REP	I	Read Error Pattern. A HIGH on this input activates the LP_0-LP_3 and Q_0-Q_7 outputs. This error pattern information is valid only after a HIGH is indicated on the EP output during correction operations.
9	ER	O	Error. HIGH on this output indicates that the BEP has detected an error. This output must be considered valid only after the last check byte during Read Normal or Read High Speed functions has been entered. The resulting syndrome is then contained in the register array. A non-zero syndrome indicates error; zero syndrome indicates no error. The ER output always reflects the state of this register array (zero or non-zero). The ER output is LOW after initialization.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
10	EP	O	Error Pattern. HIGH on this output indicates that the error pattern has been found during the correction process. When the last check byte was entered during a Read function, the resulting syndrome is contained in the register array. The error pattern information is buried in this syndrome. To extract the error pattern, the BEP is clocked while the appropriate (Correct Normal or Correct High Speed) code is applied to the C_0-C_2 inputs until EP goes HIGH. The number of clocks required to find the error pattern is used to calucate where in the data stream the error has occurred. The EP output will be LOW after initialization by the $\overline{MR}$ input. The EP output is valid only during the correction operations and must be ignored at all other times. See Correct Normal and Correct High Speed under Detailed Description for further details.
11,12,13	PM_2-PM_4	O	Pattern Match. When using the Chinese remainder theorem for error correction, information is loaded into several feedback shift registers simultaneously. The number of registers is equal to the number of factors of the polynomial. After a high speed operation, there are as many syndromes as there are factors. For correction, the register corresponding to the first factor must be shifted until the EP output indicates HIGH. Then each register corresponding to the remaining factors must be shifted until a match occurs in each register with the error pattern contained in the first register. HIGH on PM_2, PM_3 or PM_4 outputs indicates that corresponding registers match. The PM_2 corresponds to the second factor, PM_3 corresponds to the third and PM_4 corresponds to the fourth factor. PM_3 corresponds to the third and PM_4 corresponds to the fourth factor. If a polynomial has only two factors, then PM_3 and PM_4 outputs have no meaning. Indications on the PM_2-PM_4 outputs must be considered valid only during high-speed correct function and should be ignored at all other times.
6,5,4,3	P_0-P_3	I	Polynomial Shift Control. Correction procedure using the Chinese remainder theorem method requires that each syndrome obtained from the High-Speed Read function be shifted individually. The P_0-P_3 inputs provide this capability: P_0 corresponds to the first factor, P_1 corresponds to the second factor and so on. HIGH on an input allows the corresponding register to shift and LOW causes it to hold. These inputs have an effect only during the Correct High-Speed function. Any change on these inputs must occur only when the CP input is HIGH.
8	AE	O	Alignment Exception. The devices use an 8-bit parallel mechanization of the feedback shift register configurations. Under certain conditions, the error pattern will not, therefore, automatically line up in predetermined positions of the register array during the correction operations. HIGH on the AE output indicates that such a condition is detected. The Am9520 automatically switches into the one-bit shift mode. The number of clocks for which the AE output is HIGH is used in the error location calculation. See Detailed Description for further details.

00617B

Refer to page 7-1 for Essential Information on Military Devices

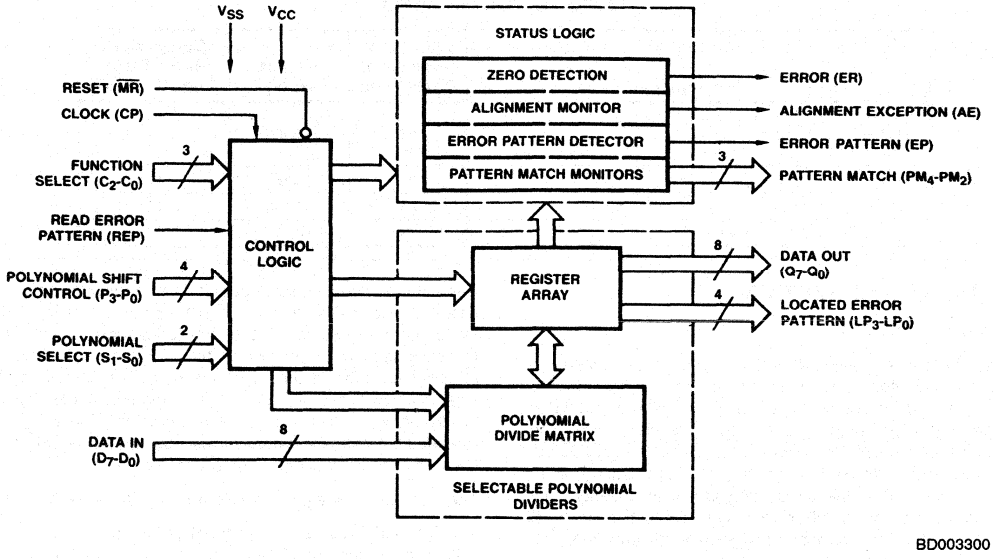


Figure 1. Am9520/Am9521/AmZ8065 Burst Error Processor

ARCHITECTURE

Figure 1 is a conceptual block diagram. It consists of four major sections – Register Array, Polynomial Divide Matrix, Status Logic and Control Logic.

Register Array

This section consists of 56 flip-flops used for check bit computation during write operation, syndrome computation during read operation and error pattern extraction during error correction operation. In general, the Polynomial Divide Matrix provides the bit patterns required for the Register Array. The combination of Register Array and Polynomial Divide Matrix mechanizes the familiar serial form of feedback shift register arrangement in an 8-bit parallel form. The Q_0–Q_7 outputs of the Am9520 are obtained from the Register Array. When correction operations are complete, the error pattern is available on 12 outputs: eight bits on the Q_0–Q_7 outputs and the remaining four bits on the LP_0–LP_3 outputs. The Read Error Pattern (REP) input must be HIGH for the error pattern to be available. The Control Logic generates Clock signals for the Register Array.

Status Logic

This section monitors the register arrays to generate the various error detection outputs of the BEP, including ER, AE, PM_2, PM_3, PM_4 and EP.

Polynomial Divide Matrix

Polynomial Divide Matrix is the heart of the BEP. The Control Logic decodes the Polynomial Select (S_0–S_1) and Function Select (C_0–C_2) inputs to generate the necessary gating signals to the matrix. The matrix establishes connections such that a byte of data presented on the D_0–D_7 inputs will be suitably divided by the selected generator polynomial. Four different polynomials are selected by logic levels on the S_0–S_1 inputs (Table 1).

These devices can be used in three fundamentally different types of operations: write, read and correct. The various functions are selected by the C_0–C_2 control inputs.

Write

While data is being written on the disk, the BEP is in the Compute Check Bits mode looking at the data bytes without affecting the flow of data to the disk. After the last data byte, the BEP is switched into the Write Check Bits function, outputting the 4, 5, 6 or 7 check bytes. This is the additional information appended to the data stream that allows the detection and correction of possible read errors.

Read

When information (data plus appended check bits) is being read, the BEP must be in either Read Normal mode or Read High Speed mode. These modes differ only in the correction algorithm that will be used if an error has occurred. In both modes parallel bytes are read into the device. After the last information byte has been entered, the ER output is checked. If it is LOW, there is no error; if it is HIGH, there is an error.

Correction

After the read operation, the syndrome held in the Register Array contains all the information necessary to find the error location and the error pattern, i.e., to allow error correction. In the Correct Normal mode, the error location is found by counting the number of clock pulses required to make the EP output go HIGH. The error pattern is then available on the LP_0–LP_3 and Q_0–Q_7 outputs and can be used to Exclusive OR with data.

In Correct High Speed mode, the error location is also found by counting clock pulses, but they are routed in succession to the different sections of the Register Array. This results in slightly more complicated but substantially faster operation.

TABLE 1. POLYNOMIALS

Polynominal	Number of Check Bits	Period (Bits)	Correctable Burst Error Length (Bits)
$(X^{22}+1)\cdot(X^{11}+7^7+X^6+X+1)\cdot$ $(X^{12}+X^{11}+X^{10}+\ldots+X+1)\cdot$ $(X^{11}+X^9+X^7+X^6+X^5+X+1)$	56	585,442	11
$(X^{21}+1)\cdot(X^{11}+X^2+1)$	32	42,987	11
$(X^{23}+1)\cdot(X^{12}+X^{11}+X^8+X^7+X^3+X+1)$	35	94,185	12
$(X^{13}+1)\cdot(X^{35}+X^{23}+X^8+X^2+1)$	48	$13\cdot(2^{35}-1)$	7

DETAILED DESCRIPTION

Compute Check Bits

The check bits to be appended to the data are computed using this function. The S_0–S_1 inputs select the desired polynomial. The Polynomial Matrix will be configured such that the generator polynomial is in the expanded form. The expanded form of a polynomial is obtained by multiplying out its factors and combining proper terms using modulo-2 arithmetic. Assume that the 32-bit polynomial is selected: the factored form of the 32-bit polynomial in Table 1 is $(X^{21}+1)(X^{11}+X^2+1)$. The corresponding expanded form is $X^{32}+X^{23}+X^{21}+X^{11}+X^2+1$.

The sequence of events to compute the check bits is as follows:

1. The CP input is in quiescent HIGH state.
2. Initialize by activating the $\overline{MR}$ input LOW and return it to HIGH.
3. Through appropriate logic levels on the S_0–S_1 inputs, specify the desired polynomial. Also, select Compute Check Bits code through the C_0–C_2 inputs.
4. Establish a byte of data on the D_0–D_7 inputs.
5. Make CP input LOW and then HIGH. See timing diagram for detailed timing specifications.
6. Keep repeating from step 4 until all data bytes are entered.

Write Check Bits

In Compute Check Bits mode, the polynomial matrix and the Register Array are mechanizing a feedback shift register configuration. However, when Write Check Bits Code is established on the C_0–C_2 inputs, the feedback paths are disabled such that the Register Array will behave as a simple shift register. When the last data byte is entered in the Compute Check Bits mode, the Register Array holds the check bits. These check bits will be available on the Q_0–Q_7 outputs, one byte at a time.

The sequence of events to obtain the check bits is as follows:

1. The CP is in quiescent HIGH state.
2. Establish appropriate code on the S_0–S_1 inputs. This code must be the same as that used for Compute Check Bits function.
3. Establish Write Check Bits code on the C_0–C_2 inputs.
4. After a propagation delay, the Q_0–Q_7 outputs will contain the first check byte.
5. Make CP input LOW and then HIGH. The next check byte will be available on the Q_0–Q_7 outputs.
6. Keep repeating from step 5 until all check bytes that correspond to the selected polynomial are read out.

Read Normal

Two methodologies are available for error correction with these devices: (a) Full period clock around (normal method) and (b) Chinese remainder theorem (high-speed method). The Read Normal function must be used for reading data from the disk if the normal method is used for error correction. When Read Normal is selected, the Polynomial Matrix establishes the polynomial in the expanded form. In this mode, the input stream consisting of data and check bytes is divided by the selected polynomial to obtain the syndrome. If the resulting syndrome is not zero, an error is detected. The ER output indicates whether the syndrome is zero or not. HIGH on the ER output indicates non-zero syndrome.

The sequence of events for Read Normal is as follows:

1. The CP input is in quiescent HIGH state.
2. Initialize the Am9520 by activating the $\overline{MR}$ input LOW and then return it to HIGH.
3. Establish proper code on the S_0–S_1 inputs. The polynomial selected for the read operation must be the same as the one originally used for generating the check bits.
4. Establish Read Normal code on the C_0–C_2 inputs.
5. Present a byte of information read from the disk on the D_0–D_7 inputs.
6. Make the CP input LOW and then HIGH.
7. Keep repeating from step 5 until the last check byte read from the disk is processed.
8. After entering last check byte, test the ER output. HIGH on this output is indicative of an error and LOW means no error detected.

Read High Speed

This function must be used for reading data if the Chinese remainder theorem method is to be used for error correction. In general, the Chinese remainder method accomplishes error correction in fewer clock cycles than the normal method. This method of correction, however, is not available for the 48-bit polynomial due to the nature of the factors that make up this polynomial. As explained later, the reciprocal polynomial technique is used for error correction when the 48-bit polynomial is selected.

The only difference between Read Normal and Read-High Speed Modes is as follows: in the Read Normal, the input stream is divided by the expanded version of the polynomial; whereas, in the Read High-Speed Mode, the input stream is simultaneously divided by all factors of the polynomial. Thus, the high-speed mode results in as many syndromes as the number of factors of the polynomial. If all syndromes are zero after entering the last check byte, the ER output will be LOW, indicating error-free operation. If there was an error, the ER will be HIGH.

The sequence of events in this mode are as follows:

1. The CP input is in its quiescent HIGH state.
2. Specify the polynomial on the S_0–S_1 input. This must obviously be the same polynomial that generated the check bits originally.
3. Specify Read High-Speed function on the C_0–C_2 inputs.

4. Initialize by activating the $\overline{MR}$ input LOW and then return it to HIGH.
5. Present a byte read from the disk on the D_0–D_7 inputs.
6. Make the CP input LOW and then HIGH.
7. Keep repeating from step 5 until all data and check bytes are entered.
8. Test the ER output after entering the last check byte. HIGH on this output is indicative of an error, and LOW signifies no error.

Correct Normal

The syndrome obtained from Read Normal operation is manipulated to extract the error pattern as well as its location using the Correct Normal function. Of the four polynomials listed in Table 1, the 48-bit version requires a separate explanation. For all cases except the 48-bit version, the polynomial is established in the expanded form.

In the Correct Normal, the syndrome is repeatedly divided by the polynomial until the error pattern is located. This division is accomplished by repeated clocking while ignoring the D_0–D_7 inputs. HIGH on the EP output signifies that the error pattern is found. The error pattern is always characterized by a known number of consecutive zeros at specified Register Array locations. The exact number of zeros and their location is a function of the select polynomial. The status logic detects this unique combination to generate the EP output. The number of clock cycles needed to locate the error pattern is a measure of the error location. If the number of clock cycles has exceeded the natural period of the selected polynomial without finding the error pattern, then an uncorrectable error has occurred. The AE output must also be considered in the Correct Normal mode of operation.

The polynomial matrix is an 8-bit parallel mechanization of the familiar serial polynomial division scheme. Because of this, there are certain conditions under which the error pattern will not line up automatically. The Status Logic also monitors this condition. When such an alignment exception is detected, the AE output of the device goes HIGH.

Internally, the device switches automatically into the one-bit shift mode. Let R_1 be the number of clock cycles for the AE output to go HIGH. Let R_2 be the number of clock cycles from the AE output going High to the EP output going HIGH. Let N be the natural period of the selected polynomial. Then $N \cdot K - 8R_1 - R_2$ is the first bit in the error burst counting from the last check bit of the record, where K is the smallest positive integer to make this expression positive. If there is no alignment exception, then $R_2 = 0$. See Table 1 for periods of the polynomials.

The error pattern provided is used externally to correct the error. The error pattern is available on the Q_0–Q_7 and LP_0–LP_3 outputs when the REP input is HIGH. Q_7 corresponds to the first bit in error. When an error pattern bit is HIGH, then the corresponding bit in the data stream must be complemented to accomplish correction.

The Correct Normal discussed so far can be summarized by the following sequence of events.

1. The CP input is in the quiescent HIGH state.
2. The ER output is HIGH, indicating error from the previous Read Normal operation.
3. Select appropriate polynomial (S_0–S_1).
4. Select Correct Normal Code (C_0–C_2).
5. Let R_1 and R_2 be two external counters both initialized to zero.
6. Check if the EP output is HIGH. If HIGH, the error pattern is found. The error location is given by $L = N \cdot K - (8R_1 + R_2)$, except for the 35-bit polynomial where $L = N \cdot K - (8R_1 + R_2 + 5)$. ($R_2$ is always zero for this case.)
7. If the EP output is LOW, test the AE output. If the AE output is HIGH, make the CP input LOW and then HIGH. Increment R_2. If the AE output is LOW, make the CP input LOW and then HIGH. Increment R_1 instead.
8. If $R_1 + R_2$ is greater than N (N is the natural period of the selected polynomial), then an uncorrectable error occurred. Abort the correction process.
9. If the error is correctable, repeat from step 6.

The flowchart in Figure 2 explains the correction process.

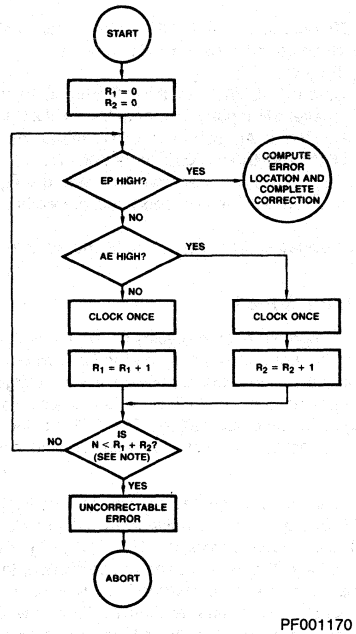


Figure 2. Flowchart

Note: For the 48-bit polynomial, N = number of actual data bits +48 check bits.

Now consider the Correct Normal Mode of operation with the 48-bit polynomial. The period of the 48-bit polynomial is so large that ordinary division is not practical. In this case the Polynomial Matrix establishes the reciprocal of the expanded polynomial. If G(X) is a polynomial of degree K, then its reciprocal $G^*(X) = X^K G(1/X)$. Because of this, the syndrome obtained using Read Normal Mode with the 48-bit polynomial is not used directly for extracting the error pattern and calculating its location. Instead, the reciprocal of the syndrome must be used.

The procedure for forming the reciprocal must be accomplished externally as follows. Assume that the Read Normal operation using the 48-bit polynomial was finished and an error was detected. Read out the syndrome using Write Check Bits function. Now reverse all these syndrome bits such that the previously most significant bit becomes the least significant bit and vice versa. The result is the reciprocal syndrome.

Now load this reciprocal syndrome into the device using the Load function (see description of Load). Once the reciprocal syndrome is loaded, Correct Normal function is established on the C_0–C_2 inputs, and the correction process can be started.

The actual correction process is exactly the same as before, except the error location in this case is given by $8R_1 + R_2 - 48$.

The sequence of events can be summarized as follows:

1. Read out the syndrome using the Write Check Bits function.
2. Form the reciprocal syndrome externally and enter it using the Load function.
3. The CP input is in its quiescent HIGH state.
4. Select the 48-bit polynomial on the S_0–S_1 inputs.
5. Select Correct Normal mode on the C_0–C_2 inputs.
6. Let R_1 and R_2 be two external counters initialized to zero.
7. Test EP output. If it is HIGH, error pattern has already been found. The error location is $8R_1 + R_2 - 48$.
8. If EP output is LOW, test the AE output. If AE is HIGH, make CP input LOW and then HIGH. Increment R_2. If the AE output is LOW, make the CP input LOW and then HIGH. Increment R_1.
9. If $R_1 + R_2 - 48$ is greater than the record length, the error is uncorrectable, so abort the correction process.
10. Keep repeating from step 7 until the error is located.

Correct High Speed

The maximum number of clock cycles needed to find the error pattern using the normal correction method is N where N is the period of the polynomial. Thus a polynomial with a large period may require a large number of clock cycles for error correction not acceptable in some applications. The BEP has facilities for high-speed correction using the Chinese remainder theorem method.

Let a polynomial consist of m factors with periods P_1, P_2...P_m. The period N of the composite polynomial is the product of the periods of the individual factors; i.e., $N = P_1 \cdot P_2 \cdot P_3 ... P_m$. If the Chinese remainder theorem is used for correction, the maximum number of clock cycles needed is $(P_1 + P_2 ... P_m)$. This number is usually much smaller than N. Thus, the Chinese remainder theorem method is faster than the normal method for error correction.

To employ the Chinese remainder theorem method, the syndromes must be obtained first using the Read High-Speed function. This function gives as many syndromes as the number of factors in the polynomial. In other words, the Register Array is divided into a number of sections; each section implementing one factor of the polynomial. The first factor of every polynomial is of the form $(X^C + 1)$. This factor is sometimes called the error pattern polynomial. The Chinese remainder theorem method requires that the syndrome obtained by the error pattern polynomial be repeatedly divided by the error pattern polynomial until the error pattern is found. This is done in a fashion similar to the Correct Normal method described before. The register section corresponding to the error pattern polynomial is repeatedly clocked. The error pattern is always characterized by a known number of consecutive zeros at predetermined bit positions. (There can be alignment exceptions while finding the error pattern, but for the purpose of this explanation, assume that alignment exceptions do not occur.)

After locating the error pattern, the error pattern register is prevented from clocking. Next, the register corresponding to the second factor is repeatedly clocked until it matches the error pattern and then this register is prevented from further clocking. This procedure is repeated for all remaining factors. As mentioned earlier, the P_0–P_3 inputs are provided to control clocking of the individual registers, and the PM_2–PM_4 outputs are provided to indicate matching of each register with the error pattern.

Let M_1 be the number of clock cycles required to find the error pattern and M_2, M_3, etc. be the number of clock cycles required to match subsequent factors as described above. The error location can then be computed by a formula of the form:

$$L = N \cdot K - (A_1M_1 + A_2M_2 + A_3M_3 + A_4 + ...).$$

Where A_1, A_2, etc. are predetermined constants for a given polynomial and K is the smallest integer that makes the right hand side of the equation positive, A_0, A_1 etc. are called Chinese remainder theorem coefficients. The number of coefficients equals the number of factors in the polynomial. Table 3 lists the coefficients for the polynomials. There is one additional adjustment for the 35-bit polynomial – the error location for this polynomial is computed by using the formula $L = N \cdot K - (A_1M_1 + A_2M_2 + 5)$. This modification is required because 35 bits are really five bytes with the last five bits being unused.

TABLE 2. POLYNOMIAL PERIODS

Polynomial	Period Factor 1	Period Factor 2	Period Factor 3	Period Factor 4	Composite Period (N)
56-Bit	22	13	89	23	585442
32-Bit	21	2047	–	–	42987
35-Bit	23	4095	–	–	94185

As in the normal method, every error detected may not necessarily be correctable. If the number of clock cycles to find the error pattern exceeds the period of the error pattern polynomial, or the number of clock cycles required to match a register exceeds the period of the polynomial corresponding to that register, the correction process must be aborted. Table 2 lists the applicable periods for polynomials.

TABLE 3. CHINESE REMAINDER THEOREM COEFFICIENTS

Polynomial	A_1	A_2	A_3	A_4
56-bit	452,387	2,521,904	578,864	2,647,216
32-bit	38,893	32,760	–	–
35-bit	4,095	720,728	–	–

The sequence of events is as follows:

1. The CP input is in the quiescent HIGH state. The ER output is HIGH indicating an error from the Read High Speed operations.
2. Select the polynomial using the S_0–S_1 inputs and specify Correct High Speed code on the C_0–C_2 inputs.
3. Set $P_1 = P_2 = P_3 =$ LOW, $P_0 =$ HIGH.
4. R_1 and R_2 are two external counters, initialized to zero.
5. Test the EP output. If the EP output is HIGH, error pattern is already found and $M_1 = 8R_1 + R_2$. Bring P_0 input LOW and go to step 10.
6. Establish HIGH on the P_0 input.
7. If the EP output is LOW, test the AE output. If the AE output is LOW, make the CP input LOW and then HIGH. Increment R_1. If the AE output is HIGH, make CP LOW and then HIGH. Increment R_2.
8. If $R_1 + R_2$ is greater than the period of the first factor, abort the correction process; the error is not correctable.
9. If the error is correctable, repeat from step 5.

10. Establish HIGH on the P_1 input.
11. M_2 is an external counter initialized to zero.
12. Test the PM_2 output. If the PM_2 output is HIGH, the second factor located the matching error pattern. Bring P_1 LOW and go to step 16.
13 If the PM_2 output is LOW, make the CP input LOW and then HIGH. Increment M_2.
14 If M_2 is greater than the period of the second factor, abort the correction process; the error is not correctable.
15. If the error is correctable, repeat from step 12.

The following additional steps are performed only for the 56-bit polynomial. In case of the 32-bit or 35-bit polynomial, proceed with computations for error location.

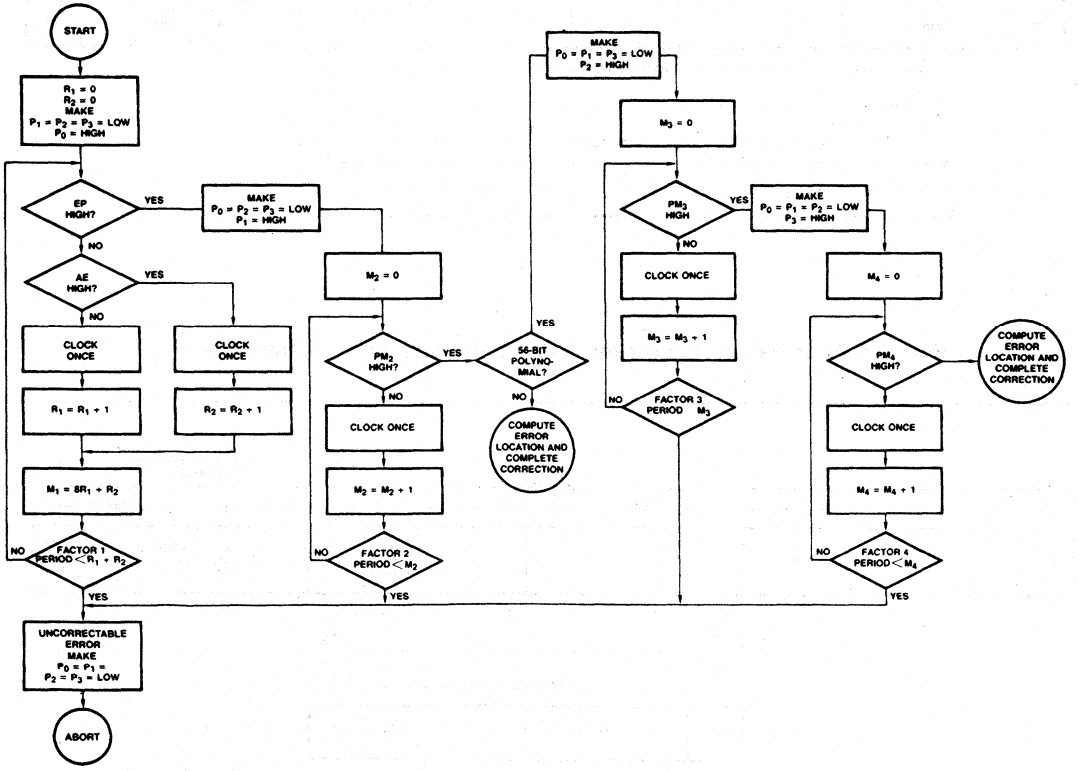
16. Establish HIGH on the P_2 input.
17. M_3 is an external counter initialized to zero.
18. Test the PM_3 output. If it is HIGH, the third factor located the matching error pattern. Bring P_2 input back LOW and go to step 22.
19. If the PM_3 output is LOW, make the CP input LOW and then HIGH. Increment M_3.
20. If M_3 is greater than the period of the third factor, abort the correction process; the error is not correctable.
21. If the error is correctable, repeat from step 18.
22. Establish HIGH on the P_3 input.
23. M_4 is an external counter initialized to zero.
24. Test the PM_4 output. If it is HIGH, the matching error pattern is found by the fourth factor. Compute the error location.
25. If the PM_4 output is LOW, make CP input LOW and then HIGH. Increment M_4. If M_4 is greater than the period of the fourth factor, abort the correction process; the error is not correctable.
26. If the error is correctable, repeat from step 24.

The following flowchart (Figure 3) summarizes the Correct High Speed function.

Load

This function enters the reciprocal of the syndrome into the Am9520. In the case of the 48-bit polynomial, the reciprocal of the syndrome must be formed externally and then entered into the Am9520 before error correction can start.

When the Load function is selected, the Register Array is configured as a simple 8-bit wide 7 deep shift register. The $D_0 - D_7$ are the inputs to this shift register. Before starting the correction process, seven bytes must be shifted in using the Load function – the first six bytes are the reciprocal of the syndrome, and the last byte is an all-zero fill byte.



PF001180

Figure 3. Correct High-Speed Function

The sequence of events for accomplishing the Load function is as follows:

1. The CP input is in its quiescent HIGH state.
2. Select 48-bit polynomial on the S_0–S_1 inputs.
3. Select Load function on the C_0–C_2 inputs.
4. Set $\overline{MR}$ LOW, and then HIGH.
5. Present a byte to be loaded on the D_0–D_7 inputs.
6. Make CP input LOW and then HIGH.
7. Repeat from step 5 until all six bytes of the reciprocal are entered.
8. Make D_0–D_7 input LOW for the all-zero dummy fill byte.
9. Make CP input LOW and then HIGH.

Error Pattern Information

The discussion of Correct Normal and Correct High Speed functions described the procedure for finding the error pattern and calculating the location of the error burst. The devices provide the error pattern on 12 outputs – eight bits on the Q_0–Q_7 outputs and four bits on the LP_0–LP_3 outputs. It was also mentioned that the REP input must be HIGH to read the error pattern.

The error location calculated using the formulas given is always in number of bits. In case of 56-bit, 35-bit and 32-bit polynomials, the calculated error location value corresponds to the beginning of the error burst counting from the last check bit. The calculated error location is such that when 12 consecutive bits of the record are exclusive ORed into the error pattern, then the error burst is corrected (see Figure 4).

Figure 5 depicts error pattern information for the 48-bit polynomial. In this case, the computed error location refers in bits to the first bit in the burst. However, the burst goes towards the beginning of the data. In the case of 56-, 32- and 35-bit, the burst was towards the check bits. This difference is caused by using the reciprocal of the syndrome.

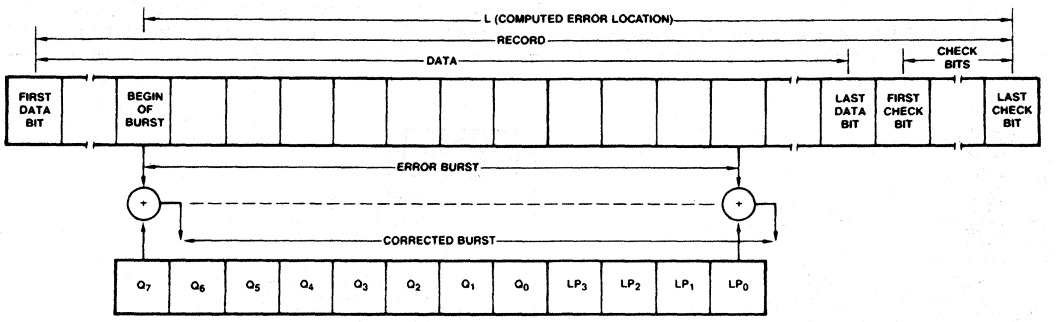


Figure 4. Error Pattern Format for 56-Bit, 35-Bit and 32-Bit Polynomials

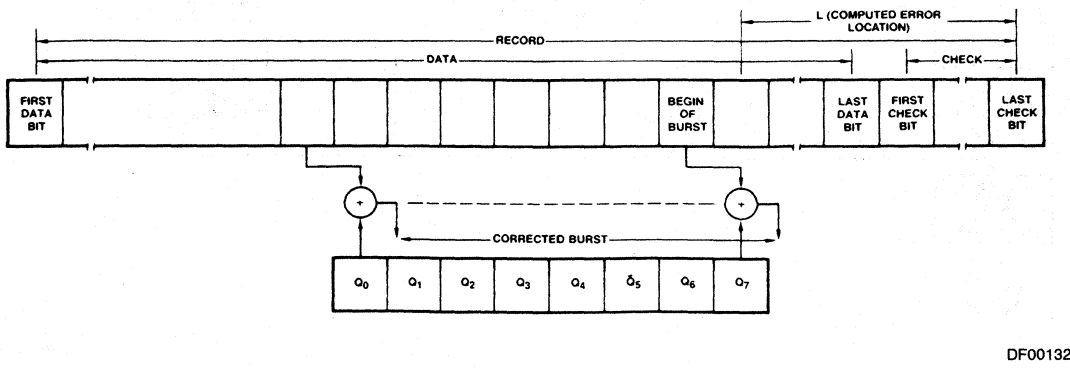
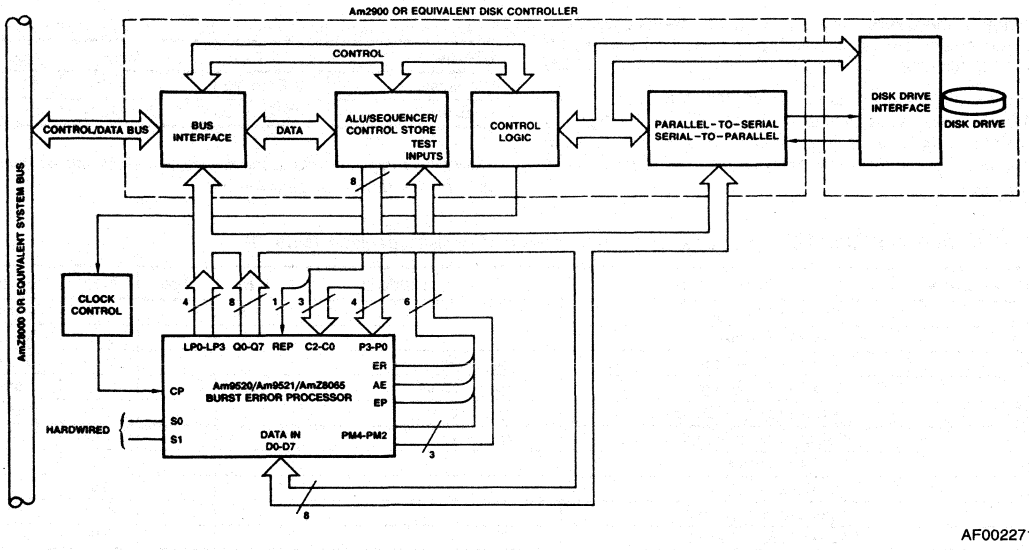


Figure 5. Error Pattern Format for 48-Bit Polynomial

TYPICAL APPLICATION DIAGRAM



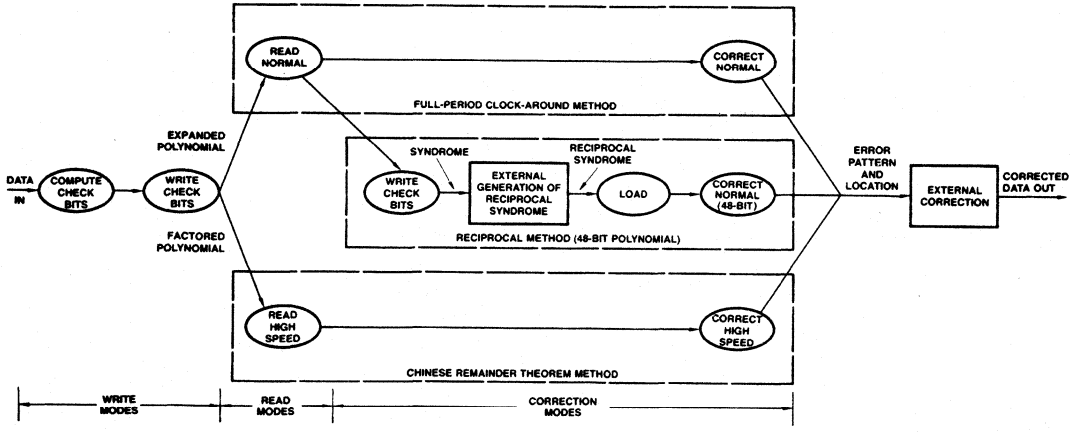
AF002271

Figure 6. Am9520/Am9521/AmZ8065 Burst Error Processor

APPLICATIONS

The BEP is designed for use in both microprogrammed and microprocessor disk controller systems. Figure 6 shows the BEP interfacing to an Am2900 bipolar bit-slice microprogrammed disk controller. The BEP can be interfaced to microprocessor-driven disk controller systems as well.

The controller in these designs would implement the control and clocking signals for the BEP necessary to execute the write, read and correction functions for a given polynomial selection. The operational flow for the methods available is shown in Figure 7.



PF001190

Figure 7. BEP Operational Flow Diagram

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any pin relative to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5V ±5%	0V
Industrial	−40°C to 85°C	5V ±10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input LOW Voltage		−0.5		+.8	Volts
V_{IH}	Input HIGH Voltage		2.0		V_{CC}	Volts
V_{OL}	Output LOW Voltage	I_{OL} = 3.2mA			0.45	Volts
V_{OH}	Output HIGH Voltage	I_{OH} = −400μA	2.4			Volts
I_{OL}	Output Leakage Current	V_{OUT} = 0.4V			10	μA
I_{LOH}	Output Leakage Current	V_{OUT} = V_{CC}			10	μA
C_{IN}	Input Capacitance				15	pF
$C_{I/O}$	I/O Capacitance				25	pF
I_{LL}	Input Leakage Current				±10	μA
I_{CC}	Power Supply Current				275	mA

Note 1. Typical values apply at T_A = 25°C and V_{CC} = 5.0V. See table above for Operating Ranges.

00617B

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING TEST INPUT/OUTPUT WAVEFORM

WF003670

Am9520/Am9521/AmZ8065 SWITCHING CHARACTERISTICS

The table below specifies the guaranteed performance of this device over the commercial operating range of 0 to +70°C with V_{CC} from 4.75V to 5.25V. All data are in nanoseconds. Switching tests are made with inputs and outputs measured at 0.8V for a LOW and 2.0V for a HIGH. Outputs are fully loaded with $C_L \geqslant 50pF$. See Switching Waveform figures for graphic illustration of timing parameters.

SWITCHING CHARACTERISTICS

Number	Parameters	Description	Am9520 Am9521 AmZ8065		Am9520-1 Am9521-1 AmZ8065-1		Units
			Min	Max	Min	Max	
1	TWCPL	CP Width LOW	180		105		ns
2	TCYCP	CP Cycle Time	400		250		ns
3	TWCPH	CP Width HIGH	180		105		ns
4	TWMRL	$\overline{MR}$ Width LOW	800		500		ns
5	TREC	$\overline{MR}\uparrow$ to CP↓ Time (Recovery)	250		250		ns
6	TSDCP	D_0-D_7 to CP↑ Setup Time	350		200		ns
7	THDCP	CP↑ to D_0-D_7 Hold Time	0		0		ns
8	TSCCP	C_0-C_2 or S_0-S_1 to CP↑ Setup Time	400		200		ns
9	THCCP	CP↑ to C_0-C_2, S_0-S_1, P_0-P_3 Hold time	0		0		ns
10	TSCCPL	C_0-C_2 or S_0-S_1 to CP↓ Setup Time	180		95		ns
11	TVCQ	C_0-C_2, S_0-S_1 to Q_0-Q_7 Valid Delay		200		150	ns
12	TIVCPQ	CP↑ to Q_0-Q_7 Invalid Delay	0		0		ns
13	TVCPQ	CP↑ to Q_0-Q_7 Valid Delay		200		150	ns
14	TIVCQ	C_0-C_2 to Q_0-Q_7 Three-State Delay		100		100	ns
15	TMRERL	$\overline{MR}$↓ to ER↓ Delay		200		200	ns
16	TCPER	CP↑ to ER Valid Delay		200		200	ns
17	TWCPCL	CP Width $\overline{LOW}$ for Correct Functions	450		450		ns
18	TWCPCH	CP Width $\overline{HIGH}$ for Correct Functions	450		450		ns
19	TCYCPC	CP Cycle Time for Correct Functions	1000		1000		ns
20	TCEP	C_0-C_2 to EP or AE Valid Delay		250		250	ns
21	TCPEP	CP↓ to EP, AE, or PM_2-PM_4 Valid Delay		400		400	ns
22	TSCPS	P_0, P_1, P_2, P_3 to CP↓ Setup Time	400		400		ns
23	TCEP	P_0 to EP or AE Delay		250		250	ns
24	TCPC	C_0-C_2, S_0-S_1 to CP↓ Setup Time for Correct Functions	400		400		ns
25	TPPM	P_1, P_2, P_3 to Corresponding PM Output Delay		250		250	ns
26	TCPEPI	CP↓ to EP, AE, PM_2, PM_3, and PM_4 Invalid Delay	0		0		ns
27	TPEPI	P_0↓ to EP, AE Invalid Delay	0		0		ns
28	TWREP	REP Pulse Width $\overline{HIGH}$	250		250		ns
29	TREPQ	REP↑ to Q_0-Q_7 and LP_0-LP_3 Delay		150		150	ns
30	TREPQI	REP↓ to Q_0-Q_7 and LP_0-LP_3 Three-State Delay		100		100	ns
31	TPPM	P_1, P_2, P_3↓ to PM_2, PM_3, PM_4 Invalid	0		0		ns
32	TCPM	C_0-C_2 to EP, AE, PM_2-PM_4 Invalid	0		0		ns

SWITCHING WAVEFORMS

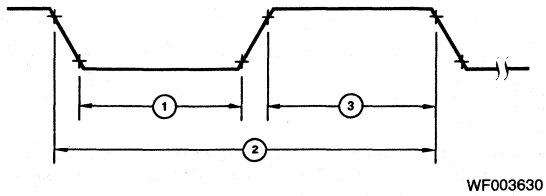


Figure 8. Clock Waveform for All Functions Except Correct Normal or Correct High-Speed

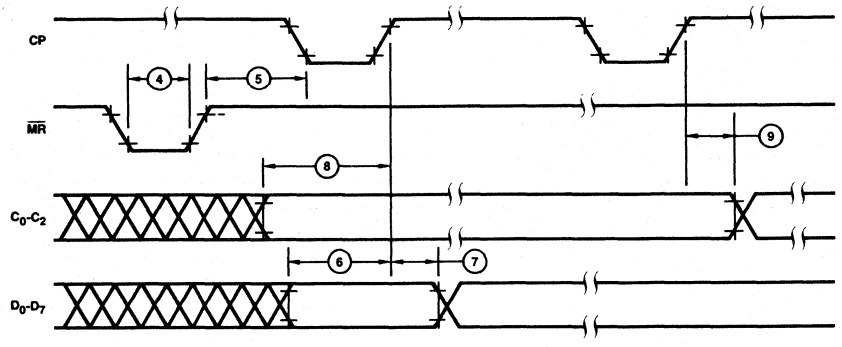


Figure 9. Timing for Compute Check Bits or Load Function

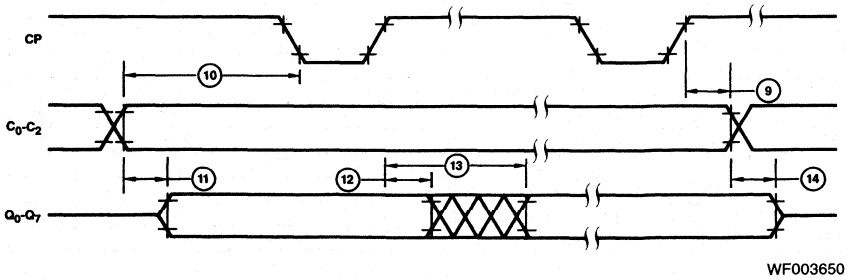


Figure 10. Timing for Write Check Bits Function

Notes: 1. REP input assumed low.
2. Q_0–Q_7 outputs will be high impedance if C_0–C_2 inputs do not specify Write Check Bits function.

SWITCHING WAVEFORMS (Cont.)

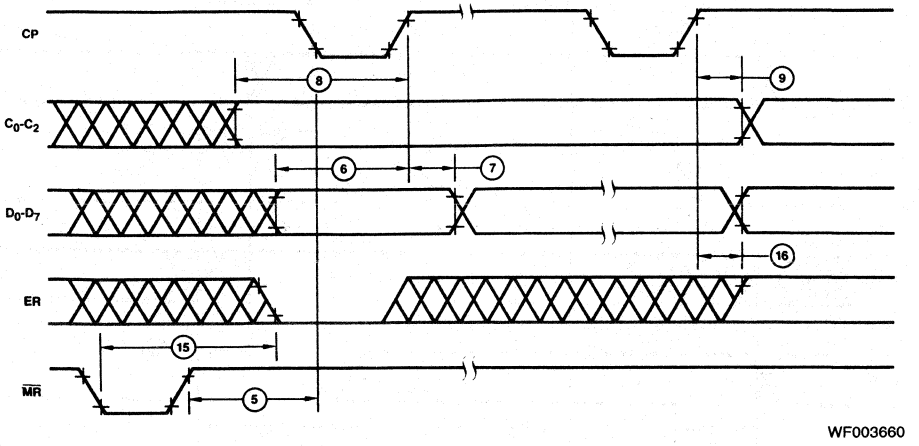


Figure 11. Timing for Read Normal or Read High-Speed Function
Note: ER output is a function of the contents in the register array flip-flops.

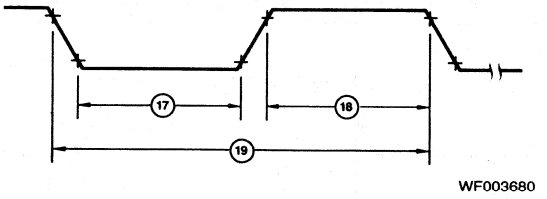


Figure 12. Clock Waveform for Correct Normal or Correct High-Speed Functions

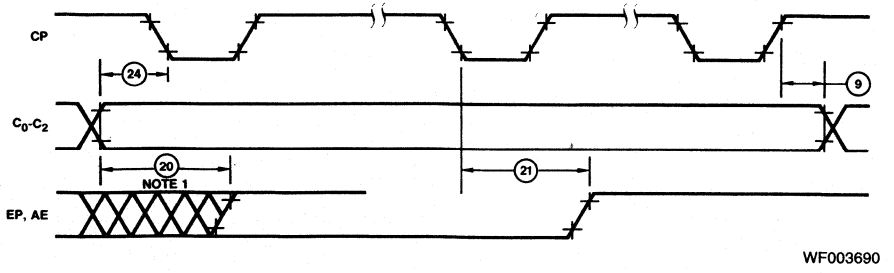


Figure 13. Timing for Correct Normal Function
Note 1: Assumes AE or EP output becomes active without any clocking.

SWITCHING WAVEFORMS (Cont.)

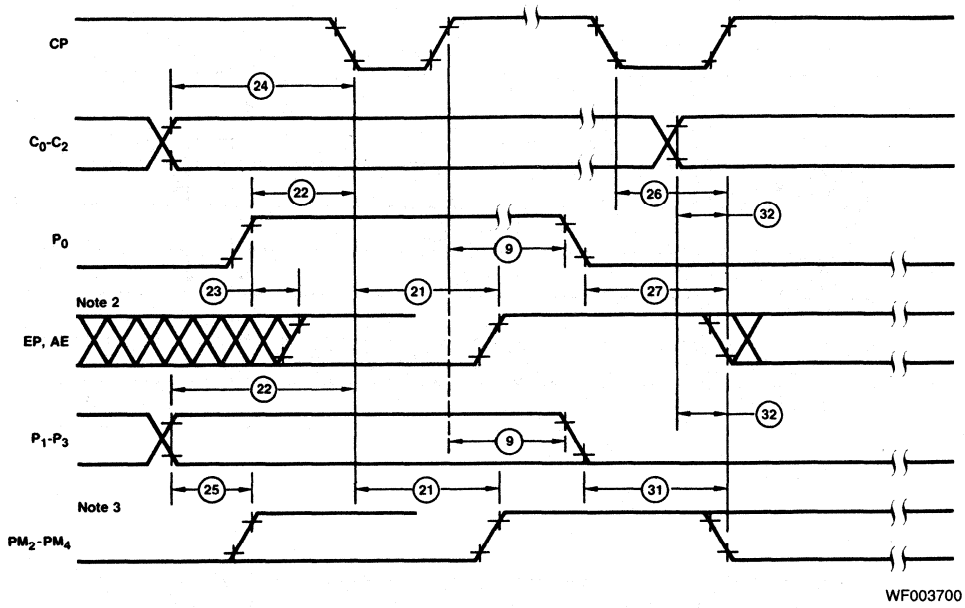


Figure 14. Timing for Correct High-Speed Function

Note 2: Assumes EP, AE becomes active without clocking.
Note 3: Assumes corresponding PM output becomes active without clocking.

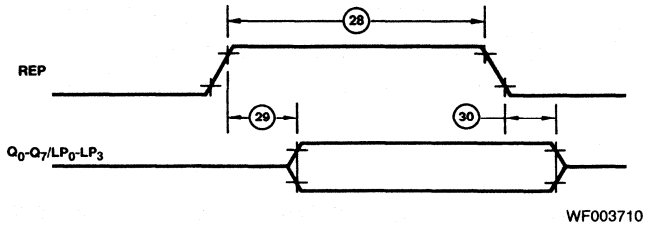


Figure 15. Read Error Pattern Timing

Am9568

Data Ciphering Processor
(DCP)

DISTINCTIVE CHARACTERISTICS

- **Encrypts and decrypts data**
  Implements National Bureau of Standards Data Encryption Standard (DES) algorithm
- **Throughput over 1.5M bytes per second**
  Operates at data rates fast enough for disk controllers, high-speed DMA, telecommunication channels
- **Supports three ciphering options**
  Electronic Code Book for disk applications, Cipher Block Chain for high-speed telecommunications, and Cipher Feedback for low-to-medium speed, byte-oriented communications
- **Three separate key registers on one chip**
  Separate registers for encryption key, decryption key and master key improve system security and throughput by eliminating need to reload keys frequently.
- **Three separate data ports provide flexible interface, improved security**
  The DCP utilizes a Master Port, Slave Port and Key Port. Functions of the three ports can be programmed by the user to provide for simple interface to iAPX86 and Am2900 systems and to provide total hardware separation of encrypted data, clear data and keys.

GENERAL DESCRIPTION

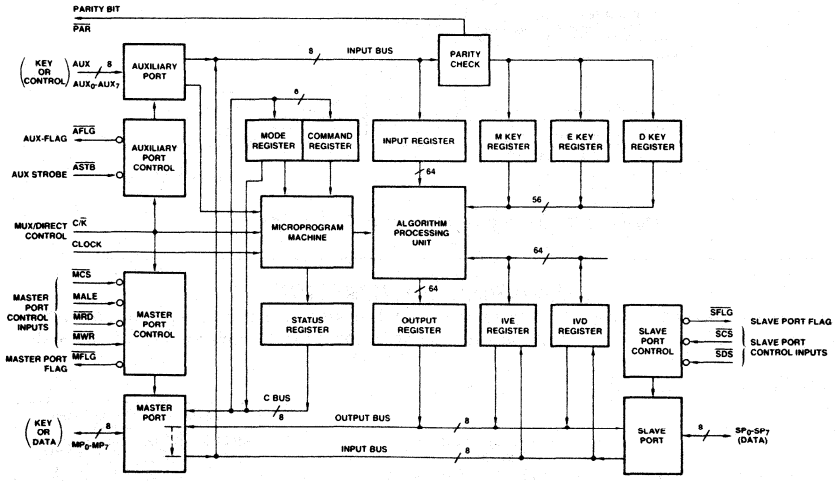
The Am9568 Data Ciphering Processor is an N-channel silicon gate LSI product containing the circuitry necessary to encrypt and decrypt data using the National Bureau of Standards Encryption Algorithm. It is designed to be used in a variety of environments, including dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems.

The DCP provides a high throughput rate using Cipher Feedback, Electronic Code Book or Cipher Block Chain operating modes. Separate ports for key input, clear data and enciphered data enhance security.

The system communicates with the DCP using commands entered in the Master Port and through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities are all performed concurrently. External DMA control can easily be used to enhance throughput in some system configurations.

This device is designed to interface directly to the iAPX86, 88 CPU bus and, with a minimum of external logic, to the 2900 and 8051 families of processors.

BLOCK DIAGRAM



BD003370

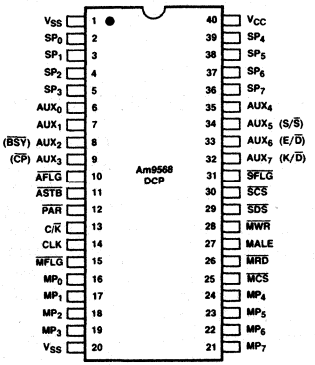
RELATED PRODUCTS

Part No.	Description
Am9518/Z8068	Data Ciphering Processor

Export of this device from the United States is subject to control by the U.S. Department of State.

Refer to page 7-1 for Essential Information on Military Devices

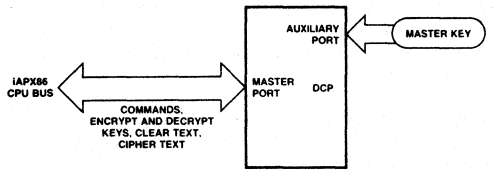
CONNECTION DIAGRAM
Top View



CD005200

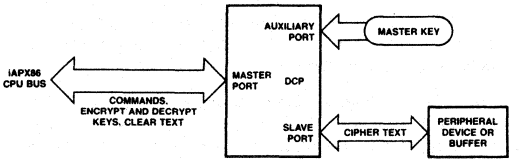
Note: Pin 1 is marked for orientation

DCP DATA FLOW OPTIONS



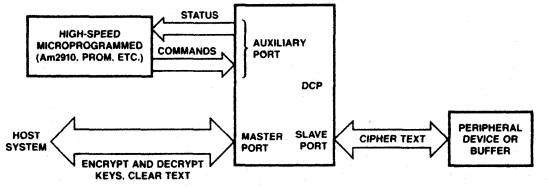
AF002480

Single-Port Configuration, Multiplexed Control



AF002490

Dual-Port Configuration, Multiplexed Control

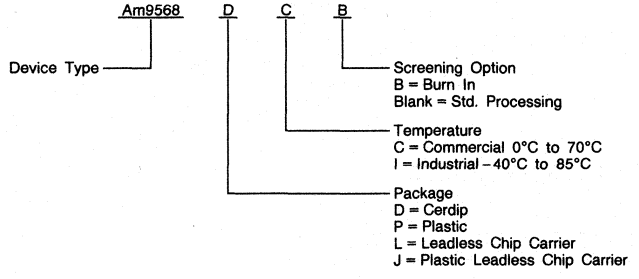


AF002500

Dual-Port Configuration, Direct Control

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9568	DC, DCB, PC, DI, PCB, PI, PIB, DIB, LC, LCB, LI, LIB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
40	V_{CC}		+5 Volt Power Supply.
1, 20	V_{SS}		Ground (2 pins).
14	CLK	I	(Clock, TTL levels). An external timing source is input via the CLK pin. The Master and Slave Port Data Strobe signals ($\overline{MRD}$, $\overline{MWR}$, $\overline{SDS}$) must change synchronously with this clock input, as must AUX_5-S/$\overline{S}$ in Direct Control Mode (C/$\overline{K}$ HIGH). In addition, the Auxiliary, Master and Slave Port Flag outputs ($\overline{AFLG}$, $\overline{MFLG}$ and $\overline{SFLG}$) will change synchronously with the clock.
13	C/$\overline{K}$	I	(Control/Key Mode Control). This input is the primary control over the operating characteristics of the DCP. A LOW input on C/$\overline{K}$ places the DCP into Multiplexed Control Mode, enabling programmed access to internal registers through the Master Port and enabling input of keys through the Auxiliary Port. A HIGH input on C/$\overline{K}$ specifies operation in Direct Control Mode, wherein several of the Auxiliary Port pins become direct control/status signals which can be driven/sensed by high-speed controller logic (such as the Am29116 or Am2901/Am2903-based processors), and access to internal registers through the Master Port is limited to the Input or Output Register.
16-19 24-21	MP_0-MP_7	I/O	(Master Port Bus). These eight bidirectional lines are used to specify internal register addresses in Multiplexed Control Mode (see C/$\overline{K}$) and to input and output data. The Master Port provides software access to the Status, Command and Mode Registers, as well as the Input and Output Registers. The three-state Master Port outputs will be enabled only when the Master Port is selected by Master Port Chip Select ($\overline{MCS}$) LOW and when Master Port Read ($\overline{MRD}$) is strobed LOW. MP_0 is the low-order bit. Data and key information are entered into this port with the most significant byte in first.
25	$\overline{MCS}$	I	(Master Port Chip Select). This active LOW input signal is used to select the Master Port. In Multiplexed Control Mode (C/$\overline{K}$ low), the level on $\overline{MCS}$ is latched internally on the falling edge of Master Port Address Latch Enable (MALE). This latched level is retained as long as MALE is LOW; when MALE is HIGH, the latch becomes transparent and the internal signal will follow the $\overline{MCS}$ input. In Direct Control Mode (C/$\overline{K}$ HIGH), no latching of Master Port Chip Select occurs; the level on $\overline{MCS}$ is passed directly to the internal select circuitry irrespective of state of Master Port Address Latch Enable (MALE).
27	MALE	I	(Master Port Address Latch Enable). In Multiplexed Control Mode (C/$\overline{K}$ low), an active HIGH signal on this pin indicates the presence of valid address and chip select information at the Master Port. This information will be latched internally on the falling edge of Address Latch Enable. When C/$\overline{K}$ is HIGH (Direct Control Mode), MALE may be HIGH or LOW without affecting DCP operation.
26	$\overline{MRD}$	I	(Master Port Read Data). This active LOW input is used in coincidence with a valid Master Port Chip Select ($\overline{MCS}$), to indicate that data is to be placed on MP_0-MP_7 for an output operation. Master Port Read ($\overline{MRD}$) and Master Port Write ($\overline{MWR}$) are normally mutually exclusive; if both go LOW simultaneously, the DCP is reset to ECB Mode and all flags go inactive.
28	$\overline{MWR}$	I	(Master Port Write). This input signal indicates to the DCP that valid data is present on MP_0-MP_7 for an input operation. The trailing edge of $\overline{MWR}$ latches the data in the selected internal register. If $\overline{MWR}$ and $\overline{MRD}$ both go LOW simultaneously, the DCP is reset.
15	$\overline{MFLG}$	O	(Master Port Flag). This active LOW flag is used to indicate the need for a data transfer into or out of the Master Port during normal ciphering operation. Depending upon control bits written to the Mode Register (see Register Description), the Master Port will be associated with either the Input Register or the Output Register. If data is to be transferred through the Master Port to the Input Register, the $\overline{MFLG}$ reflects the contents of the Input Register; after any Start command is entered, $\overline{MFLG}$ will go active (LOW) whenever the Input Register is not full. $\overline{MFLG}$ is forced HIGH by any command other than a Start. Conversely, if the Master Port is associated with the Output Register, $\overline{MFLG}$ reflects the contents of the Output Register (except in Single Port configuration – see Detailed Description). $\overline{MFLG}$ will go active (LOW) whenever the Output Register is not empty. In Single Port Configuration, the Master Port Flag reflects the contents of the Input Register, while the Slave Port Flag ($\overline{SFLG}$, see below) is associated with the Output Register.
2-5 39-36	SP_0-SP_7	I/O	(Slave Port Bus). The Slave Port provides a second data input/output interface to the DCP, allowing overlapped input, output and ciphering operations. The tri-state Slave Port outputs will be driven only when Slave Port Chip Select ($\overline{SCS}$) and Slave Port Data Strobe ($\overline{SDS}$) are both LOW and $\overline{SFLG}$ = 0, and the internal Port Control Configuration allows output to the Slave Port. SP_0 is the LOW order bit. Data entered or retrieved through this port is the most significant byte in/out first.
30	$\overline{SCS}$	I	(Slave Port Chip Select). This active LOW signal is logically combined with Slave Port Data Strobe ($\overline{SDS}$) to facilitate Slave Port data transfers in a bus environment. $\overline{SCS}$ is not latched internally, and may be tied permanently LOW without impairing Slave Port operation.
29	$\overline{SDS}$	I	(Slave Port Data Strobe). This active LOW input, in coincidence with Slave Port Chip Select ($\overline{SCS}$) LOW, indicates to the DCP that valid data is on the SP_0-SP_7 lines for an input operation, or that data is to be driven onto the SP_0-SP_7 lines for output. The direction of data flow is determined by control bits in the Mode Register (see Register Description).
31	$\overline{SFLG}$	O	(Slave Port Flag). This active LOW output indicates the state of either the Input Register or the Output Register, depending on control bits in the Mode Register. In Single Port Configuration, $\overline{SFLG}$ will go active whenever the Output Register is not empty during normal processing. In Dual Port Configuration, $\overline{SFLG}$ will reflect the content of whichever register is associated with the Slave Port. If the Input Register is assigned to the Slave Port, $\overline{SFLG}$ will go active whenever the Input Register is not full, once any of the Start commands has been entered; $\overline{SFLG}$ will be forced inactive if any other command is entered. Conversely, if the Slave Port is assigned to the Output Register, $\overline{SFLG}$ will go active whenever the Output Register is not empty.

Refer to page 7-1 for Essential Information on Military Devices

Pin Description (Cont.)

Pin No.	Name	I/O	Description
6-9 35-32	AUX_0-AUX_7	I/O	(Auxiliary Port Bus, Bidirectional). When the DCP is operated in Multiplexed Control Mode (C/$\overline{K}$ LOW), these eight lines form a key-byte input port which may be used to enter the Master and Session Keys. In fact, this port is the only path available for entering the Master Key. (session keys may alternatively be entered via the Master Port.) AUX_0 is the low-order bit and is considered to be the parity bit in key bytes. The most significant byte is entered first. When the DCP is operated in Direct Control Mode, (C/$\overline{K}$ HIGH), the Auxiliary Port's key-entry function is disabled and five of the eight lines become direct control/status lines for interfacing to high-speed microprogrammed controllers. In this case, AUX_0, AUX_1 and AUX_4 have no function (they may be tied HIGH), and the other pins are defined as below.
34	AUX_5-S/$\overline{S}$	I	(Start/$\overline{Stop}$). When this pin goes LOW (Stop),below the DCP will follow the sequence that would normally occur were a Stop command to be entered. Conversely, when this pin goes HIGH, a sequence equivalent to a Start Encryption or Start Decryption command will be followed. At the time AUX_5-S/$\overline{S}$ goes HIGH, the level on AUX_6-E/$\overline{D}$ (see below) selects either the Start Encryption or Start Decryption interpretation.
32	AUX_7-K/$\overline{D}$	I	(Key/$\overline{Data}$). When this signal goes HIGH, the DCP initiates a key-data input sequence as if a Load Clear E (or D) Key Through Master Port command had been entered. The level on AUX_6-E/$\overline{D}$ will determine whether the subsequently entered clear-key bytes are written into the E Key Register (E/$\overline{D}$ HIGH) or the D Key Register (E/$\overline{D}$ LOW). AUX_7-K/$\overline{D}$ and AUX_5-S/$\overline{S}$ are mutually exclusive control lines; when one goes active (HIGH), the other must be and remain inactive (LOW) until the first returns to an inactive state. In addition, both lines must be inactive (LOW) whenever a transition occurs on C/$\overline{K}$ (entering or exiting Direct Control Mode).
33	AUX_6-E/$\overline{D}$	I	(Encrypt/$\overline{Decrypt}$). When AUX_5-S/$\overline{S}$ goes HIGH, initiating a normal data ciphering operation, this input specifies whether the ciphering algorithm is to encrypt (E/$\overline{D}$ HIGH) or decrypt (LOW). When AUX_7-K/$\overline{D}$ goes HIGH, initiating entry of key bytes, the level on AUX_6-E/$\overline{D}$ specifies whether the bytes are to be written into the E Key Register (E/$\overline{D}$ HIGH) or the D Key Register (E/$\overline{D}$ LOW). The AUX_6-E/$\overline{D}$ input is not latched internally, and must be held constant whenever one or more of AUX_5-S/$\overline{S}$, AUX_7-K/$\overline{D}$, AUX_2-$\overline{BSY}$, or AUX_3-$\overline{CP}$ are active. Failure to maintain the proper level on AUX_6-E/$\overline{D}$ during loading or ciphering operations will result in scrambled data in the internal registers.
8	AUX_2-$\overline{BSY}$	O	(Busy). This active-low status output gives a hardware indication that the ciphering algorithm is in operation. AUX_2-$\overline{BSY}$ is driven by the BSY bit in the Status Register (see Register Description), such that when the BSY bit is "1" (active), AUX_2-$\overline{BSY}$ is LOW.
9	AUX_3-$\overline{CP}$	O	(Command Pending). This active-low status output gives a hardware indication that the DCP is ready to accept input of key bytes following a LOW-to-HIGH transition on AUX_7-K/$\overline{D}$. AUX_3-$\overline{CP}$ is driven by the CP bit in the Status Register, such that when the CP bit is "1" (active), AUX_3-$\overline{CP}$ is LOW.
11	$\overline{ASTB}$	I	(Auxiliary Port Strobe). The rising (trailing) edge of $\overline{ASTB}$ strobes the key data on pins AUX_0-AUX_7 into the appropriate internal key register in Multiplexed Control Mode (C/$\overline{K}$ LOW). This input is ignored unless $\overline{AFLG}$ and C/$\overline{K}$ are both LOW. One byte of key data is entered on each $\overline{ASTB}$, most significant byte first.
10	$\overline{AFLG}$	O	(Auxiliary Port Flag). This active LOW output signal indicates that the DCP is expecting key data to be entered on pins AUX_0-AUX_7. This can occur only when C/$\overline{K}$ is LOW and a Load Key Through AUX Port command has been entered. $\overline{AFLG}$ will remain active (LOW) during input of all eight bytes and will go inactive with the leading edge of the eighth strobe ($\overline{ASTB}$).
12	$\overline{PAR}$	O	(Parity). The DCP checks all key bytes for correct (odd) parity as they are entered through either the Master Port (Multiplexed or Direct Control Mode) or the Auxiliary Port (Multiplexed Control Mode only). If any key byte contains even parity, the PAR bit in the Status Register is set to "1" and $\overline{PAR}$ goes LOW. (See Parity Checking of Keys.) Least significant bit of key data is the parity.

DETAILED DESCRIPTION

The overall design of the DCP, as shown in the block diagram on the next page, is optimized for high data throughput. Data bytes can be transferred through both the Master and Slave Ports, and key bytes can be written through both the Auxiliary and Master Ports. Three 8-bit buses, Input, Output and C Bus, carry data and key bytes between the ports and the internal registers. Three 56-bit, write-only key registers are provided for the Master (M) Key, the Encryption (E) Key and the Decryption (D) Key. Parity checking is provided on incoming key bytes. Two 64-bit registers are provided for Initializing Vectors (IVE and IVD) required for chained (feedback) ciphering modes. Three 8-bit registers (Mode, Command and Status) are accessible through the Master Port for interfacing to a host microprocessor, such as the iAPX86.

Algorithm Processing

The DCP's Algorithm Processing Unit (see the block diagram) is designed to encrypt and decrypt data according to the National Bureau of Standards Date Encryption Standard (DES), as specified in Federal Information Processing Standards Publication 46.

The DES specifies a method for encrypting 64-bit blocks of clear data ("plain text") into corresponding 64-bit blocks of "cipher text." The DCP offers three ciphering methods selected by the Cipher Type field of the Mode Register: Electronic Code Book (ECB), Cipher Block Chain (CBC) and Cipher Feedback (CFB). These methods are implemented in accordance with Federal Information Processing Standards Publication 46. Electronic Code Book (ECB) is a straightforward implementation of the DES: 64 bits of clear data in, 64 bits of cipher text out, with no cryptographic dependence between blocks. Cipher Block Chain (CBC) also operates on blocks of 64 bits, but includes a feedback step which chains consecutive blocks so that repetitive data in the plain text (such as ASCII blanks) does not yield repetitive cipher text. CBC also provides an error extension characteristic valuable in protecting against fraudulent data insertions and deletions. Cipher Feedback (CFB) is an additive stream cipher method in which the DES generates a pseudorandom binary stream which is then exclusive-OR'd with the clear data to form the cipher text. The cipher text is then fed back to form a portion of the next DES input block. The DCP implements 8-bit cipher feedback with one byte wide data input, output, and feedback

Refer to page 7-1 for Essential Information on Military Devices

paths. This method is useful for low speed, character-at-a-time serial communications.

Multiple Key Registers

The DCP provides the necessary registers to implement a multiple-key or Master Key system. In such an arrangement, a single Master Key, stored in the DCP M Key Register, is used only to encrypt session keys for transmission to remote DES equipment, and to decrypt session keys received from such equipment. The M Key Register may be loaded (with plain text) only through the Auxiliary Port, using the Load Clear M Key command. (See Commands.)

In addition to the M Key Register, the DCP contains two session key registers: the E Key Register, used to encrypt clear text, and the D Key Register, used to decrypt cipher text.

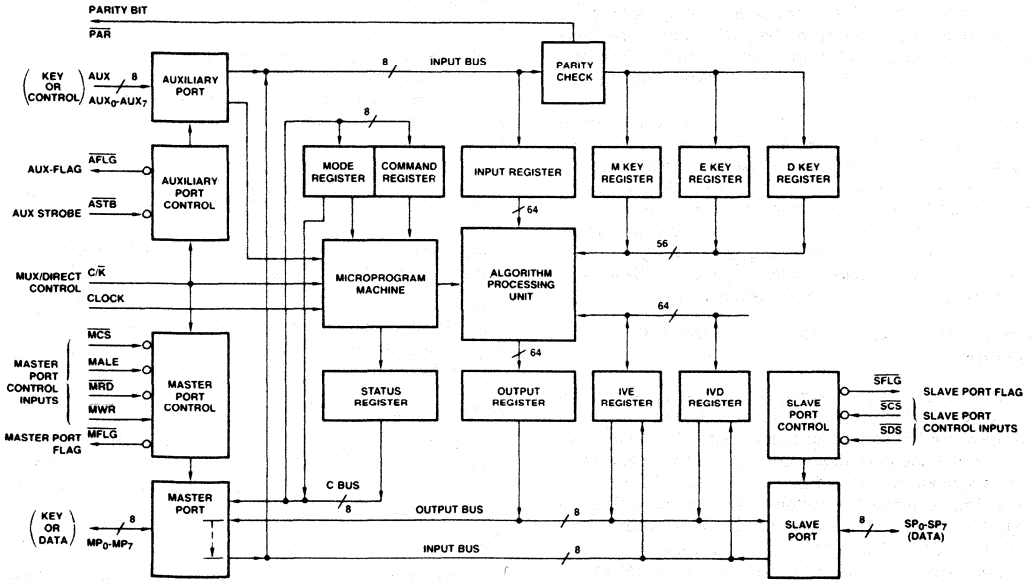
All three registers are loaded by writing commands like Load Clear E Key through Master Port into the Command Register, and then writing the eight bytes of key data to the port when the Command Pending = "1" in the Status Register. (See Commands.)

Operating Modes: Multiplexed Control vs. Direct Control

The DCP can be operated in either of two basic interfacing modes determined by the logic level on the C/$\overline{K}$ input pin. In Multiplexed Control Mode (C/$\overline{K}$ LOW), the DCP is internally configured to allow a host CPU to directly address five of the internal Control/Status/Data Registers and thereby control the device via mode and command values written to these registers. Also, in Multiplexed Control Mode, the Auxiliary Port is enabled for key-byte input.

If the logic level on C/$\overline{K}$ is brought HIGH, the DCP enters Direct Control Mode, and the Auxiliary Port pins are converted into direct hardware status or control signals that are capable of instructing the DCP to perform a functionally complete subset of its cipher processing at very high throughputs. This operating mode is particularly well-suited for ciphering data for high-speed peripheral devices, such as magnetic disk or tape.

Am9568 DCP BLOCK DIAGRAM



BD003370

Data Flow

Bits M_2, M_3 of the Mode Register control the flow of data into and out of the DCP through the Master and Slave Ports. Three basic configurations are provided: Single Port and two Dual Port configurations.

Single Port Configuration

The simplest configuration occurs when the Mode Register configuration bits are set to Master Port only. Under this operating configuration the Encrypt/Decrypt bit (M_4) controls the processing of data. Data to be encrypted or decrypted is written to the Master Port Input Register address. To facilitate monitoring of the Input Register status, the $\overline{MFLG}$ signal goes LOW when the Input Register is not full. Data is read by the host CPU through the Master Port Output Register address. $\overline{SFLG}$ goes LOW when the Output Register is not empty. Thus, $\overline{MFLG}$ redefined as a Master Input Flag and $\overline{SFLG}$ is redefined as a Master Output Flag.

Dual Port, Master Port Clear Configuration

In the Dual Port configurations, both the Master and Slave Ports are used for data entry and removal. In the Master Port Clear configuration, clear text for encryption can be entered only through the Master Port, and clear text resulting from decryption can be read out only through the Master Port. Cipher text can be handled only through the Slave Port. The actual direction of data flow is controlled either by the Encrypt/Decrypt bit (M_4) in the Mode Register or by the Start Encryption or Start Decryption commands. If encryption is specified, clear data will flow through the Master Port to the

Refer to page 7-1 for Essential Information on Military Devices

Input Register, and cipher data will be available at the Slave Port when it is ready to be read out of the Output Register. For decryption, the process is reversed: cipher data being written to the Input Register through the Slave Port and clear data being read from the Output Register through the Master Port.

Dual Port, Slave Port Clear Configuration

This configuration is identical to the previously described Dual Port, Master Port Clear configuration, except that the direction of ciphering is reversed. That is, all data flowing in or out of the Master Port is cipher text, and all data at the Slave Port is clear text.

Master Port Read/Write Timing

The DCP's Master Port is designed to operate with multiplexed address-data buses, such as the 8086/8088 processors. Several features of the Master Port logic should be stressed.

- The level on Master Port Chip Select ($\overline{MCS}$) is latched internally on the falling (trailing) edge of Master Port Address Latch Enable (MALE), thus relieving external address decode circuitry of the responsibility for latching chip select at address time.
- The levels on MP_1, MP_2 are also latched internally on the falling edge of MALE and are subsequently decoded to enable reading and writing of the DCP's internal registers (Mode, Command, Status, Input and Output). Again, this eliminates the need for external address latching and decoding.
- Data transfers through the Master Port are controlled by the levels and transitions on Master Port Read ($\overline{MRD}$) and Master Port Write ($\overline{MWR}$). Note that data transfers do not disturb either the chip-select or address latches, so that once the DCP and a particular register have been selected, any number of reads or writes of that register can be accomplished without intervening address cycles. This feature could greatly speed up loading keys and data, given the necessary transfer control external to the DCP.

Loading Keys and Initializing Vector (IV) Registers

Because the key and initializing vector registers are not directly addressable through any of the DCP's ports, keys and vector data must be loaded (and, in the case of vectors, read out) via "command data sequences" (see Commands). Most of the commands recognized by the DCP are of this type: a Load or Read command is written to the Command Register through the Master Port; the command processor responds by asserting the Command Pending output; the user then either writes eight bytes of key or vector data through the Master or Auxiliary Port, as appropriate to the specific command, or reads eight bytes of vector data from the Master Port.

In Direct Control Mode, only the E Key and D Key registers can be loaded; the M Key and IV Registers are inaccessible. Loading the E and D Key registers is accomplished by asserting the proper state on the AUX_6-E/$\overline{D}$ input (HIGH for E Key, LOW for D Key) and then raising the AUX_7-K/$\overline{D}$ input, indicating that key loading is required. The command processor will attach the proper Key Register to the Master Port and assert the AUX_3-$\overline{CP}$ (Command Pending) signal (active-low). The eight key bytes may then be written to the Master Port. In Multiplexed Control Mode, all key and vector registers are writable, and all but the Master (M) Key Register may be loaded with encrypted, as well as clear, data. If the operation is a Load Encrypted command, the subsequent data written to the Master or Auxiliary Port (as appropriate) is routed first to the Input Register and decrypted before being written into the specified Key or Vector Register.

Parity Checking of Keys

Key bytes are considered to contain seven bits of key information and one parity bit. By DES designation, the low-order bit is the parity bit. The parity checking circuit is enabled whenever a byte is written to one of three Key Registers. The output of the parity detection circuit is connected to pin $\overline{PAR}$, and the state of this pin is reflected in Status Register bit PAR (S_3). Status Register bit PAR goes to "1" whenever a byte with even parity (an even number of "1"s) is detected. In addition to the PAR bit, the Status Register has a Latched Parity Bit (LPAR, S_4) which is set to "1" whenever the Status Register PAR bit goes to "1." Once set, the LPAR bit is not cleared until a reset occurs or a new Load Key command is issued.

When an encrypted key is entered, the parity detect logic operates only after the decrypted key is available. The encrypted data is not checked for parity. The $\overline{PAR}$ signal will reflect the state of the decrypted bytes on a byte-to-byte basis, as they are clocked through the parity check logic on their way to the Key Register. Thus, the time $\overline{PAR}$ indicates the status of a byte of decrypted key data may be as short as four clock cycles. The LPAR bit in the Status Register will indicate if any erroneous bytes of data were entered.

Initialization

The DCP can be reset in several ways:

1. By the "Software Reset" command,
2. By a hardware reset, which occurs whenever both $\overline{MRD}$ and $\overline{MWR}$ go LOW simultaneously for 1 clock,
3. By writing to the Mode Register, and
4. By aborting any command.

All these sequences are the same internally, except that loading the Mode Register does not subsequently reset the Mode Register.

Once a reset process starts, the DCP is unable to respond to further commands for approximately five clock cycles.

If a power-up hardware reset is used, the leading edge of the reset signal should not occur until approximately 1 ms after V_{CC} has reached normal operating voltage. This delay time is needed for internal signals to stabilize.

Register Description

The registers in the DCP which can be directly addressed through the Master Port are shown with their addresses in Figure 1. A brief description of these registers and others not directly accessible is given below.

Command Register

Data written to the 8-bit, write-only Command Register through the Master Port is interpreted as an instruction. A detailed description of each command is given under Commands, and the commands and their binary representations are summarized in Figure 2.

Status Register

The bit assignments in the read-only Status Register are shown in Figure 4. The $\overline{PAR}$, $\overline{AFLG}$, $\overline{SFLG}$ and $\overline{MFLG}$ bits indicate the status of the like-named output pins, as do the bits Busy and Command Pending when the DCP is in Direct Control Mode (C/$\overline{K}$ HIGH). In each case, the output signal will be active LOW when the corresponding status bit is a "1." The Parity bit indicates the parity of the most recently entered key byte. The LPAR bit, on the other hand, indicates whether any key byte with even parity has been encountered since the last Reset or Load Key command.

The Busy bit will be a "1" whenever the ciphering algorithm unit is actively encrypting or decrypting data, either as a

Refer to page 7-1 for Essential Information on Military Devices

response to a command such as Load Encrypted Key (in which case the Command Pending bit will be a "1"), or in the ciphering of regular text (indicated by the Start/Stop bit being a "1"). The Busy bit will remain a "1" even after ciphering is complete if the ciphered data cannot be transferred to the Output Register because that register still contains output from a previous ciphering cycle. Busy will be "0" at all other times, including if no ciphering is possible because no data has been written to the Input Register.

The Command Pending bit will be set to "1" by any command whose execution requires the transfer of data to or from a non-addressable internal register, such as when writing key bytes to the E Key Register or reading bytes from the IVE Register. Thus, Command Pending will be set following all commands except the three Start commands, the Stop command and the Software Reset command. Command Pending will return to "0" after all eight bytes have been transferred following Load Clear, Read Clear or Read Encrypted commands, and after data has been transferred, decrypted and loaded into the desired register following Load Encrypted commands.

The Start/Stop bit is set to "1" when one of the Start commands is entered, and is reset to "0" whenever a reset occurs or when a new command other than a Start is entered.

C/$\overline{K}$	MP2	MP1	$\overline{MRD}$	$\overline{MWR}$	$\overline{MCS}$	Register Addressed
0	X	0	1	0	0	Input Register
0	X	0	0	1	0	Output Register
0	0	1	1	0	0	Command Register
0	0	1	0	1	0	Status Register
0	1	1	X	X	0	Mode Register
X	X	X	X	X	1	No Register Accessed
1	X	X	1	0	0	Input Register
1	X	X	0	1	0	Output Register

Figure 1. Master Port Register Addresses

Hex Code	Command
90	Load Clear M Key through Auxiliary Port
91	Load Clear E Key through Auxiliary Port
92	Load Clear D Key through Auxiliary Port
11	Load Clear E Key through Master Port
12	Load Clear D Key through Master Port
B1	Load Encrypted E Key through Auxiliary Port
B2	Load Encrypted D Key through Auxiliary Port
31	Load Encrypted E Key through Master Port
32	Load Encrypted D Key through Master Port
85	Load Clear IVE through Master Port
84	Load Clear IVD through Master Port
A5	Load Encrypted IVE through Master Port
A4	Load Encrypted IVD through Master Port
8D	Read Clear IVE through Master Port
8C	Read Clear IVD through Master Port
A9	Read Encrypted IVE through Master Port
A8	Read Encrypted IVD through Master Port
39	Encrypt with Master Key
41	Start Encryption
40	Start Decryption
C0	Start
E0	Stop
00	Software Reset

Figure 2. Command Codes in Multiplexed Control Mode

Pins				Command initiated
C/$\overline{K}$	AUX7 – K/$\overline{D}$	AUX6 – E/$\overline{D}$	AUX5 – S/$\overline{S}$	
H	L	L	↑	Start Decryption
H	L	H	↑	Start Encryption
H	L	X	↓	Stop
H	↑	L	L	Load D Key Clear through Master Port
H	↑	H	L	Load E Key Clear through Master Port
H	↓	X	L	End Load Key Command
H	H	X	H	Not Allowed
L	Data	Data	Data	AUX Pins Become Key-Byte Inputs

Figure 3. Implicit Command Sequences in Direct Control Mode

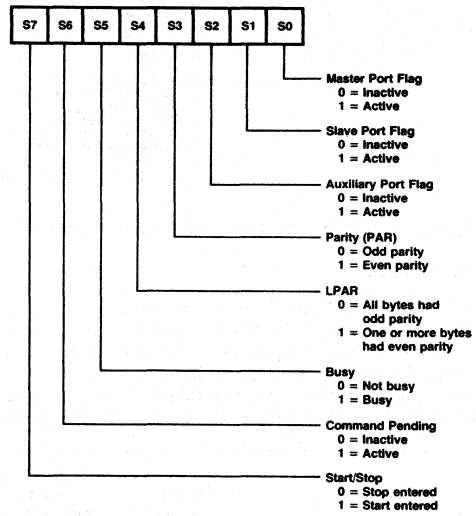


Figure 4. Status Register Bit Assignments

Mode Register

Bit assignments in this 5-bit read/write register are shown in Figure 5. The Cipher Type bits (M_1, M_0) indicate to the DCP which ciphering algorithm is to be used. On reset, the Cipher Type defaults to Electronic Code Book.

Configuration bits (M_3, M_2) indicate which data ports are to be associated with the Input and Output Registers and flags. When these bits are set to the Single Port, Master Only configuration (M_3, M_2 = 10), the Slave Port is disabled, and no manipulation of Slave Port Chip Select ($\overline{SCS}$) or Data Strobe ($\overline{SDS}$) can result in data movement through the Slave Port; all data transfers are accomplished through the Master Port. Both $\overline{MFLG}$ and $\overline{SFLG}$ are used in this configuration; $\overline{MFLG}$ gives the status of the Input Register and $\overline{SFLG}$ the Output Register.

When the Configuration Bits are set to one of the Dual Port configurations (M_3, M_2 = 00 or 01), both the Master and Slave

Refer to page 7-1 for Essential Information on Military Devices

Ports are available for input and output. When M_3, M_2 = 01 (the default configuration), the Master Port handles clear data while the Slave Port handles encrypted data. Configuration M_3, M_2 = 00 reverses this assignment. Actual data direction at any particular moment is controlled by the Encrypt/Decrypt bit.

The Encrypt/Decrypt bit (M_4) instructs the DCP algorithm processor to encrypt or decrypt the data from the Input Register using the ciphering method specified by the Cipher Type bits. The Encrypt/Decrypt bit also controls data flow within the DCP. For example, when the configuration bits are "01" (Dual Port, Master Clear, Slave Encrypted) and the Encrypt/Decrypt bit is "1" (encrypt), clear data will flow into the DCP through the Master Port and encrypted data will flow out through the Slave Port. When the Encrypt/Decrypt bit is set to "0" (decrypt), data flow reverses.

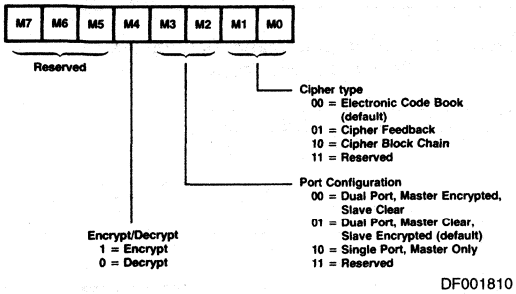


Figure 5. Mode Register Bit Assignments

Input Register

The 64-bit, write-only Input Register is organized to appear to the user as eight bytes of push down storage. A status circuit monitors the number of bytes that have been stored. The register is considered empty when the data stored in it has been or is being processed; it is considered full when one byte of data has been entered in Cipher Feedback or when eight bytes of data have been entered in the Electronic Code Book or Cipher Block Chain. If the user attempts to write data into the Input Register when it is full, the Input Register will disregard the attempt; no data in the register will be destroyed.

Output Register

The 64-bit, read-only Output Register is organized to appear to the user as eight bytes of pop-up storage. A status circuit detects the number of bytes stored in the Output Register. The register is considered empty when all the data stored in it has been read out by the host CPU, and is considered full if it still contains one or more bytes of output data. If a user attempts to read data from the Output Register when it is empty, the buffers driving the output bus will remain in a three-state condition.

The following multibyte registers cannot be directly addressed, but are loaded or read in response to commands written to the Command Register. (See Commands.)

M, E, D Key Registers

There are three 64-bit, write-only key registers in the DCP: the Master (M) Key Register; the Encrypt (E) Key Register; and the Decrypt (D) Key Register. The Master Key can be loaded only with clear data through the Auxiliary Port. The Encrypt and Decrypt Keys can be loaded in any of four ways: (1) as clear data through the Auxiliary Port; (2) as clear data through the Master Port; (3) as encrypted data through the Auxiliary Port; or (4) as encrypted data through the Master Port. In the last two cases, the encrypted data is first routed to the Input Register, decrypted using the M Key, and finally written to the target key register from the Output Register.

Initializing Vector Registers

Two 64-bit registers are provided to store feedback values for Cipher Feedback and Block Chained ciphering methods. One Initializing Vector (IVE) Register is used during encryption; the other (IVD), during decryption. Both registers can be loaded with either clear or encrypted data through the Master Port (in the latter case, the data is decrypted before being loaded into the IV Register), and both may be read out either clear or encrypted through the Master Port. (See Commands.)

Encrypt/ Decrypt M4	Port Configuration M3	M2	Input Register Flag	Output Register Flag
0	0	0	$\overline{MFLG}$	$\overline{SFLG}$
0	0	1	$\overline{SFLG}$	$\overline{MFLG}$
0	1	0	$\overline{MFLG}$	$\overline{SFLG}$
1	0	0	$\overline{SFLG}$	$\overline{MFLG}$
1	0	1	$\overline{MFLG}$	$\overline{SFLG}$
1	1	0	$\overline{MFLG}$	$\overline{SFLG}$

Figure 6. Association of Master Port Flag ($\overline{MFLG}$) and Slave Port Flag ($\overline{SFLG}$) with Input and Output Registers

Commands

All operations of the DCP result from command inputs, which are entered in Multiplexed Control Mode by writing a command byte to the Command Register. Command inputs are entered in Direct Control Mode by raising and lowering the logic levels on the AUX_7-K/$\overline{D}$, AUX_6-E/$\overline{D}$ and AUX_5-S/$\overline{S}$ pins. Figure 2 shows all commands that may be given in Multiplexed Control Mode. Figure 3 shows that subset executable in Direct Control Mode.

Load Clear M Key Through Auxiliary Port (90_H)

Load Clear E Key Through Auxiliary Port (91_H)

Load Clear D Key Through Auxiliary Port (92_H)

These commands override the data flow specifications set in the Mode Register and cause the Master (M), Encrypt (E), or Decrypt (D) Key Register to be loaded with eight bytes written to the Auxiliary Port. After the Load command is written to the Command Register, the Auxiliary Port Flag ($\overline{AFLG}$) will go

Refer to page 7-1 for Essential Information on Military Devices

active (LOW) and the corresponding bit in the Status Register (S_2) will go to "1," indicating that the device is able to accept key bytes at the Auxiliary Port pins. Additionally, the Command Pending bit (S_6) will go to "1" during the entire loading process.

Each byte is written by placing an active LOW signal on the Auxiliary Port Strobe ($\overline{ASTB}$) once data has been set up on the Auxiliary Port pins. The actual write process occurs on the rising (trailing) edge of $\overline{ASTB}$. (See Switching Characteristics for exact set-up, strobe width, and hold times.)

The Auxiliary Port Flag ($\overline{AFLG}$) will go inactive immediately after the eighth strobe goes active (LOW). However, the Command Pending bit (S_6) will remain "1" for several more clock cycles, until the key loading process is completed. All key bytes are checked for correct (odd) parity as they are entered (see Parity Checking).

Load Clear E Key Through Master Port (11_H)

Load Clear D Key Through Master Port (12_H)

These commands are available in both Multiplexed Control and Direct Control Modes. They override the data flow specifications set in the Mode Register and attach the Master Port inputs to the Encrypt (E) or Decrypt (D) Key Register, as appropriate, until eight key bytes have been written. In Multiplexed Control Mode, the command is initiated by writing the Load command to the Command Register. In Direct Control Mode, the command is initiated by raising the AUX_7-K/$\overline{D}$ control input while the AUX_5-S/$\overline{S}$ input is LOW. In this later case, the level on AUX_6-E/$\overline{D}$ determines which key register is written (HIGH = E Register).

Once the command has been recognized, the Command Pending bit (S_6 in the Status Register) will go to "1," and in Direct Control Mode, AUX_3-$\overline{CP}$ will go active (LOW), indicating that key entry may proceed. The host system then writes exactly eight bytes to the Master Port (at the Input Register address in Multiplexed Control Mode). When the key register has been loaded, Command Pending will return to "0," and in Direct Control Mode, the AUX_3-$\overline{CP}$ output will go inactive, indicating that the DCP can accept the next command.

Load Encrypted E Key Through Auxiliary Port ($B1_H$)

Load Encrypted D Key Through Auxiliary Port ($B2_H$)

Execution of these commands (in Multiplexed Control Mode only) is similar to the Load Clear E (or D) Key Through Auxiliary Port, except that key bytes are first decrypted using the Electronic Code Book algorithm and the Master (M) Key, and then loaded into the appropriate key register after having passed through the parity check logic (see Parity Checking).

The Command Pending bit (S_6) will be "1" during the entire decrypt-and-load operation. In addition, the Busy bit (S_5) will be "1" during the actual decryption process.

Load Encrypted E Key Through Master Port (31_H)

Load Encrypted D Key Through Master Port (32_H)

These commands (in Multiplexed Control Mode only) are similar in effect to Load Clear E (or D) Key Through Master Port, except that key bytes are initially decrypted using the Electronic Code Book algorithm and the Master (M) Key, and then loaded byte-by-byte into the target key register after having passed through the parity check logic (see Parity Checking).

The Command Pending bit (S_6) will be "1" during the entire necrypt-and-load operation. In addition, the Busy bit (S_5) will be "1" during the actual decryption process.

Load Clear IVE Register Through Master Port (85_H)

Load Clear IVD Register Through Master Port (84_H)

These commands (in Multiplexed Control Mode only) are virtually identical to Load Clear E (or D) Key Through Master Port except that the data written to the Input Register address is routed to the Encryption Initializing Vector (IVE) or Decryption Initializing Vector (IVD) Register instead of a key register, and no parity checking occurs. Command Pending (S_6) is a "1" during the entire loading process.

Load Encrypted IVE Register Through Master Port ($A5_H$)

Load Encrypted IVD Register Through Master Port ($A4_H$)

These commands are analogous to the Load Encrypted E (or D) Key Through Master Port commands. The data flow specifications set in the Mode Register are overridden, and the eight vector bytes are decrypted using the Decryption (D) Key and the Electronic Code Book algorithm. The resulting clear vector bytes are loaded into the target Initializing Vector Register, and no parity checking occurs. The Busy bit (S_5) does not go to "1" during the decryption process, but Command Pending (S_6) will be "1" during the entire decryption-and-load operation.

Read Clear IVE Register Through Master Port ($8D_H$)

Read Clear IVD Register Through Master Port ($8C_H$)

The effect of these commands (in Multiplexed Control Mode only) is to override the data flow specifications set in the Mode Register and to connect the appropriate Initializing Vector Register to the Master Port at the Output Register address. In this state, each IV Register appears as eight bytes of FIFO storage. The first byte of data will be available 6 clocks after the loading of the command register. The Command Pending bit will be set to "1" and will remain a "1" until sometime after the eighth byte is read out. The host system has the responsibility to read out exactly eight bytes.

Read Encrypted IVE Register Through Master Port ($A9_H$)

Read Encrypted IVD Register Through Master Port ($A8_H$)

The effect of these commands (in Multiplexed Control Mode only) is to override the specifications set in the Mode Register and to encrypt the contents of the specified Initializing Vector Register using the Electronic Code Book algorithm and the Encrypt (E) Key. The resulting cipher text is placed in the Output Register from which it can be read out as eight bytes through the Master Port. During the actual encryption process, the Busy bit (S_5) will be "1." When Busy goes to "0," the encrypted vector bytes are ready to be read out. Command Pending (S_6) will be "1" during the entire encryption-and-output process and will go to "0" when the eighth byte is read out. The host system is responsible for reading out exactly eight bytes.

Encrypt with Master (M) Key (39_H)

This command, in Multiplexed Control Mode only, overrides the data flow specifications set in the Mode Register and causes the DCP to accept eight bytes from the Master Port, which are written to the Input Register. When eight bytes have been received, the DCP encrypts the input using the Master (M) Key. The encrypted data is loaded into the Output Register, where it may be read out through the Master Port. The Command Pending (S_6) and Busy (S_5) bits are used to sense the three phases of this operation. Command Pending goes to "1" as soon as the Input Register can accept data. When exactly eight bytes have been entered, the Busy bit will go to "1" until the encryption process is complete.

When Busy goes to "0," the encrypted data is available to be read out. Command Pending will return to "0" when the eighth byte has been read.

Refer to page 7-1 for Essential Information on Military Devices

Start Encryption (41_H)

Start Decryption (40_H)

Start ($C0_H$)

The three "Start" commands begin normal data ciphering by setting the Start/Stop bit (S_7) in the Status Register to "1." The Start Encryption and Start Decryption commands explicitly specify the ciphering direction by forcing the Encrypt/Decrypt bit (M_4) in the Mode Register to "1" or "0," respectively; whereas, Start uses the current state of the Encrypt/Decrypt bit, as specified in a previous Mode Register load.

When a Start command has been entered, the Port Status Flag ($\overline{MFLG}$ or $\overline{SFLG}$) associated with the Input Register will become active (LOW), indicating that data may be written to the Input Register to begin ciphering.

In Direct Control Mode, the Start command is issued by raising the level on the AUX_5-S/$\overline{S}$ input (see Figure 3). The ciphering direction is specified by the level on AUX_6-E/$\overline{D}$. If AUX_6-E/$\overline{D}$ is HIGH when AUX_5-S/$\overline{S}$ goes HIGH, the command is Start Encryption; if AUX_6-E/$\overline{D}$ is LOW, it is Start Decryption.

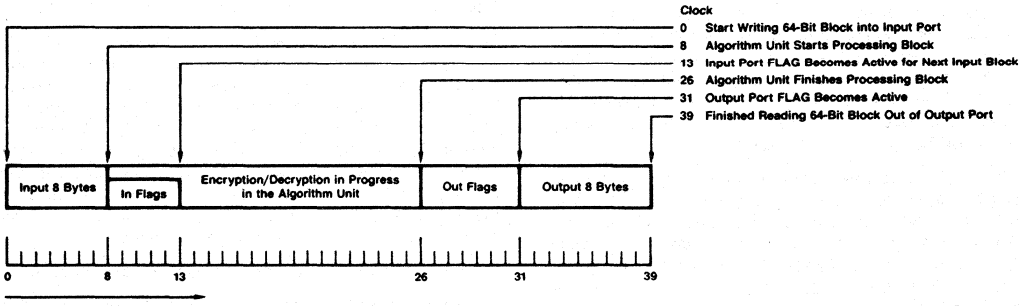
Stop ($E0_H$)

The Stop command clears the Start/Stop bit (S_7) in the Status Register to "0." This causes the input flag ($\overline{MFLG}$ or $\overline{SFLG}$) to become inactive and inhibits the loading of any further input into the Algorithm Unit. If ciphering is in progress (Busy bit (S_5) is "1" or AUX_2-$\overline{BSY}$ is active), it will finish, and any data in the Output Register will remain accessible (except in CFB Mode). In CFB Mode, the last byte of data must be read out before issuing the STOP command.

In Direct Control Mode, the Stop command is implied when the signal level on the AUX_5-S/$\overline{S}$ input goes from HIGH to LOW (see Figure 3).

Software Reset (00_H)

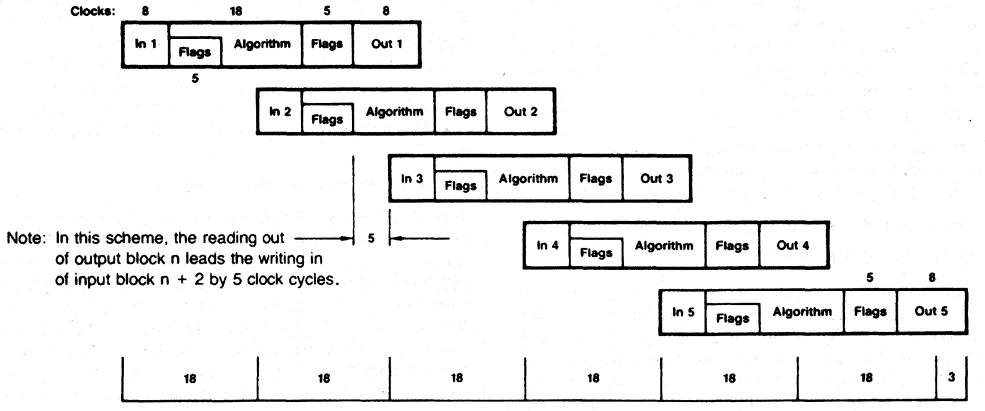
This command has the same effect as a hardware reset ($\overline{MRD}$ and $\overline{MWR}$ low): it forces the DCP back to its default configuration, and all processing flags go into Inactive Mode. The default configuration includes setting the Mode Register to Electronic Code Book Cipher Type and Dual Port Configuration with Master Port clear, Slave Port encrypted.



Detailed Timing of One Block

DF001820

Pipelining Scheme A: Minimum Timing Operation



DF001831

For n blocks, total number of clock pulses = (n + 1) x 18 + 3.

Am9568: Timing for Pipelined, Dual-Port Operation

Refer to page 7-1 for Essential Information on Military Devices

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin with
Respect to Ground..... −0.5 to +7.0V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Grade	T_A	V_{CC}	V_{SS}
Commercial	0°C to 70°C	5V ±5%	0V
Industrial	−40°C to 85°C	5V ±10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		−0.5		.8	Volts
V_{IH}	Input High Voltage		2.2		V_{CC}	Volts
V_{OL}	Output Low Voltage	I_{OL} = 3.2mA			.40	Volts
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4			Volts
I_I	Input Leakage Current	$V_{SS} \leqslant V_{IN} \leqslant V_{CC}$			±10	μA
I_{OZ}	Output Leakage Current	V_{SS} = +.40 $\leqslant V_{IN} \leqslant V_{CC}$			±10	μA
I_{CC}	Supply Current (AVER.)			150	250	mA

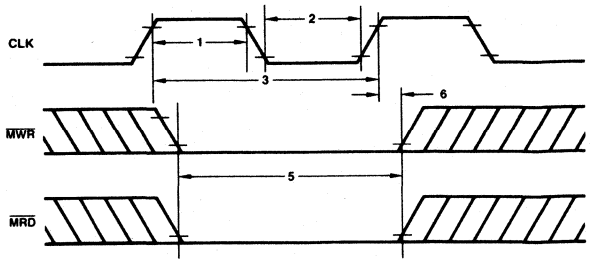
Refer to page 7-1 for Essential Information on Military Devices

Am9568 SWITCHING CHARACTERISTICS (Note 1)

The table below specifies the guaranteed performance of this device over the commercial operating range of 0 to +70°C with V_{CC} from 4.75V to 5.25V. All data are in nanoseconds. Switching tests are made with inputs and outputs measured at 0.8V for a LOW and 2.0V for a HIGH. Outputs are fully loaded, with $C_L \geq 50pF$. See Switching Waveform figures following table for graphic illustration of timing parameters.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

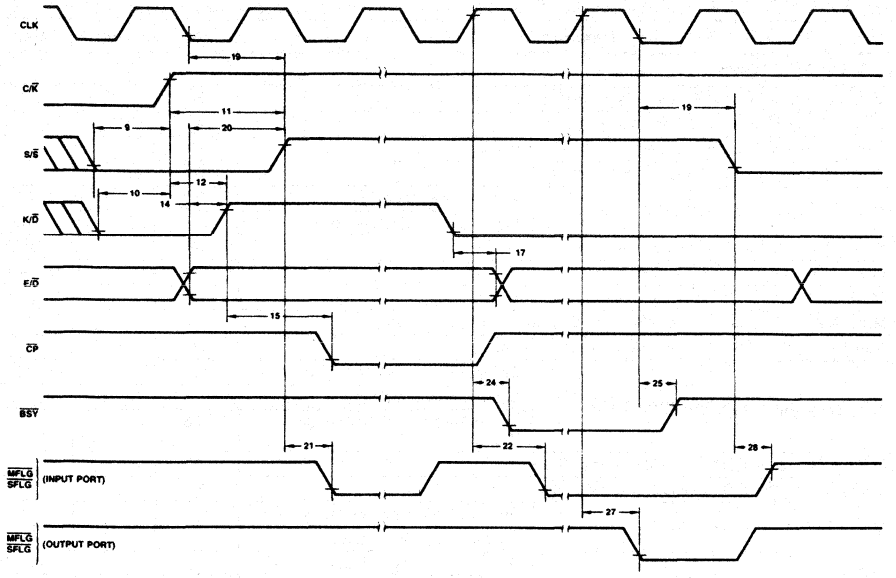
Number	Description	Min	Typ	Max	Units
Clock					
1	Clock Width HIGH (TWH)	115			ns
2	Clock Width LOW (TWL)	115			ns
3	Clock HIGH to Next Clock HIGH (Clock Cycle, TC)	250		1000	ns
Reset					
5	$\overline{MRD} \cdot \overline{MWR}$ LOW to $\overline{MRD} \cdot \overline{MWR}$ HIGH (Reset Pulse Width) (Note 12)	TC			ns
6	Clock HIGH to $\overline{MRD} \cdot \overline{MWR}$ HIGH	0		50	ns
Direct Control Mode					
9	S/$\overline{S}$ LOW to C/$\overline{K}$ HIGH (Set-up) (Note 12)	3TC			ns
10	K/$\overline{D}$ LOW to C/$\overline{K}$ HIGH (Set-up) (Note 12)	3TC			ns
11	C/$\overline{K}$ HIGH to S/$\overline{S}$ HIGH (Note 12)	6TC			ns
12	C/$\overline{K}$ HIGH TO K/$\overline{D}$ HIGH (Note 12)	6TC			ns
14	E/$\overline{D}$ VALID to K/$\overline{D}$ HIGH (Set-up) (Note 12)	3TC			ns
15	K/$\overline{D}$ HIGH to $\overline{CP}$ LOW			300	ns
17	K/$\overline{D}$ LOW to E/$\overline{D}$ INVALID (Hold) (Note 12)	TC			ns
19	Clock LOW to S/$\overline{S}$ VALID	20		80	ns
20	E/$\overline{D}$ VALID to S/$\overline{S}$ HIGH (Setup) (Note 12)	3TC			ns
21	S/$\overline{S}$ HIGH to $\overline{MFLG}$ ($\overline{SFLG}$) LOW (Port Input Flag)			230	ns
22	Clock HIGH to $\overline{MFLG}$ ($\overline{SFLG}$) LOW (Port Input Flag) (Note 2)			230	ns
24	Clock HIGH to $\overline{BSY}$ LOW			300	ns
25	Clock LOW to $\overline{BSY}$ HIGH			230	ns
27	Clock HIGH to $\overline{MFLG}$ ($\overline{SFLG}$) LOW (Port Output Flag)			230	ns
28	S/$\overline{S}$ LOW to $\overline{MFLG}$ ($\overline{SFLG}$) HIGH (Port Input Flag) (Note 3)			230	ns
Multiplexed Control Mode – Master Port					
32	MALE Width (HIGH)	40			ns
34	$\overline{MCS}$ LOW to MALE LOW (Set-up)	0			ns
35	MALE LOW to $\overline{MCS}$ HIGH (Hold)	30			ns
36	Address INVALID to MALE LOW (Address Set-up Time)	15			ns
37	MALE LOW to Address INVALID (Address Hold Time)	25			ns



WF004490

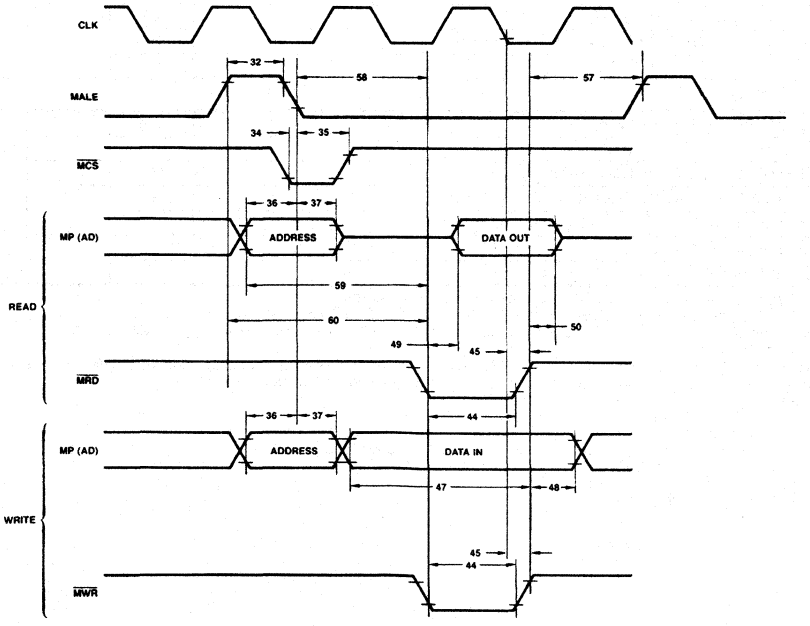
CLOCK AND RESET

Refer to page 7-1 for Essential Information on Military Devices



WF004500

CONTROL AND STATUS SIGNALS (DIRECT CONTROL MODE)



WF004510

MASTER PORT, MULTIPLEXED CONTROL MODE READ/WRITE

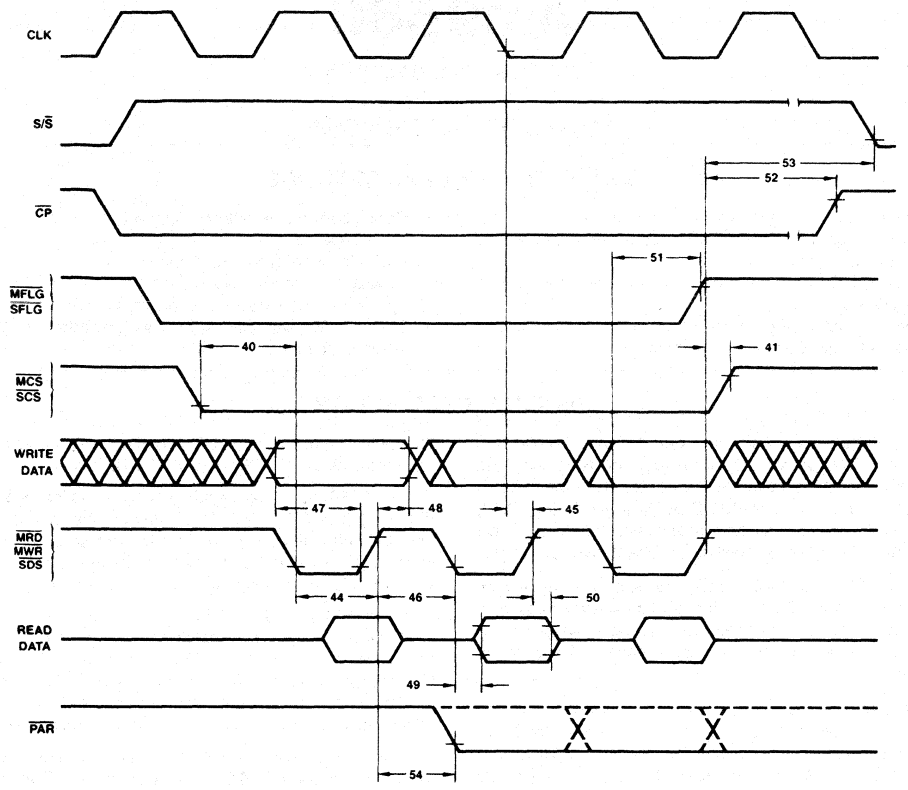
Refer to page 7-1 for Essential Information on Military Devices

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Descripton		Min	Typ	Max	Units
Master/Slave Port Read/Write						
40	$\overline{MCS}$ LOW to $\overline{MRD}$, $\overline{MWR}$ LOW (Select Set-up) (Note 4)		60			ns
	$\overline{SCS}$ LOW to $\overline{SDS}$ LOW (Select Set-up) (Note 4)		100			
41	$\overline{MRD}$, $\overline{MWR}$ HIGH to $\overline{MCS}$ HIGH (Select Hold) (Note 4)		50			ns
	$\overline{SDS}$ HIGH to $\overline{SCS}$ HIGH (Select Hold) (Note 4)		25			
44	$\overline{MRD}$, $\overline{MWR}$ LOW to $\overline{MRD}$, $\overline{MWR}$ HIGH	Width – Write, Data Read	150		1000	ns
		Width – Status Register Read	215		1000	
	$\overline{SDS}$ LOW to $\overline{SDS}$ HIGH (Read, Write)		125		1000	
45	Clock LOW to $\overline{MRD}$, $\overline{MWR}$ HIGH (Note 11)		0		TWL – 85	ns
	Clock LOW to $\overline{SDS}$ HIGH (Note 11)		0		TWL – 65	
46	$\overline{MRD}$, $\overline{MWR}$ HIGH to $\overline{MRD}$, $\overline{MWR}$ LOW (Data Strobe Recovery Time)		150			ns
	$\overline{SDS}$ HIGH to $\overline{SDS}$ LOW (Data Strobe Recovery Time)		125			
47	Write Data VALID to $\overline{MWR}$ ($\overline{SDS}$) HIGH	Set-up Time – Key Load (Note 8)	125			ns
		Set-up Time – Data Write	125			
		Set-up Time – Command/Mode Register Write	125			
48	$\overline{MWR}$ HIGH to Write Data INVALID (Hold Time)		20			ns
	$\overline{SDS}$ HIGH to Write Data INVALID (Hold Time)		25			
49	$\overline{MRD}$ LOW to Read Data VALID	Read Access Time – Status Register			215	ns
		Read Access Time – Data			150	
	$\overline{SDS}$ LOW to Read Data VALID	Read Access Time – Status Register			200	ns
		Read Access Time – Data			120	
50	$\overline{MRD}$ ($\overline{SDS}$) HIGH to Read Data INVALID (Hold Time)		5		85	ns
51	$\overline{MRD}$, $\overline{MWR}$ LOW to $\overline{MFLG}$ ($\overline{SFLG}$) HIGH (Last Strobe) (Note 5)				150	ns
	$\overline{SDS}$ LOW to $\overline{SFLG}$ HIGH (Last Strobe) (Note 5)				125	
52	$\overline{MWR}$ HIGH to $\overline{CP}$ HIGH (Note 4), (Note 12) (Last Strobe – Key Load)				TC – 520	ns
53	$\overline{MRD}$, $\overline{MWR}$ ($\overline{SDS}$) HIGH to S/$\overline{S}$ LOW (Hold Time) (Note 10), (Note 12)		4TC			ns
54	$\overline{MWR}$ HIGH to $\overline{PAR}$ VALID (Key Write)				220	ns
57	$\overline{MRD}$, $\overline{MWR}$ HIGH to MALE HIGH		75			ns
58	MALE LOW to $\overline{MRD}$, $\overline{MWR}$ LOW		25			ns
59	Address Valid to $\overline{MRD}$, $\overline{MWR}$ LOW (to guarantee 49)		100			ns
60	MALE HIGH to $\overline{MRD}$, $\overline{MWR}$ LOW		100			ns
Auxiliary Port Key Entry						
61	$\overline{ASTB}$ LOW to $\overline{ASTB}$ HIGH (Width)		160			ns
62	Clock LOW to $\overline{ASTB}$ HIGH (Note 11)		0		TWL – 65	ns
63	$\overline{ASTB}$ HIGH to Next $\overline{ASTB}$ LOW (Recovery Time)		250			ns
64	Write-Data VALID to $\overline{ASTB}$ HIGH (Data Set-up Time)		200			ns
65	$\overline{ASTB}$ HIGH to Write-Data INVALID (Data Hold Time)		80			ns
66	$\overline{ASTB}$ HIGH to $\overline{PAR}$ VALID				200	ns
67	$\overline{ASTB}$ LOW to $\overline{AFLG}$ HIGH (Last Strobe)				230	ns

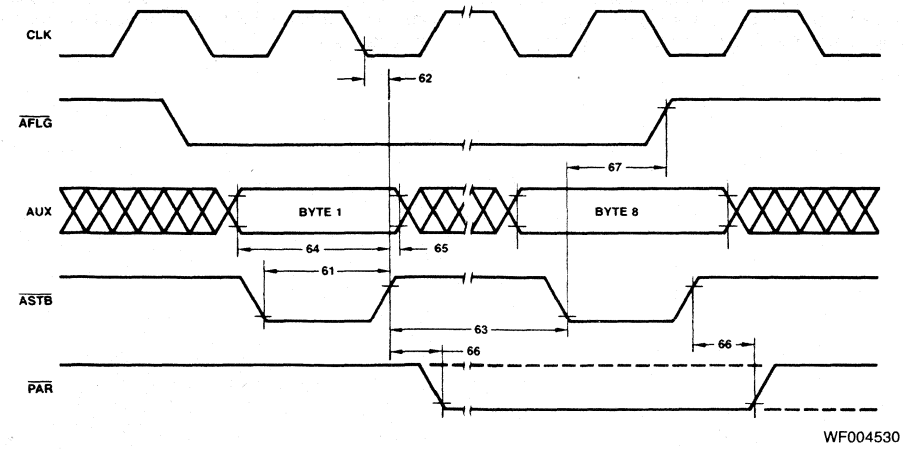
Notes:
1. All input transition times assumed ⩽ 20ns, except clock which is ⩽ 10ns.
2. Parameter 22 applies to all input blocks except the first (when S/$\overline{S}$ first goes HIGH).
3. When S/$\overline{S}$ goes inactive (LOW) in Direct Control Mode, the flag associated with the input port will turn off.
4. Direct Control Mode only.
5. In Cipher Feedback, the Port Flag ($\overline{MFLG}$ or $\overline{SFLG}$) will go inactive following the leading edge of the first data strobe ($\overline{MRD}$, $\overline{MWR}$ or $\overline{SDS}$); in all other modes and operations, the flags go inactive on the eighth data strobe.
6. Do not remove K/$\overline{D}$ until CP is inactive (HIGH).
7. Do not change E/$\overline{D}$ until $\overline{MFLG}$ ($\overline{SFLG}$) is inactive (HIGH).
8. 200ns min if parity check is needed.
9. In Cipher Feedback, $\overline{BSY}$ must be inactive (HIGH) before S/$\overline{S}$ goes inactive (LOW).
10. $\overline{AFLG}$ must go active (LOW) before $\overline{ASTB}$ goes inactive (LOW).
11. TWL is the clock width (LOW) (number 2).
12. TC is the clock cycle time (number 3).

Refer to page 7-1 for Essential Information on Military Devices



WF004520

MASTER (SLAVE) PORT READ/WRITE



WF004530

AUXILIARY-PORT KEY ENTRY

Refer to page 7-1 for Essential Information on Military Devices

Am9580

Hard Disk Controller (HDC)

ADVANCED INFORMATION

DISTINCTIVE CHARACTERISTICS

- Linked list command and data structure
- Controls up to 4 drives, any combination of hard or floppy disk drives
- ST506/412 and user-defined disk interface options supported
- Two on-chip sector buffers are programmable for sector sizes of 128, 256, and 512 bytes
- Error checking algorithms supported include
  - CRC/CCITT
  - Single burst Reed-Solomon
  - Double burst Reed-Solomon
  - External ECC (Error Correcting Code)
- On-chip DMA capability supports 32-bit addressing in 8/16-bit systems

GENERAL DESCRIPTION

The Am9580 Hard Disk Controller (HDC) is single chip solution to the problems encountered in designing Data Formatters and Disk System Controllers. Together with its companion part, the Am9581 Disk Data Separator (DDS), the Am9580 provides all the functions which until now have been found only on sophisticated board-level products.

The Am9580 has been designed with the necessary flexibility to cope with the differing requirements of today's broad marketplace while still retaining the advanced technology and innovative features that tomorrow's market will demand.

The Am9580 supports both Rigid and Flexible Disk Drives and their accompanying Data Formats. The Am9580 can control up to four drives, allowing any mix of Rigid and Flexible drives. As the characteristics of each drive are independently user-programmable, the system designer is provided with the flexibility needed to control any disk drive.

A sophisticated on-chip DMA Controller fetches the commands, writes status information, fetches data to be written on disk and writes data read from disk. The DMA operation is programmable to adjust the bus occupancy, data bus width (8-bit or 16-bit), and wait state insertion. Two sector buffers allow zero sector interleaving to access data on physically contiguous sectors, improving both file access time and system throughput. The buffers are programmable for sector sizes of 128, 256, and 512 bytes.

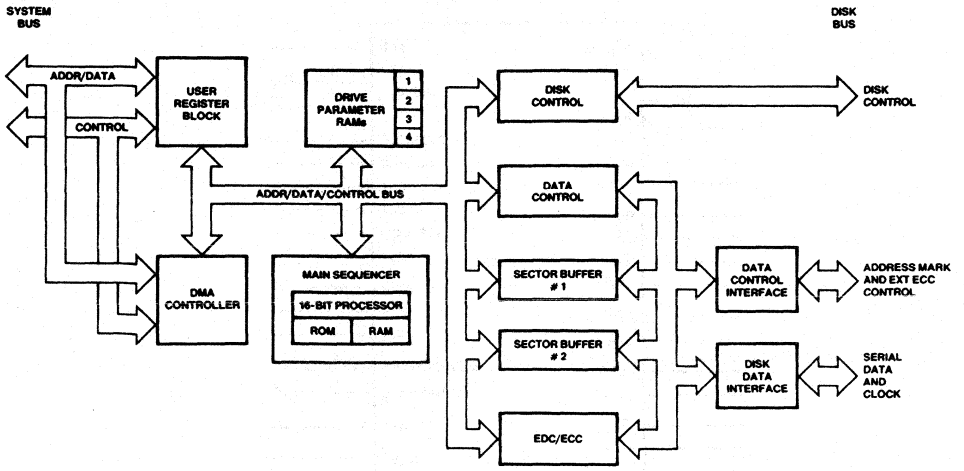
The Am9580 insures data integrity by selecting one error detecting code (CRC-CCITT) or one of two error correcting codes (single and double burst Reed-Solomon). Additionally, the HDC provides handshake signals to control external ECC circuitry to implement any ECC algorithm.

The Am9580 provides signals which are necessary to control external Encode/Decode and Address Mark circuitry such as found on the Am9581. By partitioning the Disk Control System in this way, future developments in the field of Data Encoding can still take advantage of the HDC's advanced Data Formatting and Control capabilities.

The flexible, user-programmable disk interface can be configured to control ST506/412 or standard Floppy Disk interfaces with only a few buffers required and can easily be adapted to other interface standards.

The Am9580 provides a comprehensive, high level command set for multi-sector disk I/O, marginal data recovery, diagnostics and error recovery. Commands may be linked together to be executed sequentially by the Am9580 without any host intervention. This linked list command structure also simplifies command insertion, deletion, or rearrangement.

BLOCK DIAGRAM



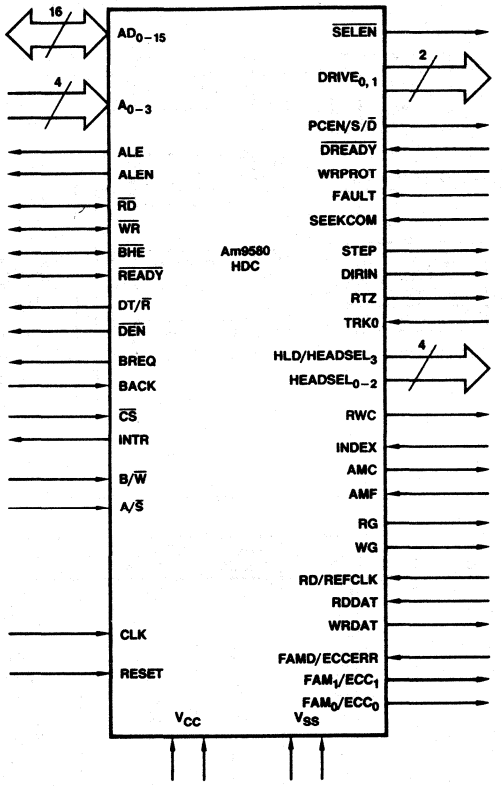
BD004030

Function	PGA Pin Number	LCC Pin Number
AD15	B1	1
AD14	B2	2
AD13	C1	3
AD12	C2	4
AD11	D1	5
AD10	D2	6
AD9	E1	7
VCC	E2	8
AD8	F1	9
AD7	F2	10
AD6	G1	11
AD5	G2	12
AD4	H1	13
AD3	H2	14
AD2	J1	15
AD1	J2	16
AD0	K1	17
$\overline{\text{READY}}$	L2	18
CLK	K2	19
RESET	L3	20
ALEN	K3	21
ALE	L4	22
$\overline{\text{CS}}$	K4	23

Function	PGA Pin Number	LCC Pin Number
$\overline{\text{WR}}$	L5	24
$\overline{\text{RD}}$	K5	25
VSS	L6	26
DT/$\overline{\text{R}}$	K6	27
$\overline{\text{DEN}}$	L7	28
$\overline{\text{BHE}}$	K7	29
A0	L8	30
A1	K8	31
A2	L9	32
A3	K9	33
BREQ	L10	34
BACK	K11	35
INTR	K10	36
A/$\overline{\text{S}}$	J11	37
B/$\overline{\text{W}}$	J10	38
INDEX	H11	39
FAMD/ECCERR	H10	40
RDDAT	G11	41
WRDAT	G10	42
VCC	F11	43
RD/REFCLK	F10	44
FAM0/ECC0	E11	45
FAM1/ECC1	E10	46

Function	PGA Pin Number	LCC Pin Number
RG	D11	47
WG	D10	48
AMC	C11	49
AMF	C10	50
RWC	B11	51
HLD/HEADSEL3	A10	52
HEADSEL2	B10	53
HEADSEL1	A9	54
HEADSEL0	B9	55
PCEN/S/$\overline{\text{D}}$	A8	56
RTZ	B8	57
DIRIN	A7	58
STEP	B7	59
VSS	A6	60
SEEKCOMB	B6	61
FAULT	A5	62
WRPROT	B5	63
$\overline{\text{DREADY}}$	A4	64
TRK0	B4	65
DRIVESEL1	A3	66
DRIVESEL0	B3	67
$\overline{\text{SELEN}}$	A2	68

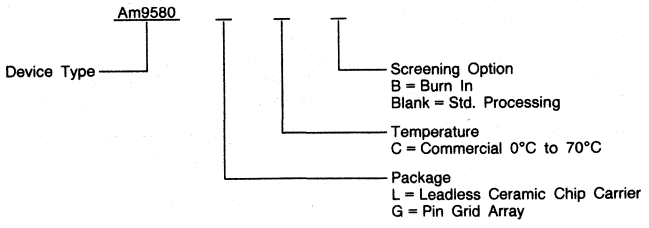
LOGIC SYMBOL



LS001351

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9580	GC (68 leads PGA) LC (68 LCC)

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

LCC Pin No.	Name	I/O	Description
SYSTEM INTERFACE LINES			
8, 43	V_{CC1}, V_{CC2}		+5V Power Supply.
26, 60	V_{SS1}, V_{SS2}		Ground.
19	CLK	I	System Clock. CLK is a TTL-compatible clock input to time DMA transfers and disk control operations (seeks).
25	$\overline{RD}$	I/O	Read (Active-Low). Read is a bidirectional, active-low, three-state signal. A LOW on this line indicates that the CPU or HDC is performing an I/O or memory read cycle. When the HDC is in Slave Mode, this is an input signal used by the CPU to read the internal registers of the HDC. When the HDC is the bus master, this signal is an output used by the HDC to access data from memory. $\overline{RD}$ and $\overline{WR}$ must not be active simultaneously.
24	$\overline{WR}$	I/O	Write (Active-Low). Write is a bidirectional, active-low, three-state signal. A LOW indicates that the CPU or HDC is performing an I/O or memory write cycle. When the HDC is in Slave Mode, it is an input signal used by the CPU to load the internal registers of the HDC. When the HDC is Bus Master, this signal is an output used by the HDC to write data into system memory. $\overline{RD}$ and $\overline{WR}$ must not be active simultaneously.
22	ALE	O	Address Latch Enable (Active-High). The trailing edge of ALE latches the lower address word (A_0 to A_{15}) into the external address latch.
21	ALEN	O	Upper Address Latch Enable (Active-High). The trailing edge of ALEN latches the upper address work (A_{16} to A_{31}) into the external address latch. The HDC executes Upper Address latch cycles as necessary to minimize bus occupancy. The address is updated when a new command is executing or whenever the upper address for the current bus cycle differs from the address of the previous bus cycle.
27	DT/$\overline{R}$	O	Data Transmit/Receive (Three-State). When the HDC is bus master, a HIGH on this output indicates that data is being transmitted from the HDC. A LOW on this output indicates that data is being transferred into the HDC. This output is three-stated when the HDC is not in control of the system bus.
28	$\overline{DEN}$	O	Data Enable (Active-Low, Three-State). When the HDC is bus master, a LOW on this output indicates that data is driven on the Address/Data bus or the Address/Data bus is three-stated for receiving data. This output is three-stated when the HDC is not in control of the system bus.
29	$\overline{BHE}$	I/O	Byte High Enable (Active-Low). $\overline{BHE}$ is a bidirectional, active-low signal to enable data onto the most significant byte of the data bus ($AD_{15} - AD_8$). $\overline{BHE}$ is LOW when data is to be accessed on the high portion of the bus. $\overline{BHE}$ / A_0 / Size L / L / Whole word L / H / Upper byte H / L / Lower byte H / H / Reserved When the HDC is a bus master, this pin is an output. It is an input when the HDC is in Slave Mode. $\overline{BHE}$ is disabled and ignored when the HDC is strapped to a byte interface.
1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17	AD < 15:0 >	I/O	Address/Data Bus (Active-High). The Address/Data Bus is a time-multiplexed, bidirectional, three-state, 16-bit bus used for all system transactions. A logic HIGH on the bus corresponds to a "1" and a logic LOW corresponds to a "0." AD_0 is the least significant bit. The presence of an address is defined by ALE or ALEN. When ALE is HIGH, the bus contains the Lower Address bits A_0 to A_{15}. When ALEN is HIGH, the bus contains Upper Address bits A_{16} to A_{31}. The 32-bit address allows the HDC to direct linear addressing of 4 GBytes. The presence of data is indicated by $\overline{RD}$ and $\overline{WR}$. When the HDC is in Slave Mode, $\overline{RD}$ and $\overline{WR}$ may be asynchronous to the system clock (CLK). In Master Mode the Address/Data Bus is driven synchronously using a four-cycle bus transfer.
30	A_0	I	Address Line 0 (Active-High). This pin selects between the odd byte (HIGH) and the even byte (LOW) of the internal registers. It is used only when the HDC is in Slave Mode.
33, 21, 31	A < 3:1 >	I	Address Line 1 to 3 (Active-High). When the HDC is in Slave Mode, these lines are the address inputs selecting one of the internal registers. These lines are ignored when the HDC is bus master.
18	$\overline{READY}$	I/O	Ready (Open Drain, Active-Low). When the HDC is bus master, this is an input to extend the bus cycle for slow memories and peripheral devices. When the HDC is in Slave Mode, this is an output indicating that the HDC is ready to complete the bus transfer. The $\overline{READY}$ input may be synchronous or asynchronous depending on the programming of the A/$\overline{S}$ strap.
34	BREQ	O	Bus Request (Active-High). The HDC activates BREQ to request the control of the system bus.
35	BACK	I	Bus Acknowledge (Active-High). BACK acknowledges the bus request of the HDC, indicating that the CPU has relinquished the system bus to the HDC. Since BACK is internally synchronized, transitions on BACK do not have to be synchronous with the HDC clock. BACK may be removed at any time to make the HDC release the bus. If the HDC DMA is preempted by removing BACK, the HDC will release BREQ for two clock cycles so that external devices may gain the mastership on the system bus.
36	INTR	O	Interrupt Request (Active-High). INTR is activated when the HDC requires service by the CPU. Interrupt Request is reset whenever the status half of the Status/Command Register (SCR) is accessed.

PIN DESCRIPTION (Cont.)

LCC Pin No.	Name	I/O	Description
23	$\overline{CS}$	I	Chip Select (Active-Low). The host processor activates $\overline{CS}$ to enable a Slave Mode access to read or write the internal registers of the HDC. This pin is ignored when the HDC is bus master.
38	B/$\overline{W}$	I	Byte/$\overline{Word}$ Strap. This pin selects either a byte (8-bit) or word (16-bit) interface. When a byte interface is selected, only AD < 7:0 > are used for data transfers and all operations are then byte operations. This pin must be strapped to either V_{CC} or V_{SS}. V_{CC} = byte interface V_{SS} = word interface
37	A/$\overline{S}$	I	Asynchronous/$\overline{Synchronous}$ Ready Strap. This line determines whether the $\overline{READY}$ input is synchronous or asynchronous. When this line is strapped HIGH, the HDC internally synchronizes the $\overline{READY}$ line to the system clock. When the line is strapped LOW, the HDC does not synchronize $\overline{READY}$. It must be synchronized to the system clock externally. V_{CC} = asynchronous $\overline{READY}$ V_{SS} = synchronous $\overline{READY}$
20	RESET	I	Reset (Active-High). When RESET is active, all outputs lines are inactive. Three-state outputs are floating. Inputs are ignored. A RESET pulse makes the chip enter the initialization procedure. Upon completion of initialization, an interrupt request will be issued. INTR will go High. If the user attempts to read from or write to the HDC prior to completion of reset, the ready line will remain inactive until the reset is completed, causing the CPU to wait.
DISK INTERFACE LINES			
68	$\overline{SELEN}$	O	Select Enable (Active-Low). $\overline{SELEN}$ enables the drive specified by DRIVE < 1:0 > . When LOW, DRIVE < 1:0 > are valid. When HIGH, no drive is selected. A selected drive should respond with $\overline{DREADY}$.
66, 67	DRIVE < 1:0 >	O	Drive Address (Active-High). These two lines in conjunction with $\overline{SELEN}$ determine the drive to be selected. The drive numbers are encoded as follows:
56	PCEN/S/$\overline{D}$	O	Precompensation Enable/Single/$\overline{Double}$ (Active-High). For all drive formats, this line signals when precompensation should be used. It enables precompensation if HIGH and disables precompensation if LOW. In addition, for floppy formats, it selects either the single- or double-density recording formats. The falling edge of $\overline{SELEN}$ may be used to latch Single/$\overline{Double}$. A latched LOW indicates double-density and a latched HIGH single-density.
64	$\overline{DREADY}$	I	Drive Ready (Active-Low). Drive Ready indicates that the currently selected drive is ready to Read, Write, or Seek. It must be asserted within 217 clocks after $\overline{SELEN}$ is asserted by the HDC. Once asserted by the selected drive, any negation of this line causes the current IOPB to be aborted. $\overline{DREADY}$ is ignored while $\overline{SELEN}$ is HIGH.
62	FAULT	I	Fault (Active-High). FAULT indicates a fault of the selected drive. Fault must be inactive as long as $\overline{DREADY}$ is active. If it is ever asserted by the selected drive, the current IOPB is aborted and the drive deselected.
63	WRPROT	I	Write Protect (Active-High). WRPROT is sampled after receipt of $\overline{DREADY}$ during any IOPB command which involves a write to the disk. It is ignored during "Read Only"-type commands.
61	SEEKCOM	I	Seek Complete (Active-High). SEEKCOM indicates that the drive is on track and that the heads have settled (head settling applies only to floppy disk drives). This line is sampled and verified High before any seek is issued. A HIGH indicates "Seek Complete." It is also used as an indication of head load completion for floppy drives.
59	STEP	O	Step (Active-High). The HDC pulses the STEP line to move the head from one track to the next track. The width and spacing of pulses is programmable. This allows easy upgrades to higher performance drives.
58	DIRIN	O	Direction In (Active-High). DIRIN indicates the direction the head should move on STEP pulses. When HIGH, the head should move toward higher track numbers (In, or towards the disk spindle). When LOW, it should move towards lower track numbers (Out).
57	RTZ	O	Return to Zero (Active-High). A pulse on the RTZ output should recalibrate the head to track 0. Optionally, the HDC may recalibrate the drive by issuing STEP pulses until track 0 is reached (TRK0 becomes active). The RTZ pulse has the same width as the STEP pulse. The drive should assert SEEKCOM as an indication of the completion of the requested recalibration.
65	TRK0	I	Track 0 (Active-High). The selected drive should assert TRK0 whenever the head is positioned over track 0.
52	HLD/ HEADSEL < 3 >	O	Head Load/Head Select < 3 > (Active-High). For floppy drives, this line functions as a head load signal. It is active as long as the head should be loaded. SEEKCOM is sampled eight clocks after the assertion of HLD. If LOW, the HDC will then wait for it to go HIGH. If HIGH, the HDC assumes that the heads are already loaded. For hard disk drives, this line is the most significant bit of the head number.
53, 54, 55	HEADSEL < 2:0 >	O	Head Select (Active-High). These lines output the lower three bits of the head number. The encoding is standard binary.
51	RWC	O	Reduced Write Current (Active-High). When the drive currently selected is configured for a floppy disk or ST506/412 interface, this line indicates that the head is positioned over an inner track where the Write Current should be reduced.

DRIVE < 1 >	DRIVE < 0 >	Drive Selected
L	L	Drive 0
L	H	Drive 1
H	L	Drive 2
H	H	Drive 3

05308A

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

LCC Pin No.	Name	I/O	Description
39	INDEX	I	Index (Active-High). INDEX marks each revolution of the disk. The HDC only notes the LOW-to-HIGH transition. INDEX should be valid as long as $\overline{\text{DREADY}}$ is asserted.
49	AMC	O	Address Mark Control (Active-High). The HDC asserts AMC in conjunction with RG or WG to command the data separator to write or detect address marks. When RG is asserted, the HDC wants the data separator to search for the next address/data mark. When WG is asserted, the HDC wants the data separator to write an address mark. In both cases, the data separator should acknowledge completion by asserting AMF.
50	AMF	I	Address Mark Found (Active-High). The data separator asserts AMF in response to AMC.
47	RG	O	Read Gate (Active-High). RG indicates that a disk read is in progress. It commands the Phase Locked Loop (PLL) of the data separator to lock the RD/REF CLK to the serial data from disk. This output changes synchronously with the RD/REF CLK.
48	WG	O	Write Gate (Active-High). WG indicates that the HDC is writing to the disk. When WG is asserted disk data is strobed out synchronously with RD/REF CLK.
44	RD/REFCLK	I	Read/Reference Clock. RD/REFCLK is the data clock for sampling read data and strobing out write data. It is assumed valid if the following conditions are met: $\overline{\text{SELEN}}$ = LOW, $\overline{\text{DREADY}}$ = LOW, SEEKCOM = HIGH, and FAULT = LOW. While valid, this clock should be free from any glitches.
41	RDDAT	I	Read Data. RDDAT is the NRZ data input from the disk. The HDC samples RDDAT with the rising edges of RD/REF CLK. Therefore, the data should be set up by the data separator at the falling edge of RD/REF CLK.
42	WRDAT	O	Write Data. WRDAT is the NRZ data output to the disk. Transitions occur on the falling edge of RD/REF CLK.
40	FAMD/ECCERR	I	Floppy Address Mark Detect/ECC Error (Active-High). This signal indicates a data read error when external ECC is used. For single-density floppy formats, it indicates detection of a deleted data mark.
46, 45	FAM < 1:0 > / ECC < 1:0 >	O	Floppy Address Mark/ECC Control (Active-High). These dual-function lines either control external ECC (hard disk format, external ECC enabled) or indicate the type of address/data mark to be used (single- and double-density floppy formats, AMC High). ECC Control: 00 = Idle — No operation in the external ECC. 01 = Reset — External ECC should reset and prepare itself for an operation. 11 = Generate — Whether reading or writing, the external ECC should strobe data in and generate a check sum. 10 = Check — When reading, this state indicates that the ECC should accept the check sum from the disk. When writing, it should gate the check sum to the disk. These states always proceed in the Gray code progression shown above: 00-01-11-10-00. The nominal state of the HDC is 00:Idle. Floppy Address Mark: Double-Density — 00 = Index Address Mark (IAM) XX = Data or Header Address Mark (DAM, IDA) Single Density — 00 = Index Address Mark (IAM) 01 = ID Address Mark (IDAM) 10 = Data Address Mark (DAM) 11 = Deleted Data Address Mark (DDAM) Floppy Address Mark takes precedence over ECC Control.

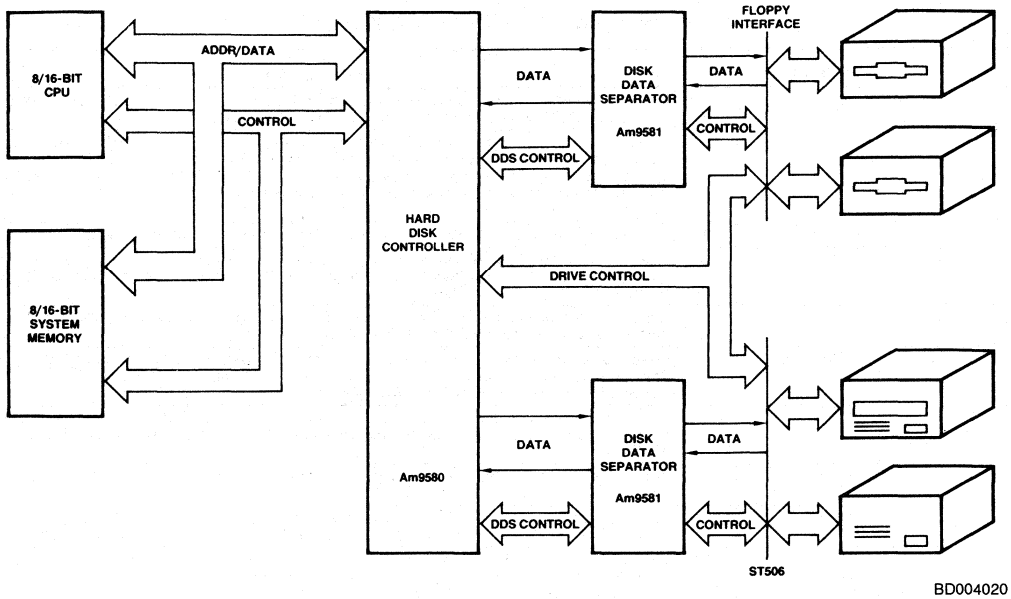


Figure 1. Disk Controller System

PRODUCT OVERVIEW

Hardware Architecture

The HDC supports two interfaces as shown in the block diagram in Figure 2; the system interface communicates with the host CPU and system memory, and the disk interface controls the data separator and the disk drives.

System Interface

The HDC is designed for easy interfacing to most 8-bit or 16-bit, multiplexed or demultiplexed, synchronous or asynchronous microprocessor buses. A strap pin programs the system interface for either byte (8-bit) or word (16-bit) mode. In Slave Mode, the host CPU can access the internal registers of the HDC. In Master Mode, the on-chip DMA controller controls the system bus.

DMA Controller

The on-chip DMA controller provides the HDC with the ability to execute complex disk I/O operations without intervention by the host CPU. The DMA controller scans the command chain stored in system memory, updates the Status Result Area when errors occur, and transfers the data between the internal sector buffers and system memory. Data may optionally be stored in noncontiguous memory to support linked-list data storage in word processing systems.

The DMA controller generates 32-bit linear addresses to access system memory of up to 4 GBytes directly. For multiple bus master systems, the DMA transfers can be throttled to dedicate only a certain share of the system bus bandwidth to the HDC. The Mode Register specifies the DMA burst length and dwell time. DMA bursts can be preempted by removing Bus Acknowledge (BACK). The HDC can insert a programmable number of software Wait States into DMA bus cycles. Additionally, hardware Wait States can be added via the $\overline{\text{READY}}$ input. The HDC updates the upper address word (A_{16} to A_{31}) when a carry out of the lower 16 bits indicates it is necessary.

User Registers

The Mode Register defines the operation of the DMA controller. The Status/Command Register controls the basic operation of the HDC itself. The Next Block Pointer (NBP) Register links to the first Input/Output Parameter Block (IOPB) of the command chain. The Status Result Pointer Register and the Status Result Length Register define the Status Result Area.

Main Sequencer

The main sequencer translates the high-level system commands into the control signals for the various independent functional sections of the HDC. The power of this 16-bit processor is used to ease the complex data manipulation burden of the system CPU.

Drive Parameter RAMs

The Drive Parameter RAMs store the specification parameters for individual drives that adapt the HDC to any combination of disk recording schemes. The contents can be altered at any time with a single IOPB. Once loaded these parameters allow disk commands to be independent of the drive format. For example, the write command is the same whether it is for a single-density floppy disk drive or a Winchester hard disk drive.

Error Checking

The HDC features two powerful Reed-Solomon error-correcting codes, as well as the industry-standard error-detection code, CRC-CCITT. It also supports user-definable, external error-correction schemes. These, along with the programmable retry and and correction attempt policy, allow maximum control of data integrity.

Sector Buffers

The HDC transfers data to or from disk without adding time constraints on the system bus bandwidth. The two internal sector buffers can be filled or emptied at any speed without interfering with the data transfer between the sector buffers and the disk. The two internal sector buffers can be toggled for zero sector interleave disk data operations. While one sector buffer is filled with data from disk, the other buffer is emptied by the DMA controller. Physically contiguous sectors on a track can thus be read or written on the fly.

Disk Control Interface

The lines of the Disk Control Interface conform to the ST506/412 disk drive interface standard. Other drive interface standards can be supported with some external circuitry. The ST506/412 interface selects one of the four disk drives and one of up to 16 heads. Any combination of floppy disk and hard disk drives can be connected. The Drive Parameter Blocks specify the seek dwell, seek pulse width, and head settle timing.

The HDC can perform implied and overlapped seeks. When the implied seek option is selected, the HDC automatically causes the appropriate seeks when issuing a read or write command. If this option is disabled, then the HDC should be issued separate seek commands before executing read or write commands.

When the overlapped seek option is selected, drives can seek in parallel, thus minimizing the seek overhead in multiple disk drive systems. After the HDC has issued a seek command to one drive and while this drive performs the seek, the HDC scans subsequent IOPBs for commands requiring seeks on other drives. If the HDC finds such commands, it issues seek commands to these drives to make them seek in parallel. On receiving a "Seek Complete" from the first drive, the HDC resumes execution of the command chain.

Disk Data Interface

The Disk Data Interface controls the Address Mark handshake with the data separator, controls the optional external ECC logic, and handles the serial data input and output. Operating asynchronously to the other blocks of the device, the Disk Data Interface is driven by the Read/Reference Clock (RD/REF CLK) generated by the data separator. The Disk Data Interface converts the data stored in the sector buffer into a serial bit stream for the disk or it deserializes the incoming bit stream to be loaded into one of the sector buffers. Non-data information, such as the header, pads, gaps, preambles, and postambles, is conditioned according to the parameters stored in the Drive Parameter RAMs to meet the defined recording standard.

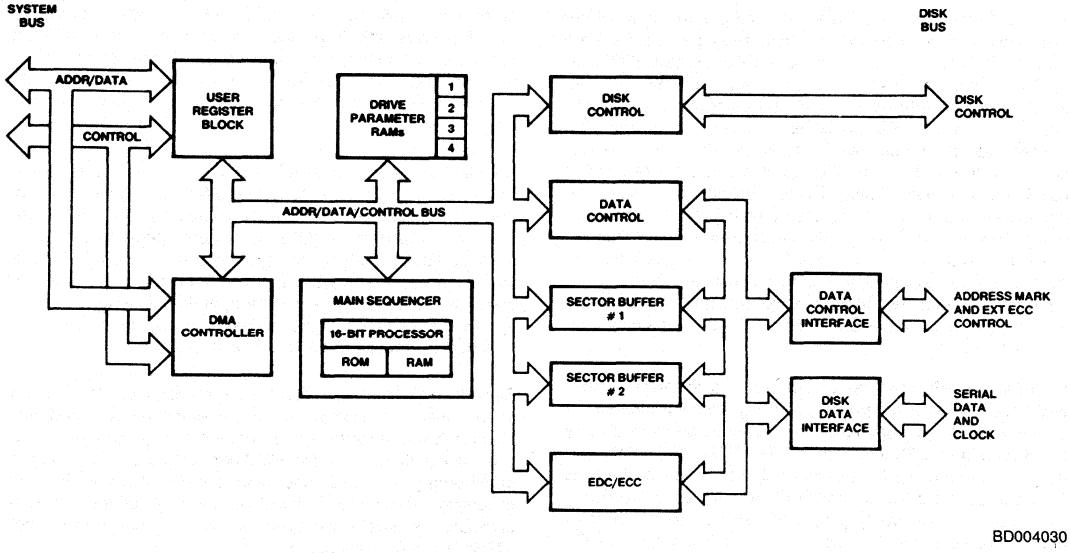


Figure 2. HDC Block Diagram

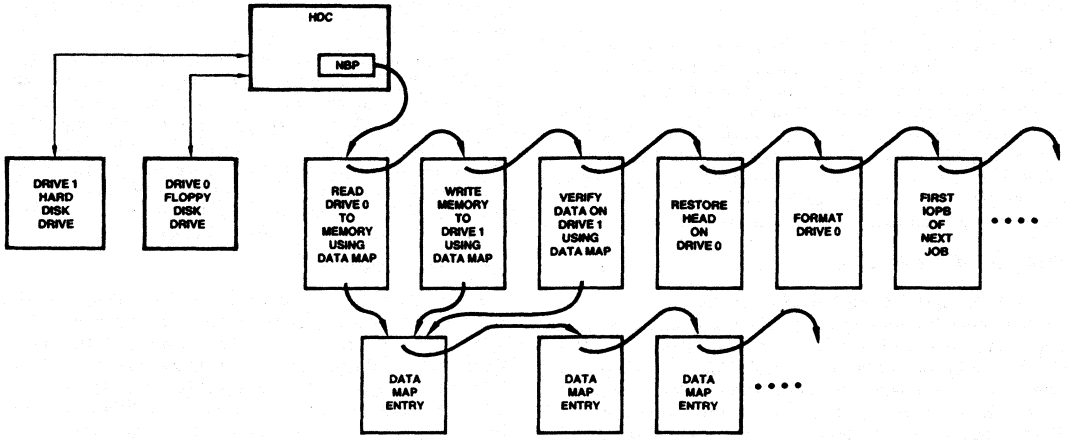


Figure 3. IOPB Linked-List

Software Architecture

IOPB Command Structure

The HDC features a high-level data and command structure. The basic unit of a command structure is the Input/Output Parameter Block (IOPB). The host CPU creates IOPBs in system memory to pass control and status information to the HDC. The HDC fetches these IOPBs using its on-chip DMA controller. Each IOPB specifies one disk command and contains all parameters needed to execute it. To start execution of an IOPB, the host CPU loads the address of the first IOPB into the Next Block Pointer Register and issues the command "Start Chain" by programming the Status/Command Register. After the IOPB is executed, the HDC reports the status information and waits for further instructions. The host CPU can examine the Status/Command Register for information about the command termination. The CPU can also get status from the Status Result Block in memory if an error occurs.

Optionally, IOPBs may be connected in a linked-list format which the HDC can interpret sequentially. With this structure, a complex list of disk commands can be set up and executed by the HDC without CPU intervention. The CPU is then totally free from any processing for disk control. For example, the host CPU can set up a list of commands to copy an entire floppy disk to a hard disk, verify that the data was copied correctly, and, if so, reformat the floppy disk – all without host CPU intervention. This is shown in Figure 3.

First, the HDC reads the data from the floppy disk into a temporary buffer in system memory. The data-mapping option lets the system use any free memory space. The subsequent write command uses the same data map to transfer data to the hard disk. The verify command ensures that data was written correctly by performing a byte-by-byte comparison of the data on the hard disk with the data in the temporary buffer. Optionally, the chain execution may be stopped in the event of an error in the write or verify command. If there was no problem with the data transfer, the HDC will perform the restore and format commands. Optionally, the system can be informed that the disk copy has been completed and that the memory space allocated for the temporary buffer of the disk data is free to be reused.

The seek-lookahead mechanism of the HDC will move forward through the IOPB chain to see where it can overlap head seeks. In this example, the HDC will move the head on the hard disk into position for the write command while the read command is still executing. When the HDC starts executing the write command, it will look ahead in the chain and start executing the restore command on the floppy drive to move the head back to track zero. If possible, the HDC will do additional head positioning for commands further down the IOPB chain.

An IOPB command chain is basically a queue of jobs waiting for execution by the HDC. This offers a pre-defined and efficient structure for the operating system to handle its disk I/O. The ID field of the IOPB allows the operating system to link a particular disk command to the user process that made the disk request. The jobs can thus be placed in the HDC job queue and then forgotten by the operating system unless an error occurs. All the information required to retrace an error is provided by the HDC Status Result Block.

Since the HDC manages the disk job queue, it can look ahead in the queue to overlap some time-consuming operations. Head movement (seeking) can require a major portion of the disk access time. Since the HDC controls up to four drives, it can perform an IOPB operation on one drive while it is executing seeks for future IOPBs on the other drives. This can eliminate the seek-time overhead when those subsequent IOPBs are finally executed.

Data Mapping

Sector data to be transferred to or from the disk may be stored in noncontiguous system memory using the data mapping option. Definable portions of a disk file can be written to or read from separate areas of memory on a byte-by-byte basis. Word processing systems can employ this feature to save text arranged in a linked-list directly on disk and eliminate the time-consuming task of converting the linked-list into a linear list. The Data Map defines the linked-list data structure. The Data Map option is processed by the HDC while the disk is in operation, so that data maps can be handled without affecting data transfer rate.

Status Result Blocks

When the HDC finds that an IOPB has caused an error, it writes a Status Result Block (SRB). Errors can be caused by invalid command codes, disk read and write errors, and seek or memory time-outs. Since the SRB contains the ID number for the IOPB which caused the error, the operating system can determine which disk I/O job caused the error and report this to the user. Depending upon the type of error and what policy has been selected, the HDC may continue with the IOPB chain automatically or wait for the host processor to tell it whether to start over or continue. The SRBs contain all the specific information required to find the exact location of the error and to make recovery as complete as possible.

Registers

When the IOPB command chain has been set up, the Next Block Pointer Register of the HDC should be set to point to the first IOPB in the chain. Writing a Start Chain command to the Status/Command Register causes the HDC to copy the first IOPB into its internal memory. It starts executing the IOPB after updating the Next Block Pointer to the next command in the IOPB chain. The Status/Command Register also reports HDC error and status codes (such as memory time-outs, IOPB option results, and other information related to the internal operation of the HDC).

The Status Result Pointer points to an area of contiguous memory reserved for Status Result Blocks (SRBs). The length of this memory block is defined by the Status Result Length Register, which specifies the number of SRBs. (Each SRB is 10 bytes.) The error-handling scheme of the operating system can manipulate this as needed to coordinate disk use.

The Mode Register controls the HDC's share of the system bus bandwidth to adapt the HDC to the system performance requirements.

Register Description

Four registers control the operation of the HDC. These registers can be accessed directly by the host CPU. The addresses are shown in Figure 4.

Mode Register

Bit assignments of this 16-bit read/write register are shown in Figure 6. This register may be read or written at any time. A hardware or software reset initializes all bits in this register to zero. DMA Burst Length and DMA Dwell Time control the share of the system bus bandwidth allocated to the HDC. DMA Burst Length defines the maximum length of a DMA burst in bytes, and DMA Dwell Time specifies the minimum time between DMA bursts that the HDC must stay off the system bus.

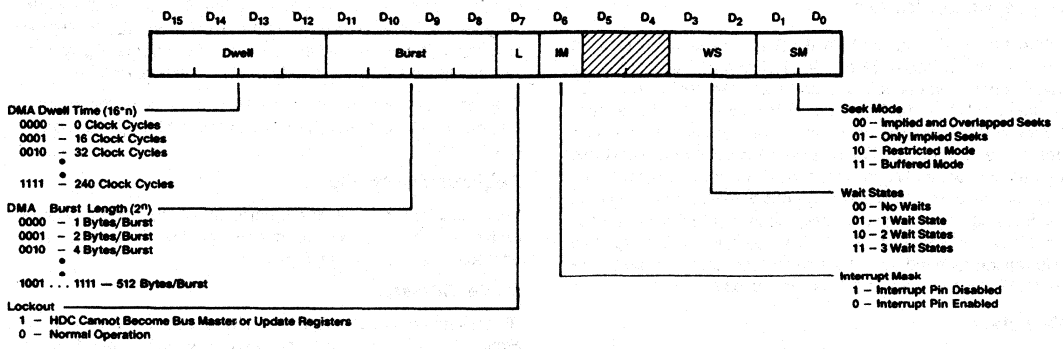
Wait Select (WS) defines the number of software Wait States that are inserted into bus transactions. The HDC inserts this number of Wait States and then waits until the $\overline{\text{READY}}$ line is activated (hardware Wait States). If no acknowledge is received within 2^{16} clock periods, a time-out error is generated.

$\overline{CS}$	A_3	A_2	A_1	Register Accessed
L	L	L	L	Status/Command Register
L	L	L	H	Mode Register
L	L	H	L	Next Block Pointer (low word)
L	L	H	H	Next Block Pointer (high word)
L	H	L	L	Status Result Pointer (low word)
L	H	L	H	Status Result Pointer (high word)
L	H	H	L	Status Result Length
L	H	H	H	Reserved
H	X	X	X	No Register Accessed

Figure 4. Register Addresses

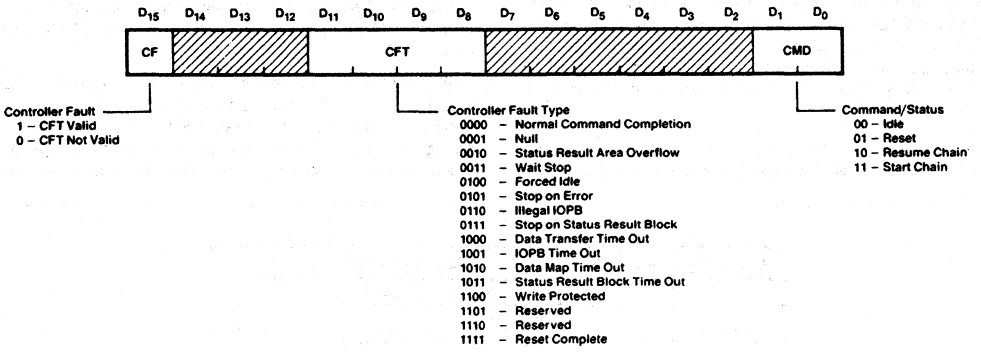
MODE	B/$\overline{W}$	$\overline{BHE}$	A_0	AD<15:8>	AD<7:0>
Word Access	L	L	L	HIGH	LOW
	L	H	L	—	LOW
	L	L	H	HIGH	—
Byte Access	H	X	L	—	LOW
	H	X	H	—	HIGH

Figure 5. Data Bus Assignment for Byte/Word Transfers



DF003790

Figure 6. Mode Register



DF003800

Figure 7. Status/Command Register

The two seek bits control the seek mode. When the Implied Seek option is enabled, the HDC automatically positions the head to the desired track when executing disk commands. This means that explicit seek commands need not precede disk read and write commands. Overlapped seeks are seeks on multiple drives in parallel. In Restricted Mode, seek operations are controlled by external logic. Therefore, STEP, DIRIN, and RTZ should not be connected. In Buffered Mode, seek operations and drive selection are handled externally. Therefore, all disk control lines should not be connected. In Buffered Mode, the HDC controls only the data transfer (RDDAT, WRDAT, RD/REF CLK and WG), the data separator handshake and ECC operation (AMC, AMF, INDEX, FAM/ECC(1:0), and FAMD/ECCERR).

When the Lock-out bit is set, the HDC does not read or modify internal registers. It also keeps the DMA off the system bus. However, dependent on the time this bit is being set, it may execute further DMA bursts to finish the current sector transfer.

The Interrupt Mask enables and disables the Interrupt output pin.

Status/Command Register

The host CPU can access the Status/Command Register any time (Figure 7). The execution of some commands modify the content of this register. A hardware or software reset initializes all bits of the Status/Command Register to zero.

Command

When the command field is written by the host CPU, it causes the HDC to enter the programmed state. By reading this field, the host CPU can determine the state of the HDC.

Idle	The HDC is not performing any action. If CF (Controller Fault) is asserted, then CFT (Controller Fault Type) gives the result of the last known action performed. Any command might be written to this command field while the HDC is idle.
Reset	The software reset is immediately executed. On completion of reset, the HDC goes to the idle state. Any access of a register while in the reset state holds READY inactive until the reset is completed, causing the CPU to wait. A software reset is equivalent to a hardware reset.

Resume Chain	The HDC resumes execution of the IOPB chain where it was interrupted due to a Status Result Area overflow. When receiving the Idle command, the HDC completes the current IOPB and then enters the Idle state.
Start	The IOPB chain pointed to by the Next Block Pointer is executed. When receiving the Idle command, the HDC completes the current IOPB and then enters the Idle state.

Controller Fault

When this bit is set, the Controller Fault Type field is valid. The CF-bit is set by the HDC when entering the Idle state and reset when leaving the Idle state.

Controller Fault Type

The Controller Fault Type indicates the status of the HDC upon entering the Idle state. Whenever the Controller Fault Type is updated, the HDC issues an interrupt.

Normal Complete	— Execution of the IOPB chain terminated without fault.
Null NBP	— A Resume or Start Chain command was given with NBP zero.
SRA Overflow	— The HDC filled the Status Result Area (SRA).
Wait Stop	— An IOPB was completed with the option Wait Stop set.
Forced Idle	— An Idle command was given while executing the Resume or Start Chain command.
Stop on Error	— Nonrecoverable disk error in conjunction with the SE-bit set caused the HDC to terminate chain execution.
Illegal IOPB	— THe HDC attempted to execute an undefined IOPB.
SRB Error	— A Status Result Block (SRB) was written when executing an IOPB with the Stop on SRB (SSRB) option set.
Data Time Out	— Memory Time Out when transferring data.
IOPB Time Out	— Memory Time Out when reading an IOPB.
DM Time Out	— Memory Time Out when reading the Data Map.
SRB Time Out	— Memory Time Out when writing a Status Result Block.
Write Protected	— WRPROT line was asserted when executing a Disk Write command.
Reset	— Hardware or software reset has been executed.

Next Block Pointer

The Next Block Pointer Register is a 32-bit register pointing to the IOPB currently being executed. The HDC updates it on IOPB completion.

Status Result Pointer

The 32-bit Status Result Pointer Register points to the system memory location where the next Status Result Block can be written. This pointer is updated after adding a new Status Result Block to the Status Result Area.

Status Result Length

The 16-bit Status Result Length Register defines the size of the Status Result Area in terms of the number of Status Result Blocks. Therefore, the maximum size of the Status Result Area is 65536 Status Result Blocks.

The Next Block Pointer, the Status Result Pointer, and the Status Result Length should only be updated while the HDC is in the Idle state.

COMMAND DESCRIPTION

All operations of the HDC result from commands. Commands are set up in system memory in IOPBs (I/O Parameter Blocks) (Figures 8 and 9). The HDC starts interpreting the command list after receiving the "Resume Chain" or "Start Chain" command from the host CPU (see Status/Command Register description). Errors and warnings on command execution are reported by adding Status Result Blocks (SRB) to the Status Result Area.

Normal Disk I/O Commands

The HDC supports three normal disk I/O commands: READ, WRITE, and VERIFY. The multi-sector operation is performed on DRIVE starting at the desired TRACK, HEAD and SECTOR. RECORD COUNT defines the number of sectors. The General Select Byte in the Drive Parameter Block specifies whether the track number or the head number is to be incremented on sector overflow. SOURCE/DESTINATION ADDRESS is the starting address of the data block in system memory (DME Low) or the address of the first Data Map Entry (DME High). These commands verify the head position before attempting the data transfer. The Data Mark option allows the DATA MARK parameter to be used instead of F8H. On single-density floppy disks, the Data Mark option also causes the search for a deleted data mark.

READ

READ performs a multi-sector data transfer from disk to system memory.

WRITE

WRITE performs a multi-sector data transfer from system memory to the disk.

VERIFY

VERIFY compares multi-sector data on disk with data stored in system memory. It reports any mismatches or data read errors by updating the Status Result Area.

Initialization Commands

FORMAT

The HDC formats the number of tracks specified by TRACK COUNT starting at HEAD and TRACK. The head and track number are incremented according to the General Select Byte in the Drive Parameter Block. The sectors are numbered like the order given in the sector map, which is sequentially loaded from system memory starting at MAP POINTER. The number of sectors per track is specified in the Drive Parameter Block of this DRIVE. The data field of each sector is filled with PATTERN.

RELOCATE TRACK

A relocation vector is written to all data fields of a bad track. The relocation vector consists of the new track number (ALTERNATE TRACK) and the new head number (ALTERNATE HEAD) (i.e., the track's relocated location). Also, the address mark is changed from FEH to FDH to mark that this track is relocated. Thus, when the HDC encounters three consecutive sectors with an FDH data mark, it assumes the track has been relocated. This command is illegal for single-density floppy disk drives.

LOAD DRIVE PARAMETER BLOCK

This command loads the Drive Parameter Block (DPB) for the selected DRIVE into the internal Drive Parameter RAM (see "Drive Parameters").

DUMP DRIVE PARAMETER BLOCK

The Drive Parameter Block for the selected DRIVE is transferred from the internal Drive Parameter RAM to the DESTINATION ADDRESS.

Marginal Data Recovery Commands

READ PHYSICAL SECTOR

This command lets the user recover a marginal data field which is unrecoverable by normal disk error recovery procedures. First it seeks to the desired track. If TRACK VERIFY is selected, the HDC reads ID fields until three consecutive sector headers show the right track number. This verifies it is on the right track. Beginning with the next index mark, the HDC starts counting the PHYSICAL SECTOR number of IDs specified by "PHYSICAL SECTOR" to locate the desired sector. It reads the data field while disregarding the ID field. The user can thus possibly recover data where headers and/or data marks have been rendered unreadable. PHYSICAL SECTOR specifies the absolute location of the sector. No retries are performed.

READ ID

READ ID attempts to recover the header ID information of a marginal sector. If LOCATOR DUMP is selected, then the first valid ID read is transferred to the DESTINATION ADDRESS. If LOCATOR DUMP is not selected, then the ID of the absolute sector specified by PHYSICAL SECTOR is transferred to DESTINATION ADDRESS. If this command is successfully executed, it updates the HDC's track position.

LOAD BUFFER

The data pointed to by the SOURCE ADDRESS is transferred to the internal sector buffer. The number of bytes transferred is determined by the sector size for the selected DRIVE.

DUMP BUFFER

The data in the internal sector buffer is transferred to system memory starting at DESTINATION ADDRESS. The number of bytes transferred is specified by the sector size for the selected DRIVE.

D_{15}	D_0
NEXT IOPB POINTER <15 : 0>	
NEXT IOPB POINTER <31 : 16>	
ID	
COMMAND CODE(1)	OPTIONS
BYTE 9	BYTE 8
BYTE 11	BYTE 10
BYTE 13	BYTE 12
BYTE 15	BYTE 14
BYTE 17	BYTE 16
BYTE 19	BYTE 18

(1) Not yet defined.

Figure 8. IOPB Structure

Command	Code	Options 7	6	5	4	3	2	1	0	Byte 9	Byte 8	Byte 11	Byte 10	Byte 13	Byte 12	Byte 15	Byte 14	Byte 17	Byte 16	Byte 19	Byte 18
Read	0C	W	SE	SSRB		DME	DM				Drive	Track		Head	Sector	Data Mark	Record Count	Destination <15:0>		Destination <31:16>	
Write	0D	W	SE	SSRB		DME	DM				Drive	Track		Head	Sector	Data Mark	Record Count	Source <15:0>		Source <31:16>	
Verify	0F	W	SE	SSRB		DME	DM				Drive	Track		Head	Sector	Data Mark	Record Count	Source <15:0>		Source <31:16>	
Format	07	W	SE	SSRB						Pat-tern	Drive	Track		Head		Track Count		Map Pointer <15:0>		Map Pointer <31:16>	
Relocate Track	0B	W	SE	SSRB							Drive	Track		Head		Alternate Track		Alter-nate Head			
Load Drive Parameter Block	00	W	SE	SSRB							Drive							Source <15:0>		Source <31:16>	
Dump Drive Parameter Block	03	W	SE	SSRB							Drive							Destination <15:0>		Destination <31:16>	
Read Physical Sector	0A	W	SE	SSRB		TV					Drive	Track		Head	Physi-cal Sector			Destination <15:0>		Destination <31:16>	
Read ID	09	W	SE	SSRB		L					Drive	Track		Head	Physi-cal Sector			Destination <15:0>		Destination <31:16>	
Load Buffer	01	W	SE	SSRB		DME					Drive							Source <15:0>		Source <31:16>	
Dump Buffer	02	W	SE	SSRB		DME					Drive							Destination <15:0>		Destination <31:16>	
Load Syndrome	04	W	SE	SSRB							Drive							Source <15:0>		Source <31:16>	
Dump Syndrome	05	W	SE	SSRB							Drive							Destination <15:0>		Destination <31:16>	
Correct Buffer	06	W	SE	SSRB		LD					Drive							Destination <15:0>		Destination <31:16>	
Seek	0E	W	SE	SSRB		TV					Drive	Track		Head							
Restore	08	W	SE	SSRB		TV					Drive										

Figure 9. IOPB Parameters

LOAD SYNDROME

The Reed-Solomon check bytes are transferred from SOURCE ADDRESS to the internal Syndrome RAM. This command generates an error if the Drive Parameter Block of the selected DRIVE specifies CRC or external ECC error-checking.

DUMP SYNDROME

The Reed-Solomon check bytes are dumped to the DESTINATION ADDRESS.

CORRECT BUFFER

This command uses the contents of the Syndrome RAM to correct the data in the internal sector buffer. It generates an error if the Drive Parameter Block of the selected DRIVE specifies CRC or external ECC error-checking. If the LOCATOR DUMP option is selected, the HDC additionally writes the location and values of errors sequentially to DESTINATION ADDRESS. If the buffer address is greater than the sector size and the error pattern is non-zero, then the error is uncorrectable for that locator. If the error pattern and the buffer address both are zero, then no error was detected by that locator. If the error pattern is zero and the buffer address is greater than the sector size, then the error occurred in the check bytes.

Head Movement Commands

SEEK

The HEAD of the selected DRIVE is moved to the desired TRACK. If TRACK VERIFY is selected, the HDC looks at the first encountered header to determine whether it is on the right track. If the track numbers mismatch, the HDC reports an error. Usually, this command is issued only if implied and overlapped seeks are disabled.

RESTORE

RESTORE moves the heads of the selected DRIVE to track 0. This command synchronizes the HDC and the drives or recovers seek errors. The HDC supports two restore options. It can restore drives by issuing step-out pulses until the drive reaches track 0 and asserts TRK0. Optionally, drives with built-in restore logic may be restored by a pulse on the RTZ (Return To Zero) line. If TRACK VERIFY is selected, the HDC will also scan the header IDs to verify that the restore was successful.

Option Byte

One byte in each IOPB contains a set of options applicable to the particular command.

W — WAIT
- 0 — After execution of current IOPB, continue with next IOPB.
- 1 — Stop IOPB execution after terminating current IOPB.

SE — STOP ON ERROR
- 0 — Option disabled.
- 1 — Stop IOPB chain execution if current IOPB causes a Status Result block to be written.

SSRB — STOP ON STATUS RESULT BLOCK
- 0 — Option disabled.
- 1 — Stop IOPB chain execution if current IOPB causes a Status Result block to be written.

DME — DATA MAP ENABLE
- 0 — Data Mapping disabled.
- 1 — Data Mapping enabled. Source/Destination Address links to first Data Map Entry.

DM — DATA MARK
- 0 — Data Mark disabled.
- 1 — Data Mark enabled (see Normal Disk I/O Commands).

TV — TRACK VERIFY
- 0—Track Verify disabled.
- 1—Track Verify enabled (see Read Physical Sector and Seek).

L — LOCATOR DUMP
- 0—Locator Dump disabled.
- 1—Locator Dump enabled (see Read ID).

DATA MAPPING OPTIONS

The Data Mapping option lets the HDC process data stored in noncontiguous system memory. This option is available to five commands: Read, Write, Verify, Load Buffer, and Dump Buffer. It is enabled by setting the Data Map Enable bit (DME-bit) in the IOPB.

The last two words of the IOPB (Source/Destination Address) link to the first Data Map Entry (Figures 10 and 11). If this pointer is zero, then the Data Map does not exist and the HDC does not transfer data. Data Map Entries are linked together via the Data Map Pointer. The Data Map linked-list is terminated if the Data Map Pointer is set to zero. Each Data Map Entry defines a data buffer in system memory starting at the address defined by Data Source/Destination Address. The size of this buffer is defined by Byte Count. When Byte Count is set to zero, the HDC assumes a size of 2^{15} bytes. When the Load Enable (LE-bit) is reset to zero, the HDC masks off a data block with the size specified by Byte Count (Figure 12).

STATUS RESULT BLOCK (SRB)

The host CPU reserves a Status Result Area defined by the 32-bit Status Result Pointer register and the 16-bit Status Result Length Register. Whenever the HDC terminates a command and an error occurred, it adds a Status Result Block (Figures 13 and 14) to the Status Result Area. A Status Result Block carries the same unique ID number of the IOPB where the error occurred.

Multi-record Command Terminated

This SRB indicates that a Multi-record command has been terminated due to an error condition at the specified TRACK, HEAD, SECTOR, and RECORD COUNT.

No IDs Found On Track

A given Track was searched for an ID and none was found.

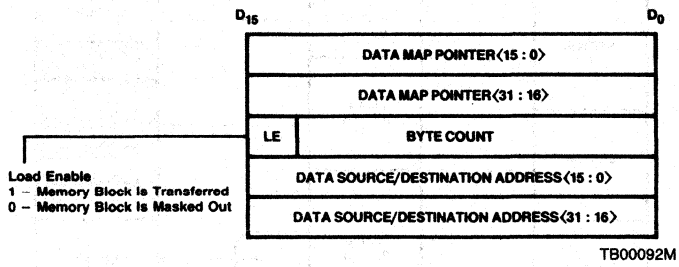


Figure 10. Data Map Entry

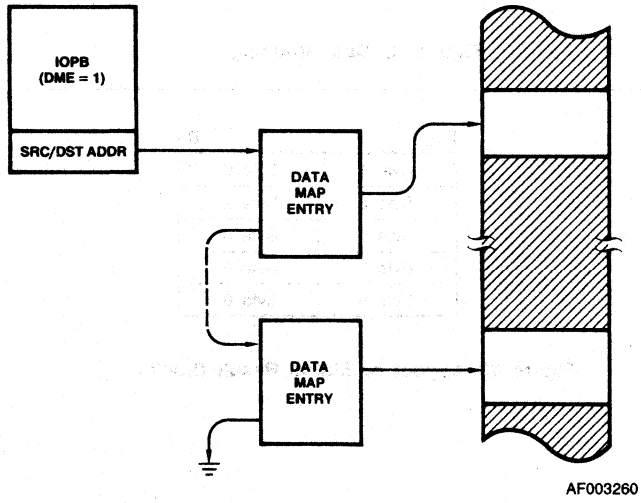


Figure 11. Data Map Entry Linked List

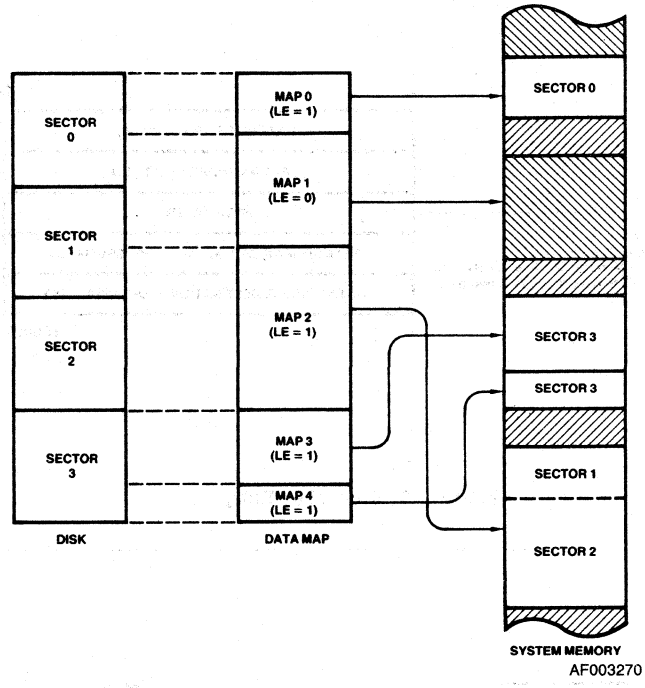


Figure 12. Data Mapping

D_{15}	D_0
Byte 1	Byte 0
Byte 3	Byte 2
Byte 5	Byte 4
Byte 7	Byte 6
Byte 9	Byte 8

Figure 13. Layout of Status Result Blocks

Status Result Block	Byte 1	Byte 0	Byte 3	Byte 2	Byte 5	Byte 4	Byte 7	Byte 6	Byte 9	Byte 8
Multirecord Command Terminated	ID		Record Count	Code(1)	Track		Head	Sector		
No IDs Found on Track	ID			01	Track		Head			
Format Error	ID			00	Track		Head			
Seek Error	ID			02	Current Track		Desired Track		Current Head	Desired Head
Fatal Seek Error	ID			08	Current Track		Desired Track		Current Head	Desired Head
Relocated Track	ID			05	Current Track		New Track		Current Head	New Head
Relocated Track (No Vector)	ID			06	Track		Head			
Record Not Found	ID			09	Track		Head	Sector		
Record Not Found (ID Errors)	ID			07	Track		Head	Sector		
Multirecord Overflow	ID			0D	Track		Head	Sector		
Data Not Recovered	ID			17	Track		Head	Sector		
Data Recovered with Retries	ID		Retry Count	0A	Track		Head	Sector		
Data Recovered with ECC	ID		Retry Count	03	Track		Head	Sector		
Data SYNC Fault	ID			04	Track		Head	Sector		
Data Mark Error	ID		Data Mark	0E	Track		Head	Sector		
Sector Size Mismatch	ID		Sector Size	0F						
Data Non-Verify	ID			0B	Track		Head	Sector		
Physical Data Recovered with ECC	ID			Code(1)						
ECC Error Not Corrected	ID			Code(1)						
ECC Not Selected	ID			10						
ECC Error in Data Field	ID			18						
Data Mark Physical Error	ID		Data Mark	Code(1)						
ID CRC Error	ID			Code(1)						
Fault While Seeking	ID			12	Track					
Restore Fault	ID		Drive Status	16	Track					
Fault While Head Select	ID			13			Head			
Drive Selection Fault	ID		Drive Status	11						
Drive Status Trap	ID		Drive Status	Code(1)						
Data Timeout	ID			0C	Byte Count		Block Address <15 : 0>		Block Address <31 : 16>	
No Single Density Floppy Relocation	ID			14						
End of Data Map	ID			15						

(1) Note yet defined.

Figure 14. Status Result Blocks

Format Error

When reading or writing, it indicates that an Index Pulse occurred when not expected. When formatting, it indicates that the Index Pulse occurred before the format was completed. This means the number of sectors times the sector size is too large for the track.

Seek Error

The HDC has read a valid ID showing that the current TRACK/HEAD differs from the expected TRACK/HEAD.

Fatal Seek Error

After a seek error, the HDC has performed a restore and a second seek, but the seek error still persists. The IOPB is aborted.

Relocated Track

A track was read as relocated and the HDC auto-vectored to the new track.

Relocated Track (No Vector)

A track was read as relocated, but the HDC could not find a valid data field to read the vector. The IOPB is aborted.

Record Not Found

No ID was found corresponding to the desired sector number.

Record Not Found (ID Errors)

Sector was not found, but one or more IDs had CRC errors.

Multi-record Overflow

The HDC aborted a Multi-record command because the last sector of a track was read and the programmed Multi-Record Policy prohibits a track/head increment.

Data Not Recovered

The HDC could not recover a sector requested by a read command.

Data Recovered with Retries

Data was recovered using read retries (RETRY COUNT).

Data Recovered with ECC

The Reed-Solomon successfully corrected a read error. Retry Count specifies the number of retries performed before the error was corrected.

Data Sync Fault

The HDC successfully read the ID for the desired sector, but could not find an Address Mark for the data field.

Data Mark Error

The Data Mark read did not correspond to the Data Mark specified by the Drive Parameter Block.

Sector Size Mismatch

Sector Size read from the Header field does not match the Sector Size defined in the Drive Parameter Block (floppy formats only).

Data Non-Verify

The Verify command detected a mismatch between disk and memory data.

Physical Data Recovered with ECC

The Read Physical Sector command had to use ECC to correct the data field.

ECC Error Not Corrected

The data transferred to system memory by Read Physical Sector contained uncorrected errors.

ECC Not Selected

Error correction attempted when ECC was not selected internally.

ECC Error in Data Field

An error was detected in the data field of a sector.

Data Mark Physical Error

The Data Mark read by Read Physical Sector did not correspond to programmed Data Mark.

ID CRC Error

A CRC error was detected when reading the ID field by the Read ID command.

Fault While Seeking

While seeking, the FAULT line was asserted or the SEEKCOM line was not asserted.

Restore Fault

The HDC could not restore the drive.

Fault While Head Select

The HDC could not select the specified head. The FAULT line was asserted or SEEKCOM was not asserted (for floppy disk drives only).

Drive Selection Fault

A fault occurred when selecting a drive. The Drive Status byte latches the levels of the status lines when the fault occurred:

D_9 = DREADY
D_{10} = FAULT
D_{11} = SEEKCOM
D_{12} = WRPROT
D_{13} = TRK0

Drive Status Trap

An unexpected change in state of one or more drive status lines occurred while the drive was selected. The Drive Status byte latches the status lines. The IOPB is aborted.

Data Time-out

A memory time-out occurred while accessing the system memory. The Block Address defines the starting location of a block of system memory where the time-out occurred. Byte Count defines the length of the block.

No Single-Density Floppy Relocation

A Relocate Track command was attempted on a single-density floppy disk drive.

End Of Data Map

Unexpected end of Data Map encountered.

DISK DATA I/O

Sector Formats

Data is stored on the disk in sectors. Each sector consists of two fundamental parts: the header and the data field. The sizes of all pads, gaps, preambles, postambles, and data fields are programmable in the Drive Parameter Block. Floppy disk formats also have an Index field at the beginning of each track.

Header

The header contains the Address Mark, the track number, the head number, and the sector number (Figure 16). Two trailing CRC check bytes protect the header. The beginning of the header is marked by an ID Address Mark (IDAM). The single-density floppy format requires a single-byte address mark while the double-density floppy and hard disk formats use a two-byte address mark. The first byte (the only byte for single-density floppy formats) of the address mark is a unique clock/data pattern written and detected by the data separator. Since the unique clock/data pattern is written by the disk data separator, its length and layout are transparent to the HDC. The second byte, which is processed by the HDC, specifies that this is a normal (FEH) or a relocated (FDH) track. This byte is not present for single-density floppy formats. A preamble and a postamble compensate for speed and mechanical variations of the disk system. The length of both fields is user-programmable.

Data Field

A Data Mark similar to the header's ID Address Mark marks the beginning of the data field (Figure 17). The single-density floppy format uses a single-byte Data Mark which is different from the Address Mark. The data field can be marked by two types of data marks to specify either a normal data field or a deleted data field. The double-density floppy and the hard disk

formats have a two-byte Data Mark consisting of a unique clock/data pattern similar to the Address Mark and a user-programmable Data Mark byte.

The data field has a user-programmable length of 128, 256, or 512 bytes. It is protected by one of three user-selectable data protection algorithms: CRC-CCITT (error detection), Single-Burst or Double-Burst Reed-Solomon (error detection and correction). Optionally, external error processing logic can be interfaced for hard disk format to implement any user-designed algorithm. The check bytes are stored in the ECC field. For external error processing, the user-designed external error processing logic reads and writes the ECC field. The number of ECC bytes appended varies from 2 to 15 bytes for the internal ECC options and 1 to 256 bytes (determined by the drive parameter block) for external ECC.

Data Separator Interface Signals

The data lines (RDDAT and WRDAT) transfer the data to and from the disk. The transfer is controlled by five signals: INDEX, AMC, AMF, RG, and WG. The FAM/ECC<1:0> and FAMD/ECCERR lines interface with external hardware to allow user-definable error-detection/correction. For floppy formats, these lines interface with the data separator to allow address/data mark detection and generation.

Header Search Mechanism

The unique clock/data patterns of the address/data marks are written and detected by the data separator only. When the HDC attempts to read a header or to access a data field, it activates Address Mark Control (AMC).

For floppy formats, the FAM/ECC<1:0> outputs differentiate between the various index, address, and data marks defined by the IBM Single- and Double-density Format. Whenever the HDC activates the AMC line to read or write a special mark from a floppy disk, the data separator should use the FAM/ECC<1:0> to determine the unique clock/data pattern to read or write. Additionally, for single-density floppy formats, the FAMD/ECCERR pin indicates that a deleted data address mark should be written or read.

Next, the HDC activates Read Gate (RG) and waits for the data separator acknowledge. The data separator should assert Address Mark Found (AMF) when it detects an address/data mark. The rising edge of AMF indicates that valid data will be on the RDDAT pin on the next rising edge of the RD/REF CLK (see Figure 23). For double-density floppy and hard disk formats, the HDC checks the next eight bits for an FDH or FEH. If the check fails, a data address mark was found; the HDC deactivates AMC and RG. After AMF becomes inactive, AMC is reasserted then followed by RG, and the whole procedure is repeated until a header is found or it is determined that no header can be found (indicated by three index pulses). For single-density floppy formats, the rising edge of AMF indicates that a header address mark has been found.

After finding a valid header address mark, the HDC compares the next 48 bits (6 bytes) of serial data with the track, head, and sector number of the desired sector (and sector size for floppy formats) to determine if the right header has been found. A CRC check ensures the correctness of the header information. If the track or the head numbers recovered from the header do not match the desired track or head number, an error is flagged and the command is aborted. For floppy formats, sector size is also checked against the sector size defined for the current drive. The sector number is checked to see if this is the desired sector.

If the correct sector number was not found, then the process is started all over again until either the desired sector is found or the HDC determines that the sector cannot be found on the current track. If three index pulses are detected (indicating two or more disk revolutions), the Sector Not Found error is flagged.

If the correct sector number was found, the search for the data mark begins.

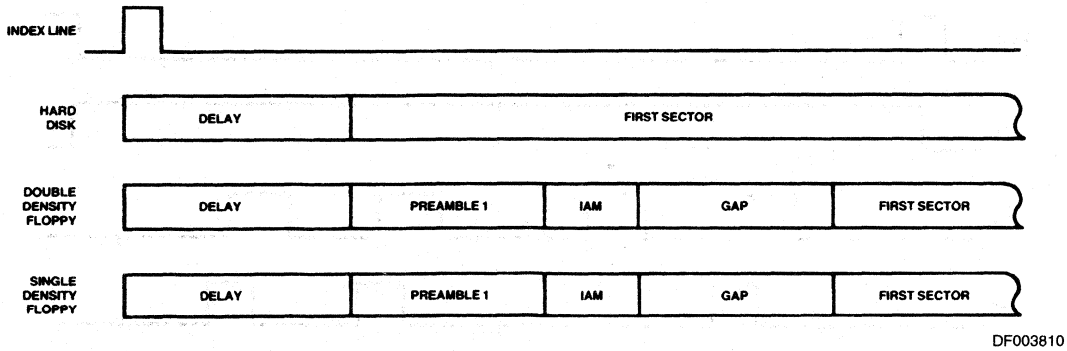


Figure 15. Track Initialization Field

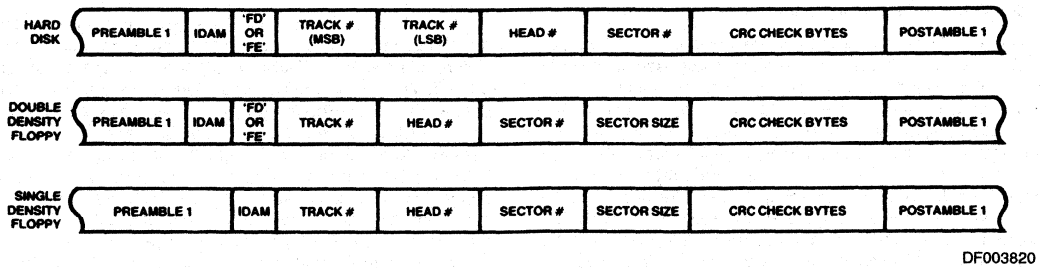


Figure 16. Sector Header Formats

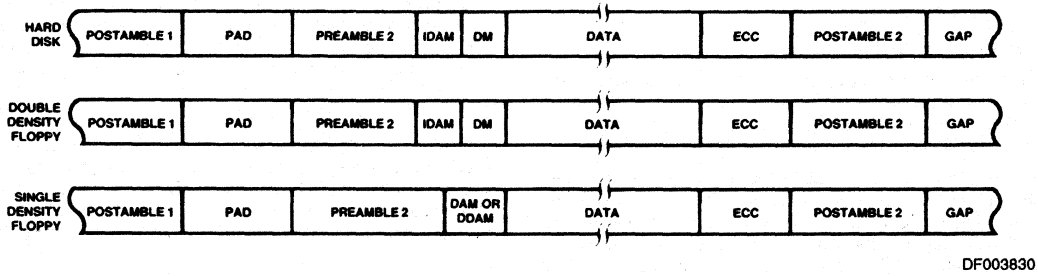


Figure 17. Sector Data Fields Format

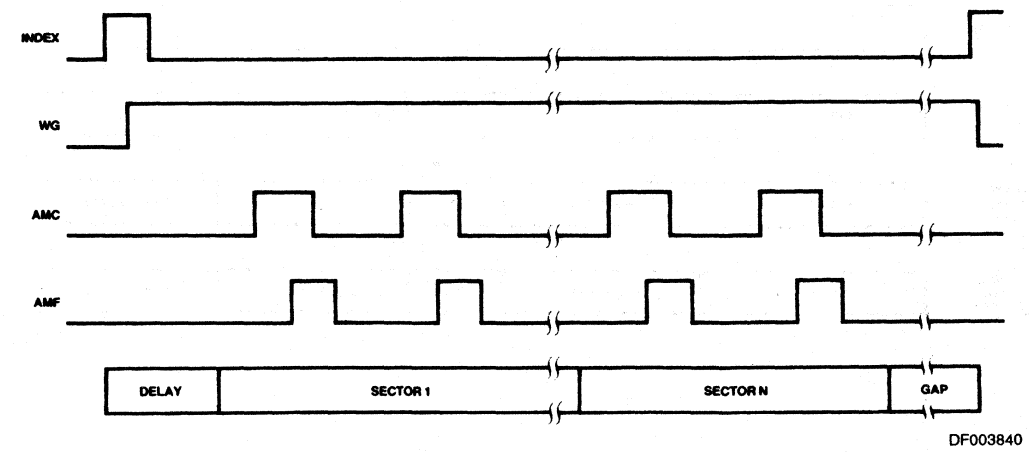


Figure 18. Write Track Sequence

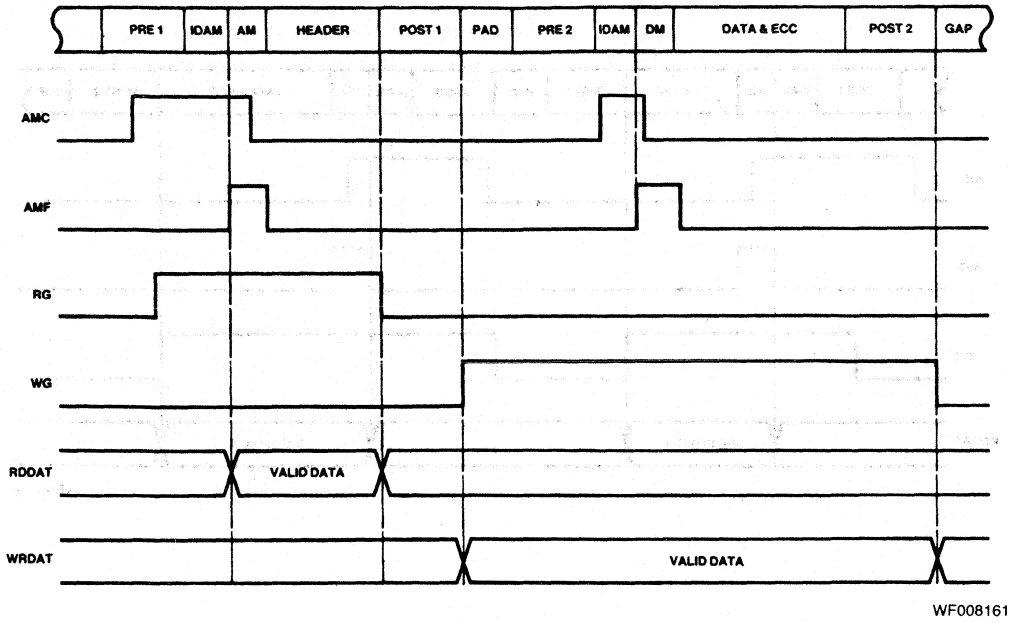


Figure 19. Write Sector Control Sequence (For Hard Disk and Double-Density Floppies)

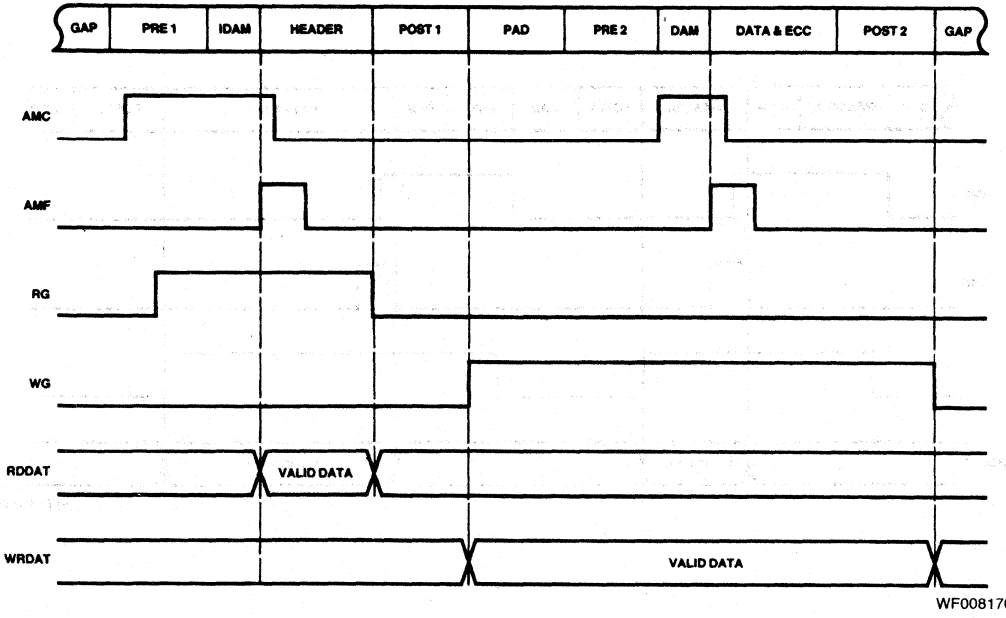


Figure 20. Write Sector Control Sequence (For Single-Density Floppies)

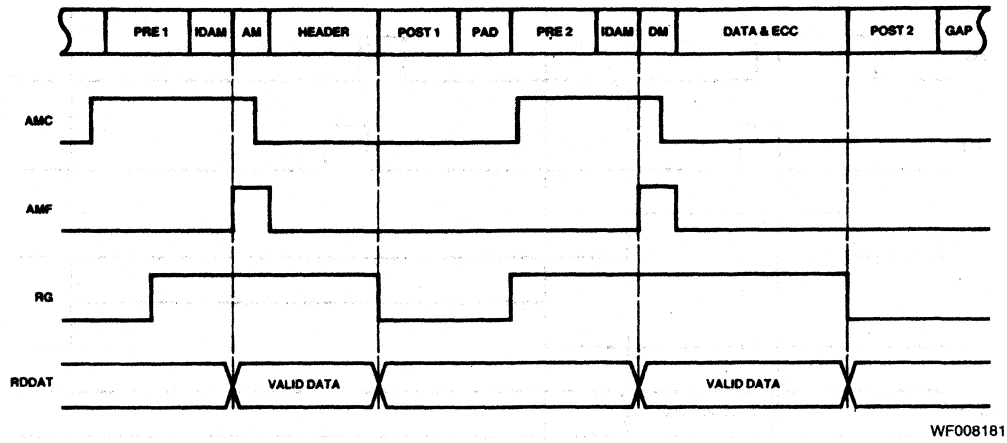


Figure 21. Read Sector Control Sequence (For Hard Disks and Double-Density Floppies)

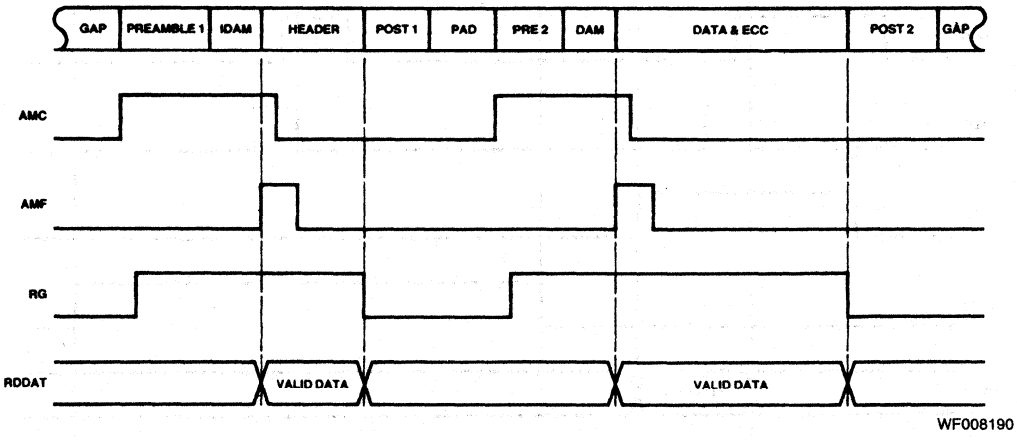


Figure 22. Read Sector Control Sequence (For Single-Density Floppies)

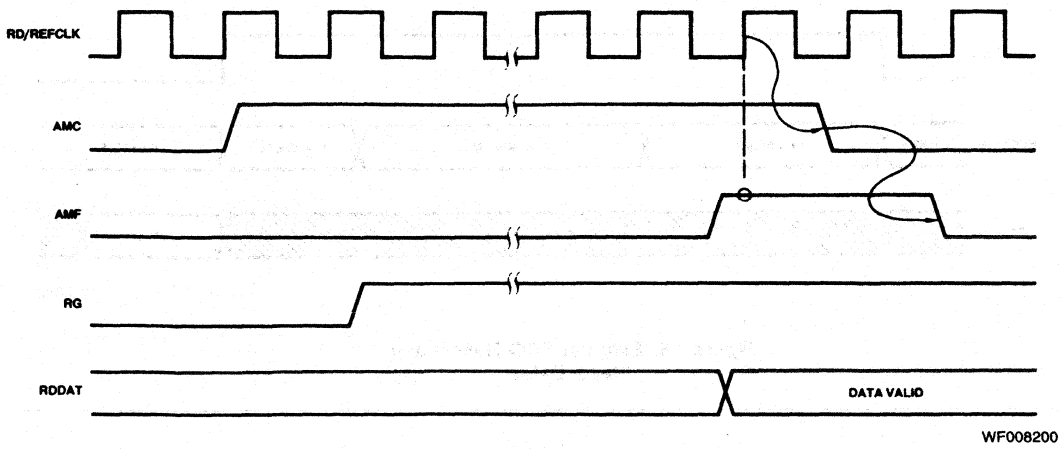


Figure 23. Address Mark Control/Address Mark Found Handshake (Read Data)

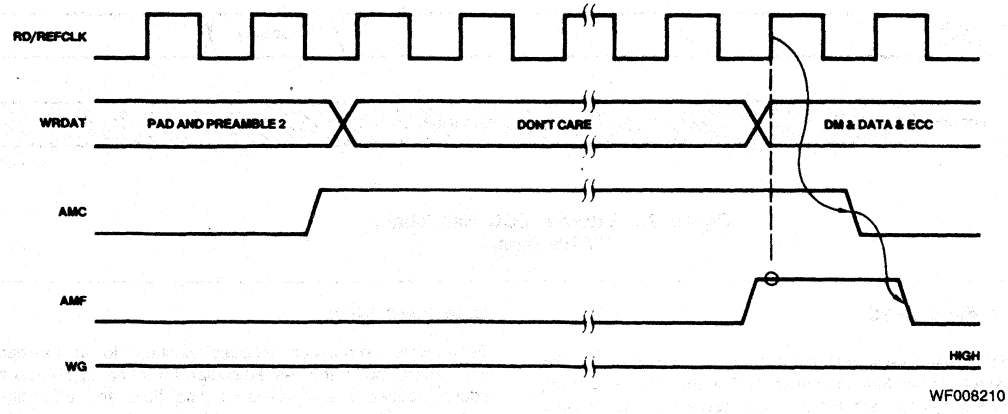


Figure 24. Address Mark Control/Address Mark Found Handshake (Write Data)

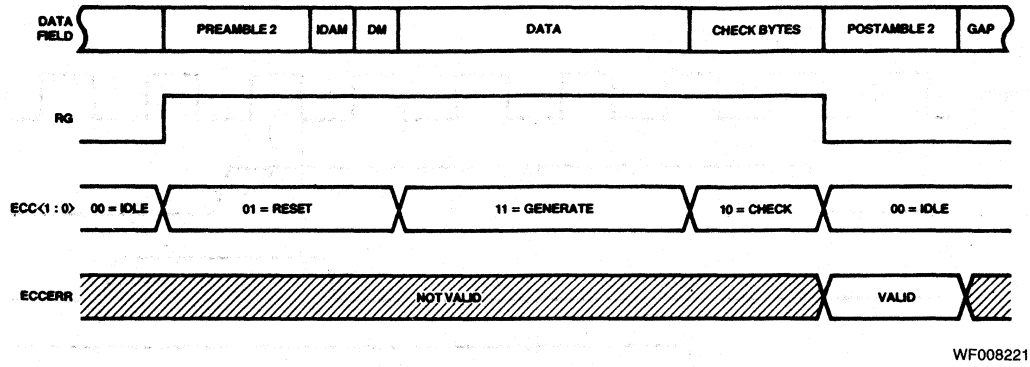


Figure 25. External ECC Handshake (Read Data)

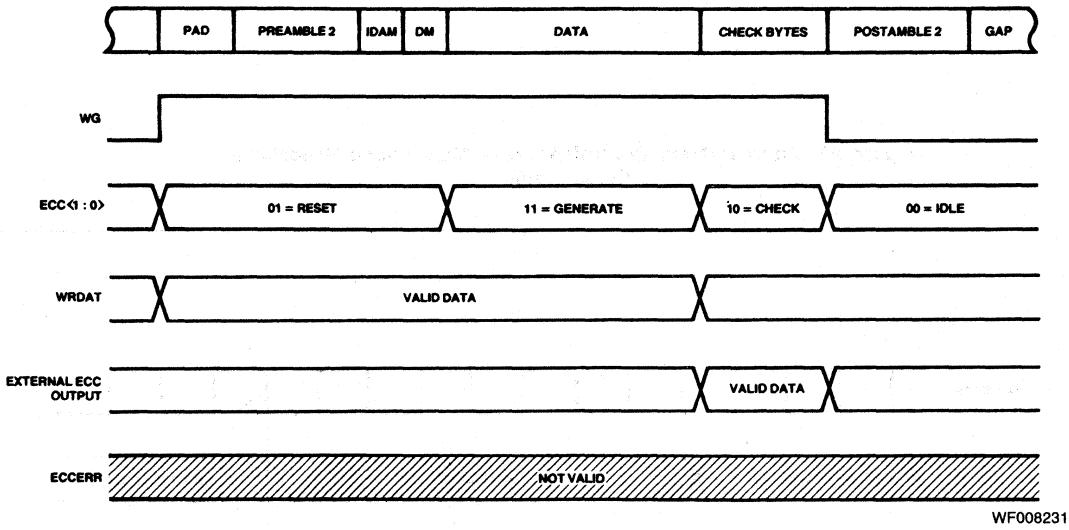


Figure 26. External ECC Handshake (Write Data)

Data Field Read

The HDC uses the same handshake procedure outlined above to search for the data address mark. For double-density floppy and hard disk formats, the HDC compares the byte following the data address mark to the programmed data mark to confirm that the data field has been found. The HDC reads the data field into the sector buffer and the ECC bytes into the error logic. For hard disk formats, the HDC ignores the ECC field if external ECC is selected. The external error processing logic should read the ECC field to verify the integrity of the data field. FAM/ECC<1:0> and FAMD/ECCERR control the external error logic handshake. Immediately after the ECC field has been read, RG is turned off and the sector-read operation is terminated.

Data Field Write

Sector-write commands proceed similarly to sector-reads while searching for the desired sector. After finding the desired sector header, the subsequent data field and ECC field, including pad, preamble2, data address mark and postamble2 are overwritten. The HDC activates Write Gate (WG) at the beginning of the pad. Write Data on WRDAT line is valid with the next rising edge of the clock. First, the HDC writes the pad and preamble2 fields. On completion of preamble2, AMC is activated. For floppy formats, FAM/ECC<1:0> indicates to the data separator the data address mark to write. The data separator should assert AMF after writing the last data address mark bit. The HDC resumes data output with the next clock. For double-density floppy and hard disk formats, the HDC writes a user-programmable data mark first. Next the

data field is output on the WRDAT line. Depending on the selected ECC mode, either the HDC or the external error logic appends the check bytes to the data field. FAM/ECC<1:0> and FAMD/ECCERR control the external error logic handshake. WG is turned off after the postamble2 field is written. This completes the write-sector sequence.

Format Track

Format Track always writes the entire track. It is the only command that writes the sector header. Beginning with the rising edge of the INDEX pulse, the HDC asserts WG and outputs the pattern for the delay field (Figure 18). Note that disk drives require that the delay field be wider than the INDEX pulse width. For floppy formats, an Index Address Mark (IAM) field follows the delay field. Then the HDC starts writing sectors.

For writing the headers, the HDC uses the track, head and sector size information supplied by the format IOPB. A sector map in the system memory supplies the logical sector number sequence. The first byte of the sector map is written in the sector number field of the first physical sector. The second byte is written in the second physical sector. This process continues until the required number of sectors have been formatted. For multiple track format commands, the sector maps for the tracks to be formatted are organized in a linear list in the system memory. The Map Pointer of format IOPB points to the beginning of this sector map list.

For writing address/data marks, the HDC proceeds as described in the section "Data Field Write." The data field is filled with the user-supplied pattern byte in the format IOPB. The gap between the end of the last sector and the rising edge of the index pulse is filled by the gap pattern. The length of this gap field depends on the track capacity and the number of sectors/track. It is different from the intersector gap which has a user-definable length. The patterns written for all fields are shown in the table below.

Field	Single-density floppy formats	Double-density floppy and hard disk formats
Delay	FFH	4EH
Preamble1	00H	00H
Postamble1	FFH	4EH
Pad	00H	00H
Preamble2	00H	00H
Postamble2	FFH	4EH
Gap	FFH	4EH

DRIVE PARAMETER PROGRAMMING
(Figure 27)

The HDC contains one set of Drive Parameter Registers for each drive. The system can only access this register set indirectly by either of the following commands: Load Drive Parameter Block and Dump Drive Parameter Block. The block is set up in contiguous system memory. Byte values in this table, when set to zero, represent the value 256.

General Select Byte (Figure 28)

The EDCP field determines which of the error-checking algorithms the HDC should use to recover marginal data. The Auto Vector Enable bit (AVE) allows the HDC to vector automatically to the new track if a given track is found to be relocated. This bit is ignored for single-density floppy formats. The Multi-Record Policy (MRP) field determines the policy followed for multiple sector transfers. Multi-record commands may require that the HDC select a new track to access subsequent sectors. One option is to increment the head number. If the head number overflows, the HDC resets it to zero and increments the track number. If the track number overflows, it is reset to zero. The second option allows the track number to be incremented first. On overflow, the head number is incremented. The RTZEN bit defines the recalibration mode (move head to track 0). The HDC can either assert the RTZ pin (16 clock cycles active) or issue STEP pulses until TRK0 becomes active. The maximum number of STEP pulses for recalibration is 65535.

Data Select Byte (Figure 29)

The DSZE field defines the size of the data field for a particular drive.

Retry Policy Byte (Figure 30)

The most significant three bits define the error-checking policy. These bits are ignored if CRC or external ECC is used. If ECC Before Retry is selected, the HDC uses ECC even when retry attempts are disabled. The RE-bit enables the number of retry attempts defined by Retry Count (RC).

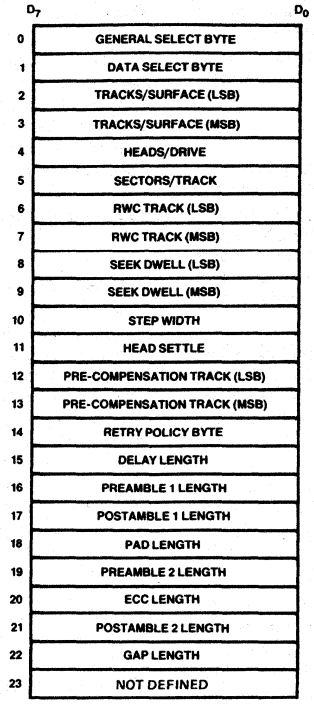


Figure 27. Drive Parameter Block

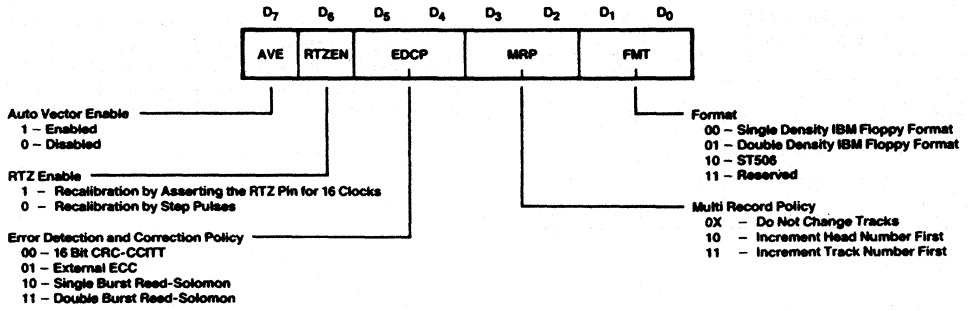


Figure 28. General Select Byte

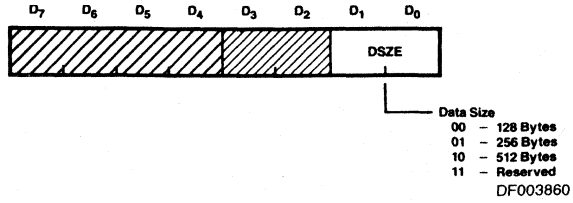


Figure 29. Data Select Byte

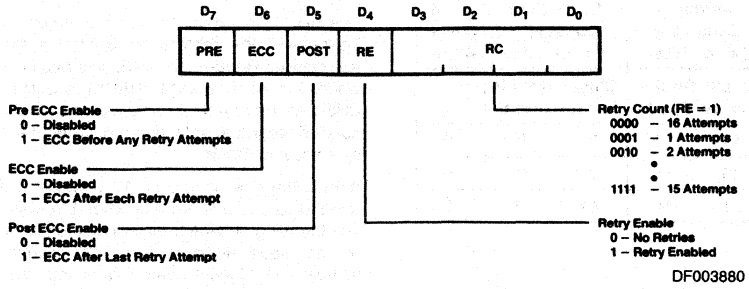


Figure 30. Retry Policy Byte

Drive Parameters

Tracks/Surface specifies the number of cylinders (tracks) in the range from 0 to 65536. Head/Drive specifies the number of moving heads (1 to 256). However, only the four least significant bits are output at the head select pins. Sectors/Track defines the number of sectors per track (1 to 256). RWC Track specifies the track number where the Reduced Write Current pin should be activated (only for ST506 and floppy formats). If the current track number is greater than RWC Track, the pin (RWC) is activated. Pre-Compensation Track specifies the track where the pre-compensation starts.

Seek Timing

Step Width (8-bit) determines the width of step pulses. Seek Dwell (16-bit) sets the delay between the falling edge and the rising edge of step pulses. If either Step Width or Seek Dwell is set to 1, then the step cycle is equal to 18 system clocks (50% duty cycle). If neither of these values is 1, then the Step Width is equal to 13 x (Step Width + 8) and the dwell time is equal to 17 + (Seek Dwell x 8). Head Settle defines the time to allow the heads of the selected drive to settle, after the head select lines change, in increments of 4 clock cycles.

Sector Format

The sector format parameters specify the sizes of particular sector header and data fields.

DISK DATA PROTECTION

All data stored on disk is protected by error-checking algorithms. The HDC supports four modes:

- 16-bit CRC-CCITT (error detection)
- Single-Burst Reed-Solomon (single-burst correction)
- Double-Burst Reed-Solomon (double-burst correction)
- External ECC

The error-checking scheme for each drive is defined by the EDCP field in the General Select Byte of the Drive Parameter Block. It is possible to use a different ECC for each sector or track provided that the Drive Parameter Block is constantly changed. However, this is not a recommended procedure.

CRC-CCITT

The CRC-CCITT code is a cyclic-based, error-detecting/non-correcting code. It is the industry standard error checking code for magnetic disk systems. CRC-CCITT is mandatory for the protection of the sector ID field, but the data field can be protected by any of the four modes mentioned above. The CRC generator polynomial is:

$$X^{16} + X^{12} + X^5 + 1$$

The guaranteed capabilities of the code are listed below:

- Detects all odd number bit errors
- Detects all single-burst errors of 16 bits or less
- Detects all single, double, and triple bit errors

Single-Burst Reed-Solomon

The HDC supports two error correction codes, Single-Burst Reed-Solomon and Double-Burst Reed-Solomon. Single-Burst Reed-Solomon corrects single-burst errors and detects double-burst and some triple-burst errors. A single burst of errors is defined as any number of bit errors (contiguous or noncontiguous) where the distance between the first and the last bit error does not exceed the burst length given in the table below. The ECC code protects the check bytes as well as the data. The guaranteed performance of the Single-Burst Reed-Solomon code is shown below.

Sector Size (# of bytes)	Detection Capability (# of bits)		Correction Capability (# of bits)		# of Check Bytes
	Single Bursts	Double Bursts	Single Bursts	Double Bursts	
128	33	9	9	0	6
256	33	9	9	0	6
512	57	17	18	0	9

This table shows, for instance, that in a sector of 256 bytes any single burst of errors with a length of up to 33 bits will be detected. Note, that two single bit errors separated by more than 32 bits will count as a double-burst error. Alternatively, any two random single-bursts (double-burst) of up to 9 bits each will be detected also. In this example, this code can correct a single burst of errors up to 9 bits and cannot correct double-burst errors. The table presents the guaranteed capabilities of this code. Under certain circumstances the code is capable of detecting longer bursts or even triple-burst errors.

Double-Burst Reed-Solomon

Double-Burst Reed-Solomon is an enhanced version of Single-Burst Reed-Solomon. This code can detect and correct single- and double-burst errors of the maximum size listed in the following table.

Sector Size (# of bytes)	Detection Capability (# of bits)		Correction Capability (# of bits)		# of Check Bytes
	Single Bursts	Double Bursts	Single Bursts	Double Bursts	
128	49	16	25	9	10
256	49	16	25	9	10
512	81	24	41	17	15

External ECC

The HDC features external ECC to allow schemes other than CRC-CCITT or Reed-Solomon. External ECC can only be used for the protection of data fields. Three lines (ECC < 1:0 > and ECCERR) simplify interfacing the external ECC hardware. ECC < 1:0 > present the status of the HDC to allow the external ECC to run synchronously. The status is coded in Gray code; only one bit changes when going from one state to the next (Figure 26).

In the IDLE state, no data field of a sector is written or read. The external ECC should be inactive.

RESET should reset the external ECC to prepare itself for an ECC process. At the end of the data mark, while reading or writing the last bit, the status lines change into the GENERATE state.

On the next rising edge of the RD/REF CLK, the external ECC must be prepared to receive valid data on either the RDDAT or WTDAT lines (depending on whether RG or WG is asserted). The external ECC must generate the check bytes. When reading or writing the last bit of the data field, the lines change to the CHECK state.

CHECK enables the external ECC to either multiplex the check bytes on WRDAT (WG active) or to compare the generated check bytes with the bytes read from RDDAT (RG active). With the external ECC, a programmable number of check bytes can be added to the data field of a sector (1 to 256 bytes).

On completion of the check byte field, at the last bit of the last check byte, the status lines change back to the IDLE state. On the next rising edge of the RD/REF CLK, the IDLE state will be in effect. During the last byte of the Postamble2 field following the check bytes, the ECCERR pin will be sampled by the HDC for an "Error Found" signal from the external ECC. If the ECCERR pin is not asserted (LOW), then the HDC assumes that the data is valid. If the ECCERR line is asserted (HIGH), then the HDC assumes that an error occurred in the data field.

INTERFACING

System Interface

For both Slave Mode and Master Mode, the system bus interface can be programmed for byte (B/$\overline{W}$ High) or word (B/$\overline{W}$ Low) transfers as shown in Figure 5.

Slave Mode

In Slave Mode, the host CPU can access the five internal registers of the HDC. A_0 to A_3 indicate the address of the internal register (Figure 4). In Byte Mode, the High byte is accessed if A_0 is HIGH and the Low byte is accessed if A_0 is LOW. In Byte Mode, $\overline{BHE}$ is ignored. The HDC asserts $\overline{READY}$ to indicate that it is ready to complete the access.

Master Mode

The HDC is in Master Mode when it controls the system bus. To request the mastership on the system bus, the HDC asserts Bus Request (BREQ). The bus is granted to the HDC when Bus Acknowledge (BACK) is active. The HDC keeps BREQ asserted until it releases the bus after finishing a DMA burst of programmable length or when the burst is preempted by removing BACK.

A byte transfer occurs in Word Mode when only one byte remains to be transferred or when the system address is odd. The throttling of DMA transfers on the system bus is controlled by the Mode Register. The HDC inserts a programmable number of software Wait States into the DMA bus cycle. Additionally, it inserts hardware Wait States until the memory asserts $\overline{READY}$.

Upper Address Latch Enable (ALEN) may latch the upper address word (A_{16} to A31) in an external address latch. The upper address is only updated if a change demands this update.

Interrupts

The HDC interrupts the host CPU when it has completed the initialization procedure executed after a hardware or software reset or when it has completed a command chain. The initialization interrupt cannot be disabled. The interrupt on command chain completion can be enabled or disabled by the Interrupt Mask bit in the Mode Register.

Disk Control Interface

The Disk Control Interface selects drives and heads and controls the head positioning. It is programmable either to provide a floppy disk type of interface or to conform with the ST506/412 drive interface standard.

Drive Selection

Drives are deselected when $\overline{SELEN}$ is HIGH. The two bits DRIVE < 1:0 > select one of up to four drives. DRIVE < 1:0 > are valid when $\overline{SELEN}$ is LOW. $\overline{SELEN}$ remains LOW as long as the drive is selected. The selected drive acknowledges the selection by activating $\overline{DREADY}$. If $\overline{SELEN}$ is not acknowledged within 2^{16} clocks, then the HDC assumes that the selected drive is not present and generates a time-out error. If $\overline{FAULT}$ is asserted after activating $\overline{DREADY}$, then the HDC deselects the drive and generates a fault error.

Head Positioning

Seek operations are performed via the lines STEP, DIRIN, SEEKCOM, RTZ, and TRK0. Normal seeks pulse the STEP line to move the head to the desired track. Restore may pulse the STEP or RTZ line to move the head to track 0. DIRIN specifies the direction in which the head should move on SEEK pulses.

SEEKCOM is asserted by the drive to indicate that the head has moved to the desired track. Once the drive has acknowledged the completion of a seek by activating SEEKCOM, it must keep SEEKCOM active as long as it is selected or until it receives another seek command; otherwise the HDC issues a drive fault error. When executing the restore or seek commands, the drive must acknowledge the first STEP pulse or the RTZ pulse (SEEKCOM pulsed LOW) within 2^{16} clocks or the HDC will generate a seek error.

Write Protect

The Write Protect line is sampled just prior to execution of a WRITE or FORMAT command. If the line is HIGH, the command is aborted.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage to Ground
Potential Continuous ..... −0.5 to +7.0V
DC Voltage Applied to Outputs
for High Output State ..... −0.5V to +V_{CC}
DC Input Voltage ..... −0.5 to +7.0V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	V_{SS}
Am9580 PG	0°C to 70°C	5V ±5%	0V
Am9580 LC			

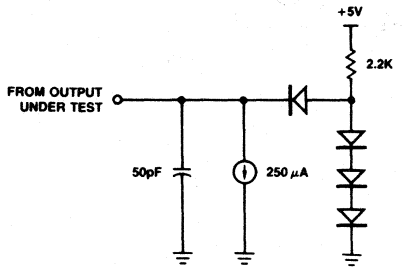
Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} +0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 3.2mA		0.4	V
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		V
I_I	Input Leakage Current	$V_{SS} \leq V_I \leq V_{CC}$		±10	μA
I_O	Output Leakage Current	$V_{CC} \leq V_O \leq V_{SS}$ +.40		±10	μA
I_{CC}	Supply Current				mA
C_{IN}	Input Capacitance	Unmeasured Pins Returned to Ground. f = 1MHz Over Specified Temperature Range			pF
C_{OUT}	Output Capacitance				pF
$C_{I/O}$	Bidirectional Capacitance				pF

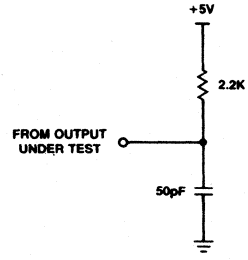
Standard Test Conditions

Standard Test Load



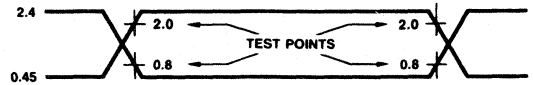
TC002050

Open Drain Test Load



TC002060

Switching Test Input Waveform

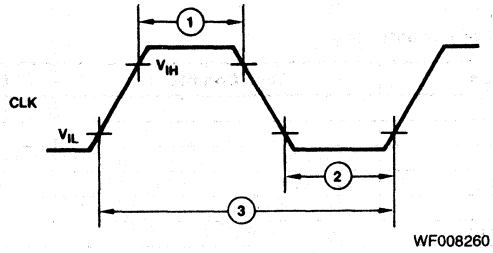


WF008250

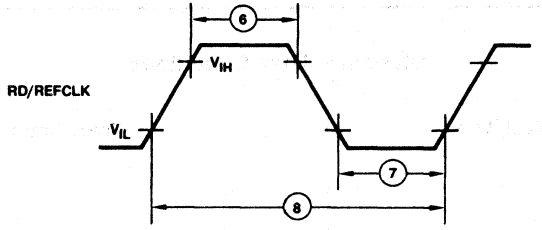
SYSTEM CLOCK AND RD/REFCLK

Number	Parameters	Description	Min	Max	Units
1	t_{CPH}	System Clock High Time	38		ns
2	t_{CPL}	System Clock Low Time	38		ns
3	t_{CPC}	System Clock Cycle Time	100	1000	ns
6	t_{DCPH}	Disk Clock High Time	30		ns
7	t_{DCPL}	Disk Clock Low Time	30		ns
8	t_{DCPC}	Disk Clock Cycle Time	83	10000†	ns

† Can be static if no drive selected.



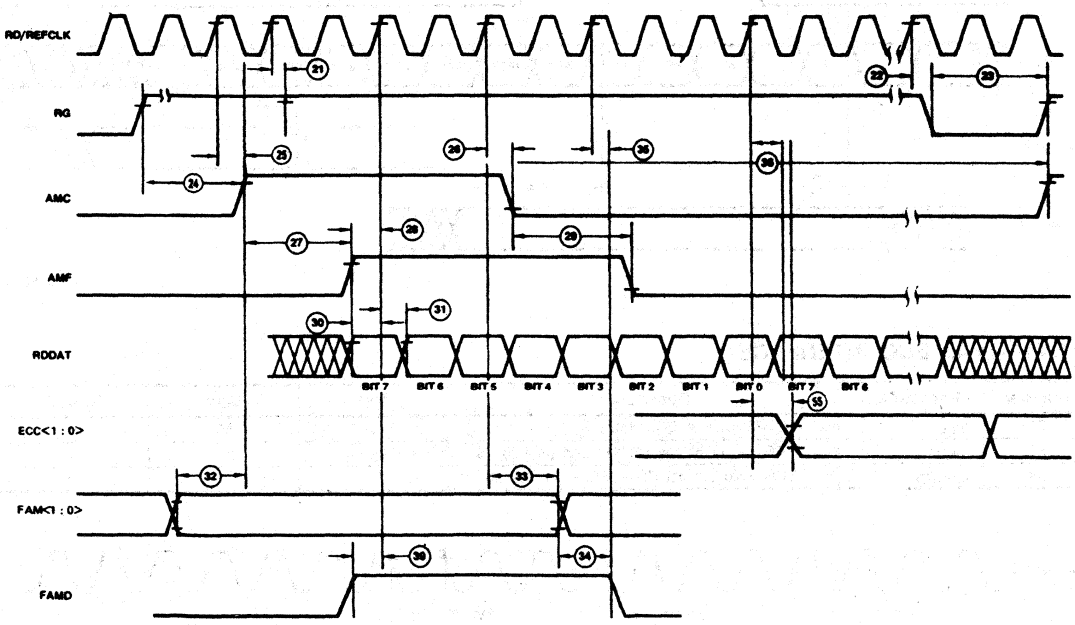
WF008260



WF008270

DISK DATA READ

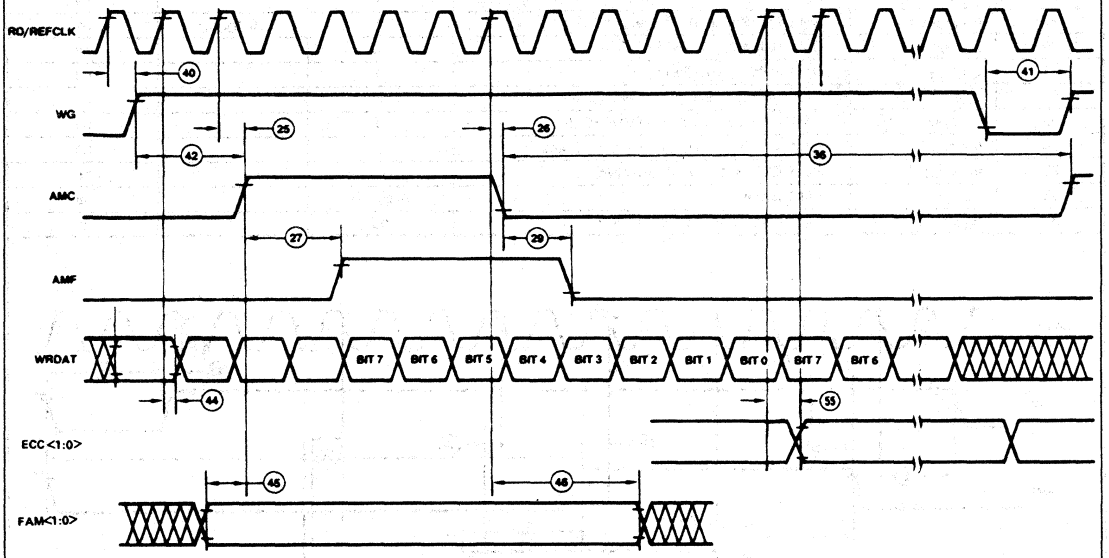
Number	Parameters	Description	Min	Max	Units
21	t_{DRGA}	RD/REFCLK to RG Active	0	50	ns
22	t_{DRGI}	RD/REFCLK to RG Inactive	0	50	ns
23	t_{RGDW}	RG Dwell	8 t_{DCPC}		ns
24	t_{AMCRG}	RG to AMC	8 t_{DCPC}		ns
25	t_{DAMCA}	RD/REFCLK to AMC Active	0	50	ns
26	t_{DAMCI}	RD/REFCLK to AMC Inactive	0	50	ns
27	t_{AMCAMF}	AMC Active to AMF Active	2 t_{DCPC}		ns
28	t_{SUAMF}	AMF Setup to RD/REFCLK	30		ns
29	$t_{DAMCAMF}$	AMC Inactive to AMF Inactive	0	16 t_{DCPC}	ns
30	t_{DATSCC}	Data Setup to RD/REFCLK	30		ns
31	t_{DATHD}	Data Hold to RD/REFCLK	0		ns
32	t_{SUFAM}	FAM <1 : 0> Setup to AMC Active	30		ns
33	t_{HDFAM}	FAM <1 : 0> Hold to RD/REFCLK	0		ns
34	t_{HDFAMD}	FAM <1 : 0> Change to FAMD Inactive	0		ns
35	t_{DFAMDI}	RD/REFCLK to FAMD Inactive	0		ns
36	t_{AMCDW}	AMC Dwell	8 t_{DCPC}		ns
39	t_{SUFAMD}	FAMD Setup to RD/REFCLK	30		ns



WF008281

DISK DATA WRITE

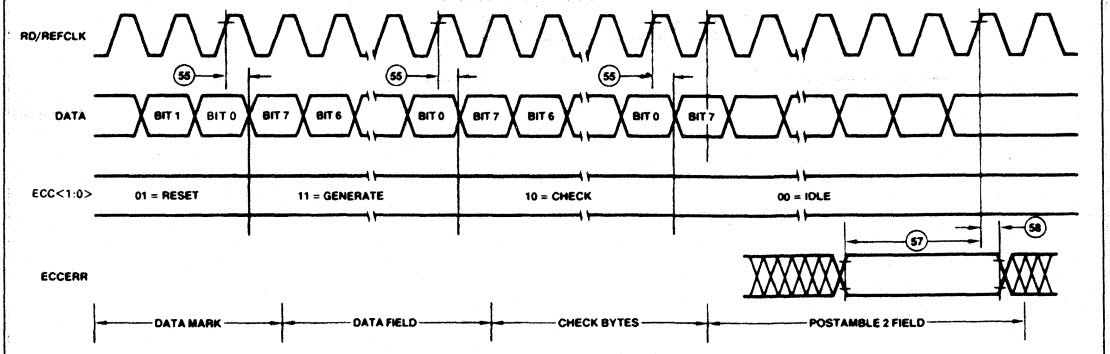
Number	Parameters	Description	Min	Max	Units
40	t_{DWG}	RD/REFCLK to WG Active	0	50	ns
41	t_{WGDW}	WG Dwell	8 t_{DCPC}		ns
42	t_{DWGAMC}	WG Active to AMC Active	8 t_{DCPC}		ns
44	t_{HDWD}	RD/REFCLK to WRDAT	0	50	ns
45	t_{SUFAM}	FAM <1 : 0> Set-up to AMC	t_{DCPC} – 50		ns
46	t_{HDFAM}	FAM <1 : 0> Hold to AMC Inactive	0		ns



WF008291

EXTERNAL ECC INTERFACE

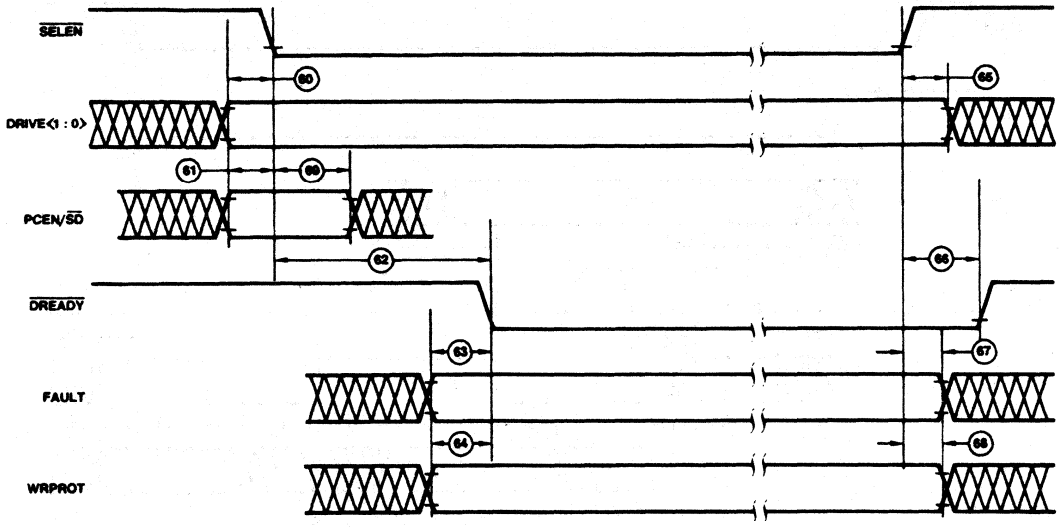
Number	Parameters	Description	Min	Max	Units
55	t_{DECC}	RD/REFCLK to ECC<1:0>	0	50	ns
57	t_{ERRSU}	ERR Setup to Last RD/REFCLK (In Post 2)	8 t_{DCPC}		ns
58	t_{ERRHD}	ERR Hold to Last RD/REFCLK (In Post 2)	0		ns



WF008841

DISK INTERFACE

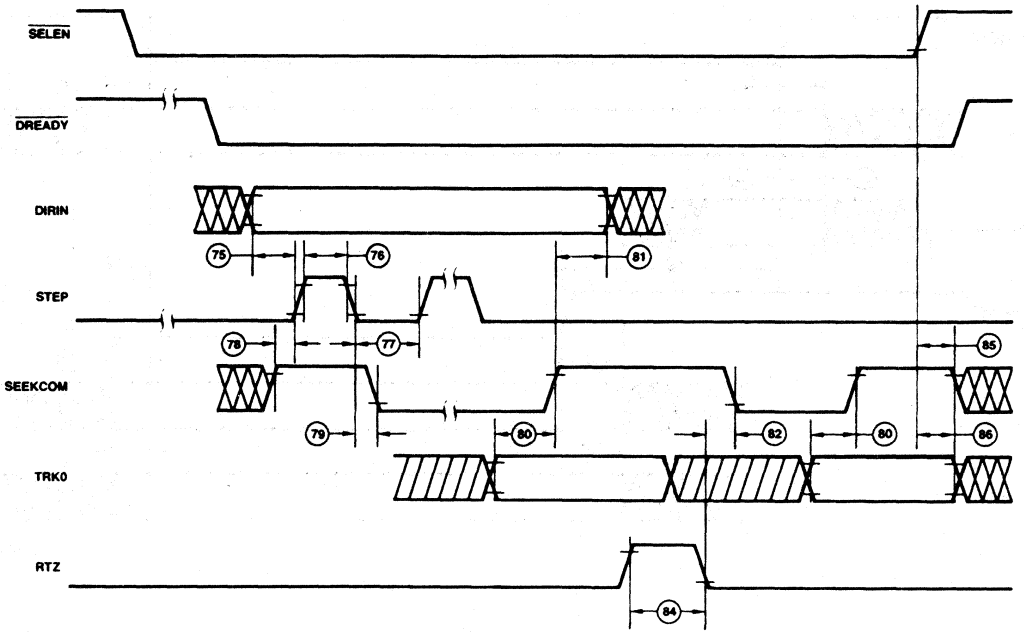
Number	Parameters	Description	Min	Max	Units
60	t_{DRSU}	DRIVESEL <1 : 0> Set-up to $\overline{SELEN}$ Active	400		ns
61	t_{PCSU}	PCEN/$\overline{SD}$ Set-up to $\overline{SELEN}$ Active	400		ns
62	$t_{SELDRDY}$	$\overline{SELEN}$ Active to $\overline{DREADY}$ Active	0	12.8	ms
63	t_{FLTSU}	Fault to $\overline{DREADY}$ Set-up	0		ns
64	t_{WPSU}	WRITEPROT to $\overline{DREADY}$ Set-up	0		ns
65	t_{DSHD}	$\overline{SELEN}$ Inactive to DRIVESEL <1 : 0> Hold	400		ns
66	t_{DRDYHD}	$\overline{SELEN}$ Inactive to $\overline{DREADY}$ Hold	0		ns
67	t_{FLTHD}	$\overline{SELEN}$ Inactive to FAULT Hold	0		ns
68	t_{WPHD}	$\overline{SELEN}$ Inactive to WRITEPROT Hold	0		ns
69	t_{PCHD}	PCEN/$\overline{SD}$ Hold	400		ns



WF008301

DISK INTERFACE

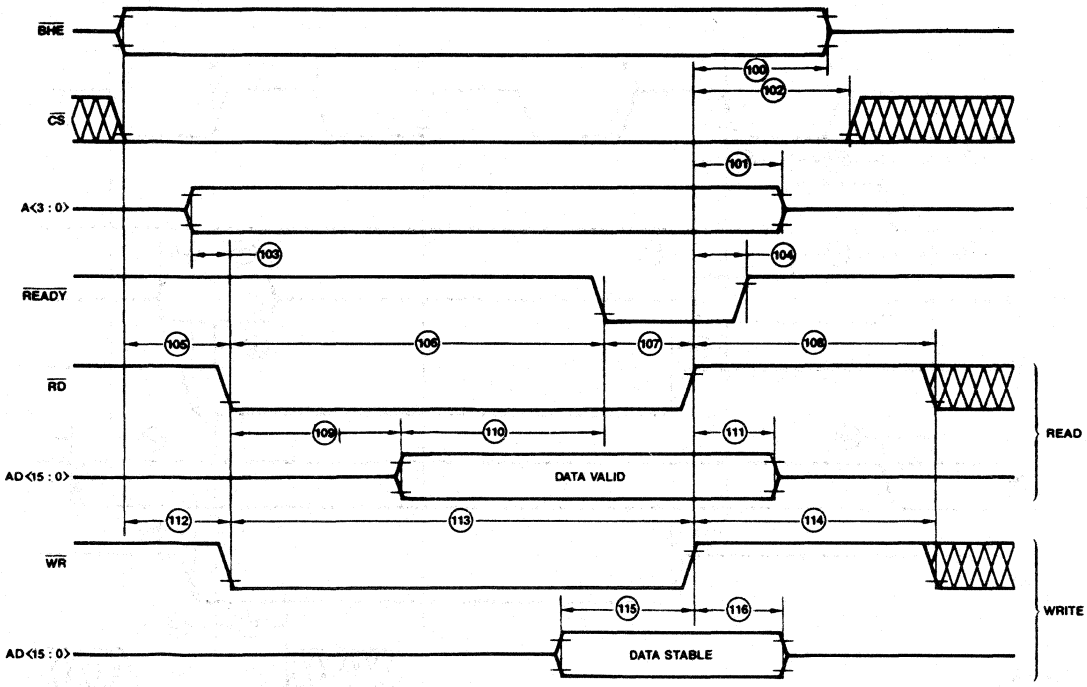
Number	Parameters	Description	Min	Max	Unit
75	t_{DIRSU}	DIRIN Set-up	400		ns
76	t_{STPWD}	STEP High Width	User Programmable		ns
77	t_{STLW}	STEP Low Width	User Programmable		ns
78	t_{SKSU}	SEEKCOM Set-up to STEP	400		ns
79	t_{STPSK}	STEP Low to SEEKCOM Low		10	ms
80	t_{TRKSK}	TRK0 to SEEKCOM Set-up	0		
81	t_{SKDIRI}	SEEKCOM to DIRIN Inactive Hold	400		
82	t_{RTZSK}	RTZ Low to SEEKCOM Low		10	ms
84	t_{RTZWD}	RTZ Pulse Width	User Programmable		ns
85	t_{SELSKI}	$\overline{SELEN}$ Invalid to SEEKCOM Invalid	0		ns
86	$t_{SELTRKI}$	$\overline{SELEN}$ Invalid to TRK0 Invalid	0		ns



WF008311

BUS SLAVE READ/WRITE

Number	Parameters	Description	Min	Max	Units
100	t_{HHRW}	$\overline{BHE}$ to $\overline{RD}$, $\overline{WR}$ Hold	0		ns
101	t_{HADD}	Address Hold	0		ns
102	t_{HCS}	$\overline{CS}$ Hold Time	0		ns
103	t_{SUADD}	Address Set-up	0		ns
104	t_{NRDY}	$\overline{READY}$ Negate Time	0	50	ns
105	t_{CSRD}	$\overline{CS}$, $\overline{BHE}$, to $\overline{RD}$ Set-up	0		ns
106	t_{RRDY}	$\overline{READY}$ Response Time	2 t_{CPC}	16 t_{CPC}	ns
107	t_{HSTB}	Strobe Hold	10		ns
108	t_{RRCV}	Read Recovery	t_{CPC} + 10		ns
109	t_{DATON}	Data Turn On	t_{CPC}	15 t_{CPC}	ns
110	t_{SUDAT}	Data Set-up	t_{CPC} − 30		ns
111	t_{DATOFF}	Data Turn-Off	0	50	ns
112	t_{CSWR}	$\overline{CS}$, $\overline{BHE}$, to $\overline{WR}$ Set-up	0		ns
113	t_{STBWR}	$\overline{WR}$ Strobe Width	t_{CPC}		ns
114	t_{WRCV}	Write Recovery	t_{CPC} + 10		ns
115	t_{SUDATW}	Data Write Set-up	40		ns
116	t_{HDDATW}	Data Write Hold	0		ns

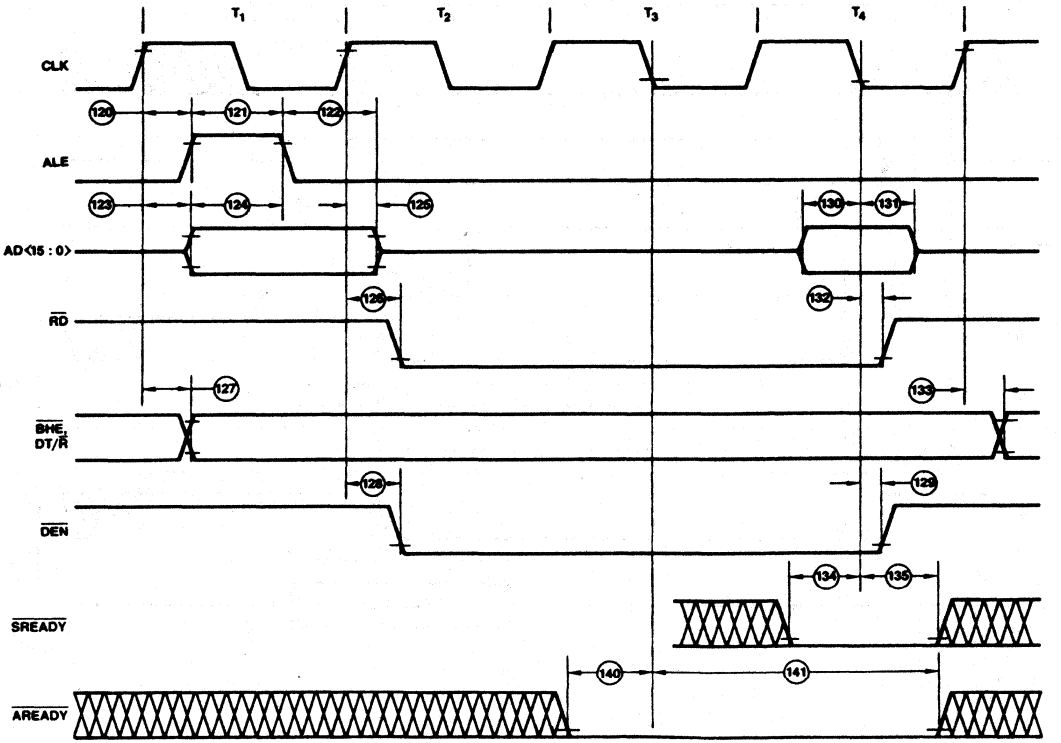


WF008321

BUS MASTER READ

Number	Parameters	Description	Min	Max	Units
120	t_{DALE}	CLK to ALE Active	0	50	ns
121	t_{WALE}	ALE Pulse Width	40		ns
122	t_{ADDHD}	Address <15 : 0> Hold Time	40		ns
123	t_{DADD}	CLK to Address <15 : 0> Valid	0	50	ns
124	t_{SUADD}	Address <15 : 0> Set-up	40		ns
125	t_{IADD}	CLK to Address <15 : 0> Invalid	0	50	ns
126	t_{DRD}	CLK to $\overline{RD}$ Active	0	50	ns
127	t_{VCNTL}	CLK to $\overline{BHE}$, DT/$\overline{R}$ Valid	0	50	ns
128	t_{DDEN}	CLK to $\overline{DEN}$ Active	0	50	ns
129	t_{IDEN}	CLK to $\overline{DEN}$ Inactive	0	50	ns
130	t_{SUDAT}	Data In Set-up	15		ns
131	t_{HDDAT}	Data In Hold	0		ns
132	t_{IRD}	CLK to $\overline{RD}$ Negate	0	50	ns
133	t_{ICNTL}	$\overline{BHE}$, DT/$\overline{R}$ Negate (if changing)	0	50	ns
134	t_{SURBY}	$\overline{SREADY}$ Set-up	15		ns
135	t_{HDRDY}	$\overline{SREADY}$ Hold	0		ns

BUS MASTER READ



WF008331

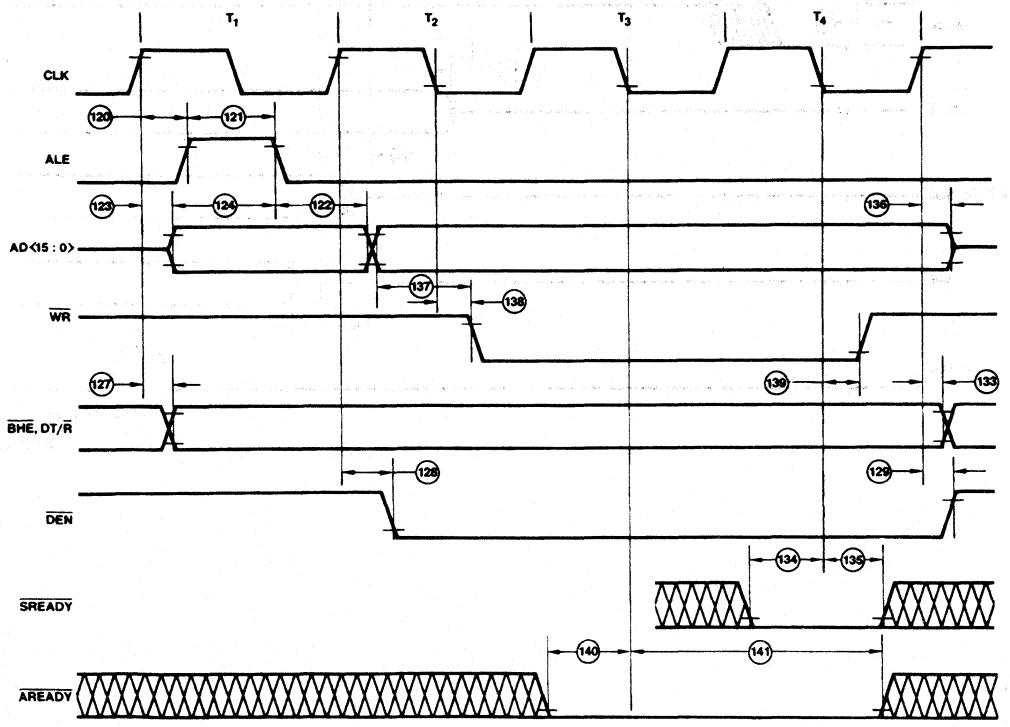
BUS MASTER WRITE

Number	Parameters	Description	Min	Max	Units
136	t_{IDO}	Data Out Invalid	0	50	ns
137	t_{SUDT}	Data Out Set-up	40		ns
138	t_{DWR}	$\overline{CLK}$ to $\overline{WR}$ Active	0	50	ns
139	t_{HDWR}	$\overline{CLK}$ to $\overline{WR}$ Inactive	0	50	ns

$\overline{AREADY}$

Number	Parameters	Description	Min	Max	Units
140	t_{ARSU}	$\overline{AREADY}$ Set-up to CLK Low	15		ns
141	t_{ARHD}	$\overline{AREADY}$ Hold to CLK Low	0		ns

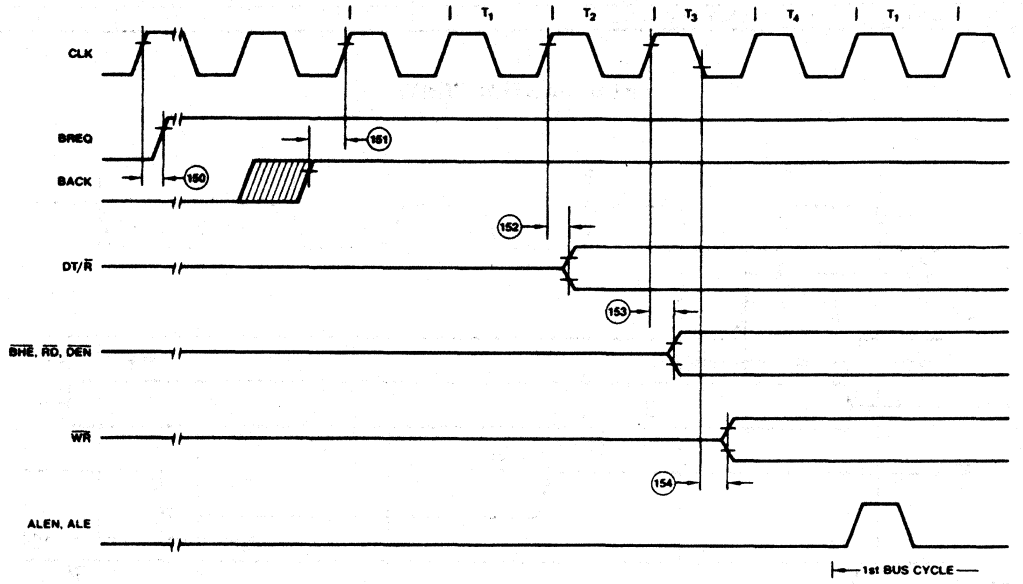
BUS MASTER WRITE



WF008341

BUS REQUEST

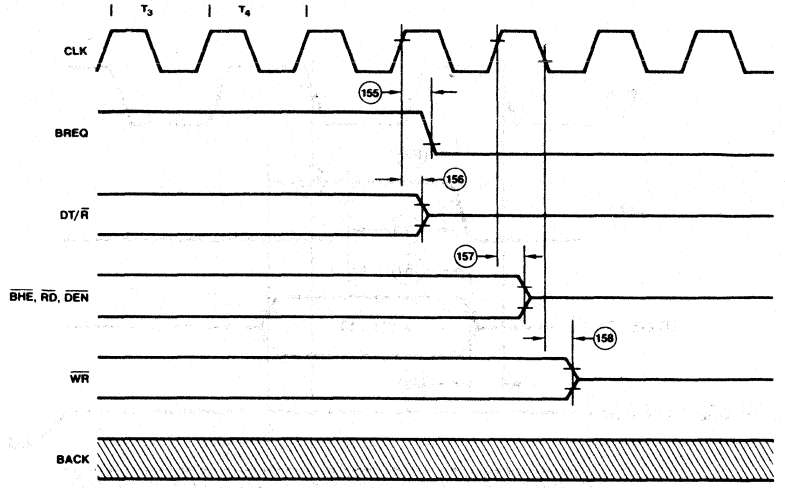
Number	Parameters	Description	Min	Max	Units
150	t_{DBR}	CLK to BREQ Active	0	100	ns
151	t_{DBA}	BACK to CLK Set-up (Synchronous Operation)	20		ns
152	t_{DDTR}	CLK to DT/R Valid	0	100	ns
153	t_{DCNTL}	CLK to $\overline{BHE}$, $\overline{RD}$ and $\overline{DEN}$	0	50	ns
154	t_{DWR}	CLK to $\overline{WR}$ Valid	0	50	ns



WF008360

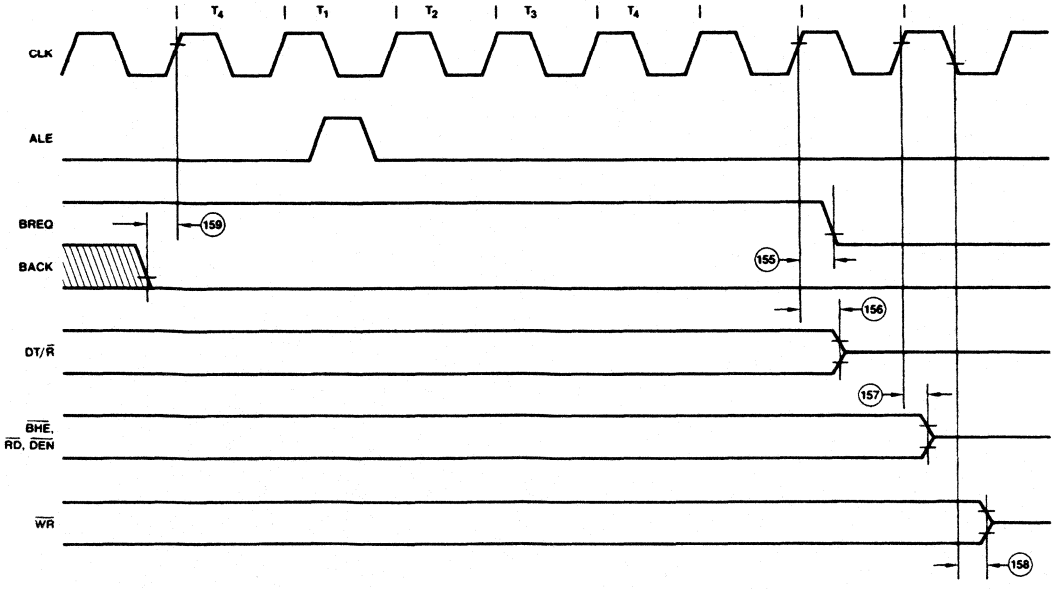
BUS RELEASE

Number	Parameters	Description	Min	Max	Units
155	t_{IBREQ}	CLK to BREQ Inactive	0	100	ns
156	t_{IDTR}	CLK to DT/$\overline{R}$ Invalid	0	100	ns
157	t_{ICNTL}	CLK to $\overline{BHE}$, $\overline{RD}$, and $\overline{DEN}$ Invalid	0	50	ns
158	t_{IWR}	CLK to $\overline{WR}$ Invalid	0	50	ns
159	t_{SBA}	BACK to CLK Set-up (Synchronous Operation)	20		ns



WF008350

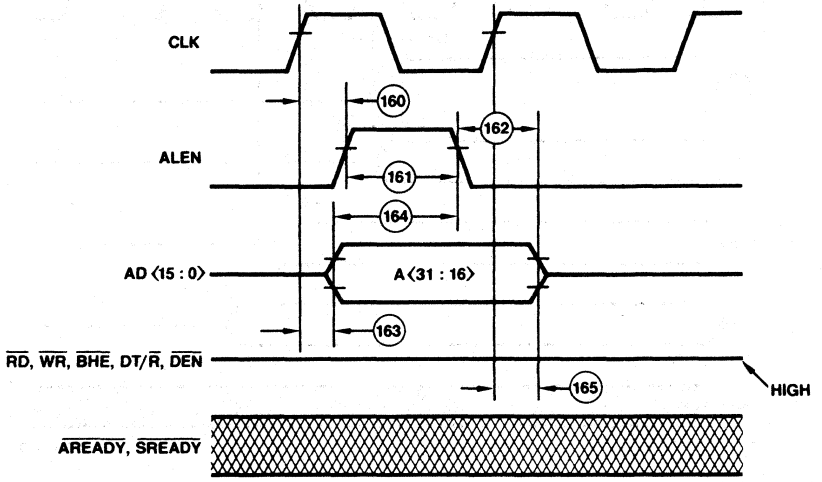
PREEMPTIVE BUS RELEASE



WF008370

UPPER ADDRESS LATCH

Number	Parameters	Description	Min	Max	Units
160	t_{DALEN}	CLK to ALEN Active	0	50	ns
161	t_{WALEN}	ALEN Pulse Width	40		ns
162	t_{DADDH}	AD <15 : 0> Hold Time	40		ns
163	t_{DADD}	CLK to AD <15 : 0> Valid	0	50	ns
164	t_{SUADD}	AD <15 : 0> Set-up Time	40		ns
165	t_{IADD}	CLK to AD <15 : 0> Invalid	0	50	ns



WF008381

Am9581

Floppy/Hard Disk Data Separator
ADVANCED INFORMATION

DISTINCTIVE CHARACTERISTICS

- Complete on-chip Phase-Locked-Loop (PLL)
- On-chip MFM/FM Encoder/Decoder
- Supports: 4 to 16Mbits/sec MFM data rates for Hard Disks
  125 to 500Kbits/sec Single Density (FM) and
  250K to 1Mbits/sec Double Density (MFM) for Floppy Disks
- On-chip Write Pre-compensation
- On-chip Address Mark Generator/Detector
- Provides Clock extracted from the input data for Run-Length-Limited codes

GENERAL DESCRIPTION

The Am9581 Floppy/Hard Disk Data Separator (DDS) is a single-chip solution to several functions associated with reading and writing information to disk drive memory systems. The Am9581 is divided into three sections: read, write, and control.

The read section contains an on-board Phase-Locked-Loop (PLL) to provide a clock signal that tracks the FM/MFM serial data read off the disk. This data is then fed into the MFM/FM decoder to be converted into NRZ data. Also in this section is the Address Mark Detector for both floppy and hard disks.

The write section contains an encoder to serialize NRZ data and its reference clock into a single bit stream of data (MFM or FM) to store on the disk. Also in the write section are the Address Mark Generation and the Write Pre-compensation sections. Each sector on a disk contains several Address Marks. These enable the controller to identify sector types, sector numbers and data fields. The Write Pre-compensation section is used to overcome the problem of bit shifting caused by the method of storage on the media.

The control section handles mainly the drive select and also the head loading logic for floppy disks.

Used in conjunction with the Am9580 Hard Disk Controller (HDC), the chip set provides the total solution for ST-506 and ST-412 disk interface.

Block Diagram

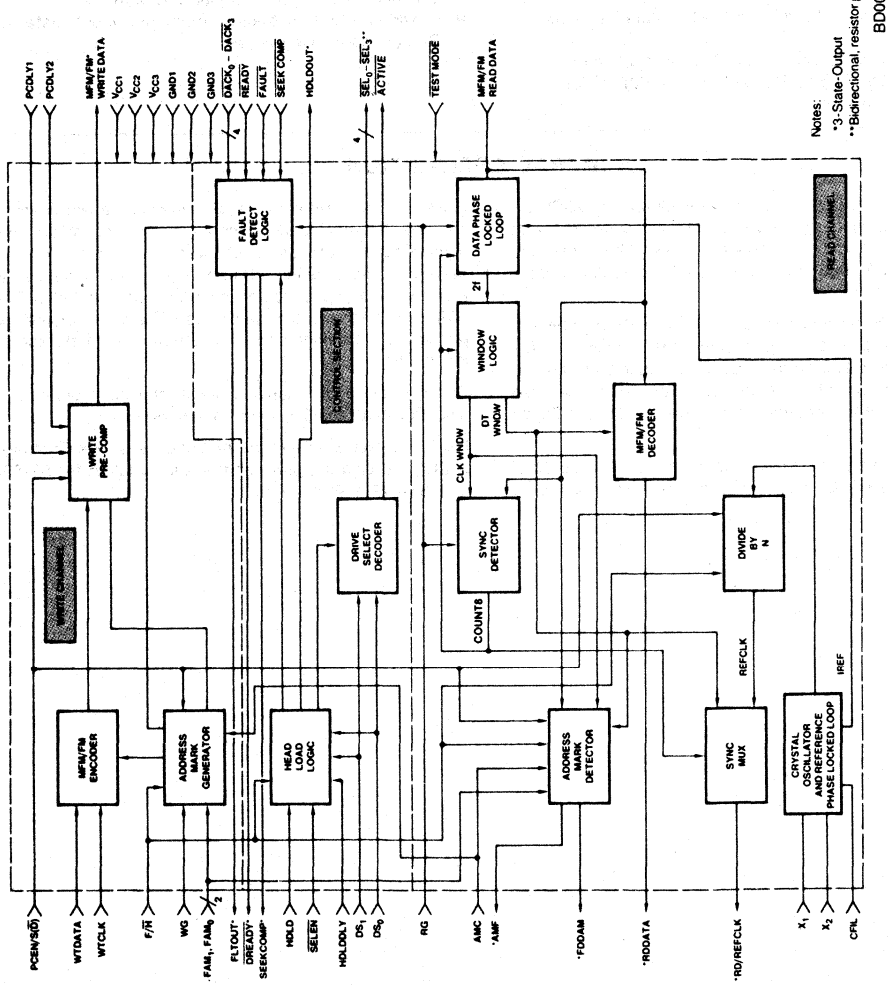


Figure 1.

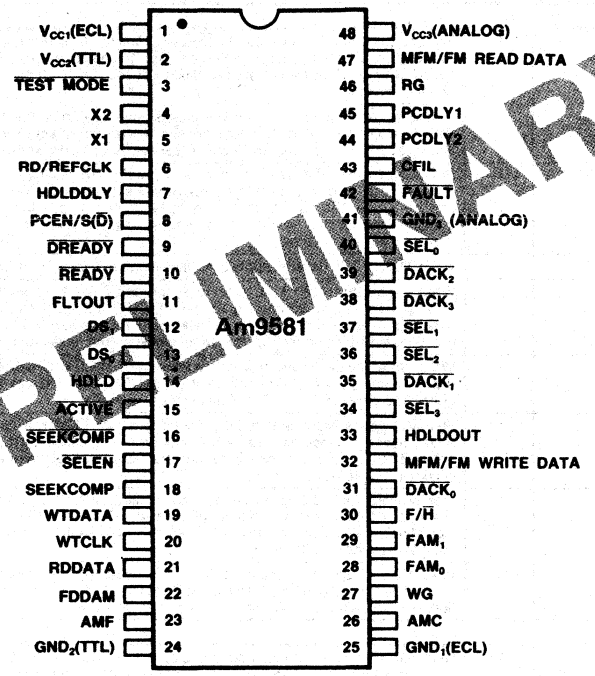
Notes: * 3-State output.
** Bidirectional, resistor pull-up.

05309A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

48-Pin

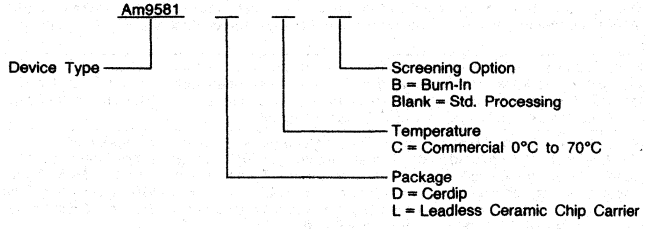


CD005631

Figure 2.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9581	DC (48 leads), LC (52 LCC)

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
WRITE SECTION			
8	PCEN/S($\overline{D}$)	I	Pre-Compensation Enable/Single/$\overline{\text{Double}}$ Density. This pin has two functions which are multiplexed. When a drive is selected ($\overline{\text{SELEN}}$ going from HIGH to LOW), the logic level on PCEN/S($\overline{D}$) is stored internally. A HIGH causes single density operation. A LOW causes double density operation. During a Write operation, a HIGH causes data pre-compensation to take place; otherwise, the data is uncompensated. If single density has been selected when $\overline{\text{SELEN}}$ goes LOW, then PCEN has no effect and the data is always uncompensated.
45, 44	PCDLY1, PCKLY2	I	Pre-Compensation Delay. The ratio of resistors (R_{PC1}, R_{PC2}) connected between PCDLY1 and PCKLY2 pins and V_{CC} sets the pre-compensation delay. The range of pre-compensation delay can be set by the user from 1% to 20% of the bit cell time with an accuracy of ±5% of the chosen value or 1ns, whichever is greater using 1% resistors. The minimum precomp delay that can be set is 2ns.
19	WTDATA	I	Write Data. NRZ data is applied to this input pin for FM or MFM encoding.
20	WTCLK	I	Write Clock. Write Clock is used to synchronize the NRZ write data for magnetic recording. This pin is normally connected to the RD/REFCLK output pin. The user should note that the NRZ input data must meet set-up and hold requirements to the write clock. Skew may be compensated by connecting the write clock to the delayed RD/REFCLK.
27	WG	I	Write Gate. When HIGH, this line enables the WRITE CHANNEL so that writing data onto the disk is made possible. When both WG and RG (Read Gate) are HIGH, a fault condition is generated causing the FLTOUT line to go to HIGH.
29, 28	FAM_1, FAM_0	I	Floppy Address Mark Select Bits. These two input bits are used to select the desired Address Mark related to floppies. These bits are further qualified by F/$\overline{H}$ (Floppy/Hard Disk input select bit) and S/($\overline{D}$) inputs. These lines are also used to set up the chip for use with RLL codes as shown in Table 1.
30	F/$\overline{H}$	I	Floppy/$\overline{\text{Hard}}$ Disk Select. When this pin is tied HIGH, the DDS chip is selected for use with single or double density floppies. When tied LOW, the DDS is selected for hard disks. This input line is also used to select the Address Marks for floppies or hard disks. The selection criteria are listed in Table 1.
32	MFM/FM WRITE DATA	O	Write Data (3-State). Encoded output data used for the actual recording process. MFM format is used for most hard disk and double density floppies, and FM is used for single density floppies. This line is 3-stated when none of the drives connected to the DDS is selected.
READ SECTION			
46	RG	I	Read Gate. A HIGH on this pin enables the READ CHANNEL so that processing of the read back data using the data Phase-Locked-Loop (PLL) can proceed. If both RG and WG are active, a fault condition is generated, causing FLTOUT to go HIGH.
47	MFM/FM READ DATA	I	Read Data. The input to this pin is the MFM/FM encoded information read back from the disk. The Data Separator chip separates the clock and the data information from this MFM/FM data and restores it to the original NRZ format.
5, 4	X_1, X_2	I	Reference Frequency. These two input pins are for connections to an external crystal. An external TTL level clock source can be used instead and tied to the X_1 pin with X2 grounded. The frequency of the crystal to be used for various modes of operation is shown in Table 1. The XTAL frequency is divided down internally to produce the necessary 1f and 2f frequencies. (The 2f reference clock is used internally only for FM encoding.)
26	AMC	I	Address Mark Control. This control input line is used to generate and write the Address Marks during a WRITE operation (with WG) and detect the Address Marks during a READ operation (with RG).
23	AMF	O	Address Mark Found (3-State). Becomes active in response to an active AMC to indicate that the selected type of Address Marks specified in Table 1 have been found during a read and that an Address Mark of appropriate type has been written during a write. The line is 3-stated when none of the drives connected to the DDS are selected.
22	FDDAM	O	Floppy Deleted Data Address Mark (3-State). For single density floppies, this output pin indicates the detection of a deleted Data Address Mark. During Read operations, this status signal is generated by the Am9581 only when a request is made by the HDC. For MFM hard disks and double density floppies, this pin will always be LOW when the DDS is selected. This pin is 3-stated when none of the drives connected to DDS is selected. In RLL hard disk mode, this pin is used to output 2f clock synchronous with RDDATA.
21	RDDATA	O	NRZ Read Data (3-State). This line contains the binary NRZ data decoded from the MFM/FM READ DATA. In RLL mode, the input data on the MFM/FM READ DATA pin is passed directly out to the RDDATA output without any decoding taking place.
6	RD/REFCLK	O	Read/Reference Clock (3-State). Read/Reference clock is synchronously multiplexed such that during a READ operation, the clock source is the RDCLK which is derived from the MFM/FM READ DATA bit stream. When not doing a valid READ operation, the clock source is derived from the input to pins X_1 and X_2. The switching from RDCLK to REFCLK is a glitch-free operation. The pin is 3-stated when none of the drives connected to DDS is selected. For RLL hard disks, this pin always outputs REFCLK.
43	CFIL	I	Filter Capacitor. The filter capacitor for the reference PLL is connected between this pin and ground.
CONTROL SECTION			
14	HDLD	I	Head Load. This input pin is applicable to floppies only and is used for controlling the head loading (when HIGH) and unloading (when LOW) mechanism.

05309A

Refer to page 7-1 for Essential Information on Military Devices

Pin Description (Cont.)

Pin No.	Name	I/O	Description
33	HDLDOUT	O	Head Load to Drive (3-State). This is the Head Load signal to the floppy drive. When HDLD from the controller chip goes HIGH, HDLDOUT goes HIGH immediately. When HDLD goes LOW, there is a delay (set by HDLDDLY) before HDLDOUT goes LOW. Refer to the timing diagram of Figure 3 for the different cases involving the HDLDOUT signal.
7	HDLDDLY	I	Head Load Delay. A resistor R_X connected between this input pin and V_{CC}, and a capacitor C_X connected between this input pin and ground establish a time delay (selected by user) associated with the unloading of the head in floppies. This time delay is necessary so that a drive may be deselected and selected again without unloading and reloading the head. The user may set the time delay from a minimum of 5ms to a maximum of 500ms with a maximum tolerance of 0.5ms or ±5%, whichever is larger.
17	$\overline{SELEN}$	I	Select Enable. An active LOW signal to this pin enables the drive specified by the drive select bits DS_1, DS_0. The falling edge of $\overline{SELEN}$ and the status of the PCEN/S($\overline{D}$) pin determine single- or double-density operation for floppies.
13, 12	DS_0, DS_1	I	Drive Select. These two pins specify the selection of up to four drives, as indicated below: DS_1 / DS_0 / $\overline{SEL}$ PIN ACTIVATED LOW / LOW / $\overline{SEL_0}$ LOW / HIGH / $\overline{SEL_1}$ HIGH / LOW / $\overline{SEL_2}$ HIGH / HIGH / $\overline{SEL_3}$
40, 37, 36, 34	$\overline{SEL_0}$ – $\overline{SEL_3}$	O	Select$_{0\text{-}3}$ (Resistor Pull-Up). These four output lines are the decoded DS_1, DS_0 bits used to select up to four drives. When a particular $\overline{SEL}$ pin is grounded, the DDS will not acknowledge a request for this drive from the HDC (all interface lines to the HDC will be tri-stated). In a typical floppy/Winchester disk drive system with 2 DDS's each attached to 2 drives, the two remaining $\overline{SEL}$ lines are grounded. In the example on page 17, one DDS interfaces with drives 2 and 3 and has $\overline{SEL_0}$ and $\overline{SEL_1}$ grounded. Grounding the 2 $\overline{SEL}$ lines prevents this DDS from acknowledging requests for drives 0 and 1.
31, 35, 39, 38	$\overline{DACK_0}$ – $\overline{DACK_3}$	I	Drive Acknowledge (Schmitt Trigger Inputs). These input pins reflect the response of an individual drive, acknowledging that it has been selected. Therefore, these inputs are used with the $\overline{SEL_0}$-$\overline{SEL_3}$ status to test whether the drive that acknowledged is the one being selected; otherwise, a fault condition will be detected.
10	$\overline{READY}$	I	Ready (Schmitt Trigger Input). This input pin indicates that the drive selected is ready to READ, WRITE, or SEEK (access a track).
9	$\overline{DREADY}$	O	Drive Ready (3-State). This status output pin is used to indicate that the selected drive is ready to READ, WRITE or SEEK. When a fault condition is sensed by the F/HDDS, this line is set LOW so that the controller can detect the fault by sensing FLTOUT. The line is 3-stated if none of the drives connected to F/HDDS are selected.
42	$\overline{FAULT}$	I	Fault (Schmitt Trigger). This pin indicates a fault condition when selecting a drive or in the drive itself. An active $\overline{FAULT}$ is considered only after the Drive Acknowledge bit corresponding to the drive selected is active LOW.
11	FLTOUT	O	Fault Out (3-State). This output pin is a status output of the fault detection logic. The fault conditions detected are: 1. WG and RG both active. 2. Active FAULT from the selected drive. 3. Unselected drives acknowledging drive selection. 4. More than one drive acknowledge in response to a drive selection. The pin is 3-stated when none of the drives connected to F/HDDS is selected.
16	$\overline{SEEKCOMP}$	I	Seek Complete (Schmitt Trigger). This input line indicates that a SEEK operation has successfully been completed and that the head has settled on the desired track. This signal is generated by the drive.
18	SEEKCOMP	O	Seek Complete (3-State). This status output line is the same as $\overline{SEEKCOMP}$ for Hard Disk mode. In Floppy mode this output is driven by the Head Load Delay logic (see Detailed Description for details). The line is 3-stated if none of the drives connected to F/HDDS is selected.
3	$\overline{TEST\ MODE}$	I	This line is used to test the chip logic independent of the PLL.
15	$\overline{ACTIVE}$	O	(TTL.) Control for external buffers, if used. This status output pin is used to indicate that none of the drives is selected and all of the control output pins are in tristate.
25, 24, 41	GND_1 – GND_3		Ground.
1, 2, 48	VCC_1 – VCC_3		Power Supply + 5V.

DETAILED DESCRIPTION

The Am9581 simplified block diagram shown in Figure 1 is divided into three sections - (1) the CONTROL section, (2) the WRITE section, and (3) the READ section. The WRITE section and the READ sections are independent of each other and under the supervision of the CONTROL section. The functions of each section are described below.

Control Section

This section consists of the HEAD LOAD LOGIC, DRIVE SELECT DECODER, and the FAULT DETECT LOGIC. The HEAD LOAD LOGIC is enabled only if the DDS is to be used in floppy disk drives. The HEAD LOAD LOGIC block controls the loading and unloading of the floppy disk drive head as specified in the timing diagram of Figure 3. Each of the three operations is described below.

Normal Operation

The controller selects a drive by asserting $\overline{SELEN}$ LOW, causing one of the $\overline{SEL}_{0\text{-}3}$ lines to be asserted. Once the drive is selected, the controller can request the DDS to load the head by asserting HDLD. The DDS responds with HDLDOUT going HIGH and asserts SEEKCOMP after a time delay ($t_{HDLDDLY}$) set by the resistor R_X and the capacitor C_X at the HDLDDLY pin. This time delay set by the user is to accommodate the setting time associated with the head loading and unloading: $t_{HDLDDLY}$ can be calculated as $(1.1\ R_X\ C_X) \pm 5\%$.

To deselect the drive, the controller drives $\overline{SELEN}$ HIGH and HDLD LOW. The actual deselection of the drive and the unloading of the head do not occur until after $t_{HDLDDLY}$.

Same Drive Reselected

The selection of the drive and the loading of the head is the same as described in the normal operation. However, if the same drive (DRIVE 0) is deselected and selected again within the time interval defined by $t_{HDLDDLY}$, then the head can remain loaded (HDLDOUT remaining HIGH). In this case, a different track can be assessed while the head remains loaded.

Different Drive Selected

When a drive (DRIVE 0 for example) is deselected and then a different drive (DRIVE 1) is selected during the time interval $t_{HDLDDLY}$, HDLDOUT will remain HIGH. This means that the head of the newly selected drive will be loaded automatically as shown in Figure 3.

Hard disk drives do not have head loading/unloading mechanisms since the heads are basically floating when the drive is running. (When a hard disk drive is not running, the heads are in contact with the disk in a specified area called the landing zone.) Each time a different track is to be accessed, a hard disk drive is said to be in SEEK mode. When the desired track is found, the drive issues an active $\overline{SEEKCOMP}$. This status is indicated to the controller by the Am9581 as an active $\overline{SEEKCOMP}$.

Write Section

FM/MFM Encoding

The serial NRZ data pattern is transformed into pulses that occur in a time window specified as the bit cell time. The bit cell time is derived from a very stable reference frequency which can be labeled as the WTCLK. The encoding process takes place during a write operation – i.e., WG is HIGH and RG is LOW. The result of the encoding process is the transformation of the NRZ data and its reference clock into a single bit stream consisting of a CLOCK and/or a DATA pulse for each bit cell depending upon the NRZ value within the bit cell time.

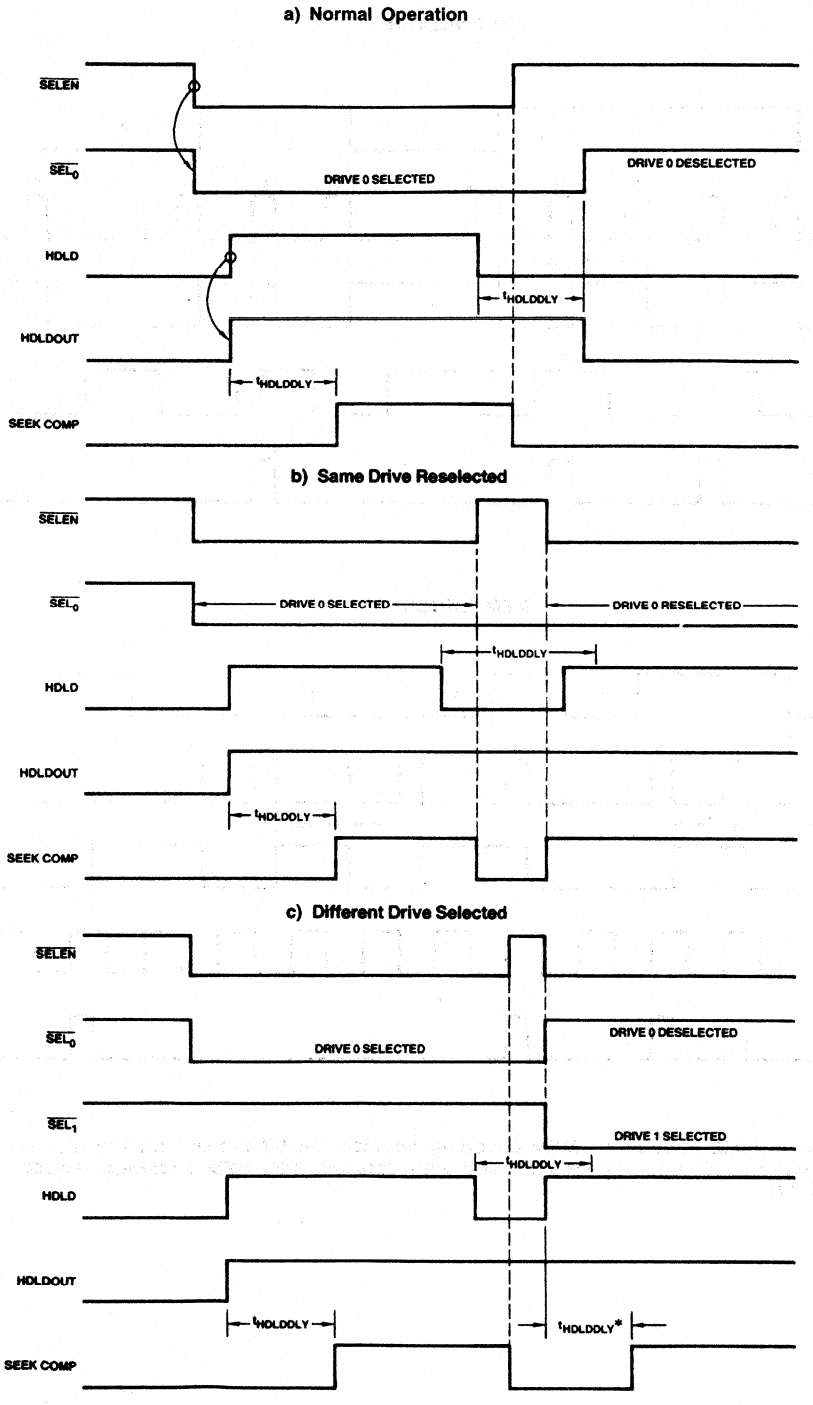
For FM encoding, a CLOCK pulse is always inserted at the beginning of the bit cell. When NRZ data is "1" during the bit cell time, a DATA pulse is inserted in the middle of the bit cell; otherwise, no DATA pulse is inserted. Therefore, for FM encoding, a CLOCK pulse and a DATA pulse can both be present within the bit cell time (which is 4μs for single density floppies running at 250 Kbit/sec).

In MFM encoding, when the NRZ data is "1" during the bit cell time, a data pulse is inserted in the middle of the bit cell. If the NRZ data is "0" during the bit cell time, no data pulse is inserted. However, unlike FM, the clock pulses are not automatically inserted at the beginning of each bit cell. The clock pulses are inserted at the beginning of a bit cell if and only if the NRZ data is "0" during the current bit cell as well as the previous bit cell. Otherwise, no clock pulses are written (see Figure 4). Therefore, in MFM, only a CLOCK pulse or a DATA pulse can be present within the bit cell time (which is 2μs for a 500Kbit/sec double density floppy and 200ns for a 5Mbit/sec hard disk drive). FM and MFM encoding are shown in Figure 4.

Address Mark Generation

Each sector on the disk contains several Address Marks. These enable the controller to identify sector types, sector numbers, and data fields. So that the Address Marks are unique and always distinguishable from all possible data patterns, the encoding rules (either for FM or MFM) are deliberately violated when an Address Mark is written. This is done by deleting some of the clock bits in the encoded clock/data pattern. When a sector is read back from the disk, these missing clocks are identified, and then the controller is assured that correct synchronization has taken place. The DDS will generate all the standard IBM Address Marks for floppy (IBM format) and Winchester (ST506/SA1000 formats). AMC (ADDRESS MARK CONTROL) is sampled on the rising edge of WTCLK when a WRITE operation is taking place. An Address Mark is then inserted, AMF (ADDRESS MARK FOUND) is brought HIGH in acknowledgement.

The type of Address Mark generated is dependent on FAM_0, FAM_1, and the operating mode, i.e., floppy or hard, and single or double density. The type of Address Mark selected is listed in Table 1.

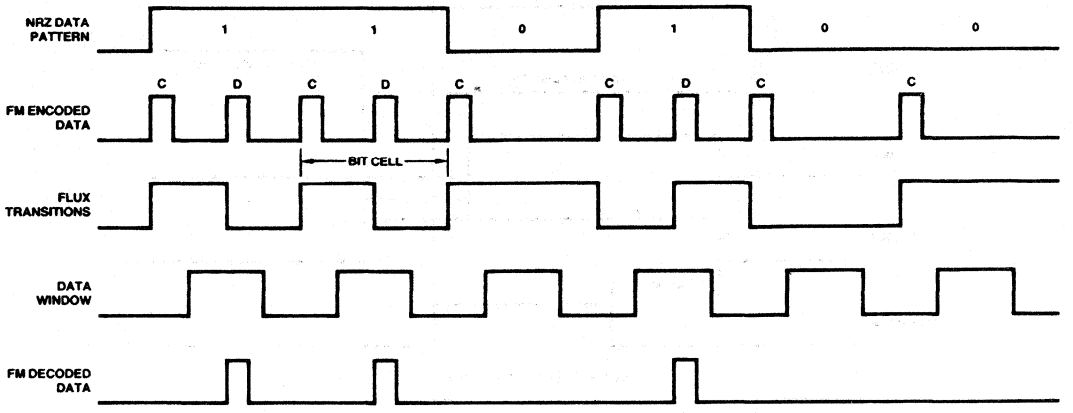


*This time out starts from the falling edge of $\overline{\text{SELEN}}$ or rising edge of HDLD, whichever occurs later.

WF008391

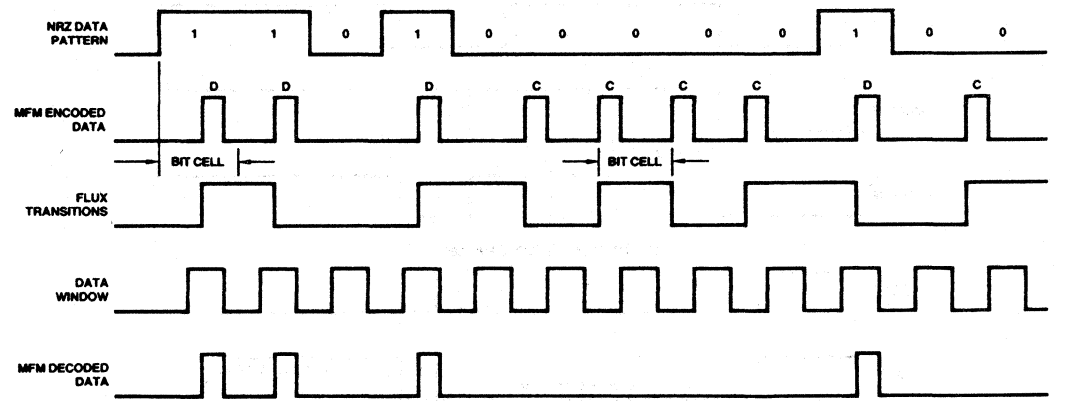
Figure 3. Floppy Timing Sequence

FM ENCODING



WF008410

MFM ENCODING



WF008420

Figure 4. FM vs MFM Encoding - MFM encoding doubles the bit density on the disk by replacing clock bits (C) used in FM encoding with data bits (D). MFM encoding reduces the bit cell by 1/2.

The different Address Marks listed below are formed by combinations of the Hexadecimal Data and Clock Patterns.

SINGLE DENSITY FLOPPY

TYPE	Data Pattern	Clock Pattern	Number of Bytes
IXAM	FC	D7	1
IDAM	FE	C7	1
DAM	FB	C7	1
DDAM	F8	C7	1

DOUBLE DENSITY FLOPPY*

TYPE	Data Pattern	Clock Pattern	Number of Bytes
IXAM	C2	14	3
DAM and IDAM	A1	0A	3

HARD DISK*

TYPE	Data Pattern	Clock Pattern	Number of Bytes
DAM and IDAM	A1	0A	1

***Note:** The ID Address Marks and Data Address mark for double density floppy and hard disks are normally qualified by an extra byte following the Address Mark. However, these extra bytes are normally encoded (no missing clocks), which the controller can easily detect during a READ operation.

Write Pre-Compensation

Bit shifting is a phenomenon caused by the flux changes stored on the disk interacting with each other. Its effect is to make nearby flux changes move away from each other, creating a timing uncertainty which can cause errors when data is read. This phenomenon is much worse on the inside tracks of a disk because the flux changes are closer together. To overcome this data interaction, the WRITE DATA stream is pre-compensated; i.e., the direction of each bit shift is anticipated and the bit is moved in the other direction by the Am9581 before being written. When PCEN is HIGH, pre-compensation is enabled and each bit will be made either EARLY, NOMINAL, or LATE, depending on its interaction with its neighbors. (In FM mode PCEN has no effect.) The amount of time shift between NOMINAL and LATE is set by two external resistors on the PCDLY pins. The pre-compensation is shown in Figure 5.

For RLL codes, NRZ to RLL encoding, write pre-compensation and address mark detection have to be performed by an external circuit.

Read Section

The read section of Am9581 consists of a data Phase-Locked-Loop (PLL), Window Logic, Sync Field Detector, Address Mark Detector, MFM/FM Decoder, Synchronized Multiplexer, crystal controlled oscillator, a reference PLL, and a divide by N counter as shown in Figure 1.

Data Phase-Locked-Loop (PLL)

The main function of the data PLL is to provide a clock signal (shown as 2f in Figure 1) that closely tracks the FM/MFM serial data read off the disk. The 2f signal is then used by the window logic to generate clock and data windows.

When the chip is in the write mode (WG active), the data PLL is synchronized to the REFCLK derived from the XTAL oscillator and the reference PLL. When the data is read from the disk (RG active), the data PLL is locked to the data stream from the disk.

The Window Logic

The main function of the window logic is to generate clock and data windows using the 2f signal provided by the data PLL. When the Read Gate (RG) is asserted, the window logic assigns clock and data windows arbitrarily.

The Sync Field Detector

The sync field detector looks for the sync field consisting of 8 consecutive pulses in clock windows. When this pattern is detected, the COUNT 8 signal is activated, freezing the window polarity. If the sync pattern is not found, the window logic flips the window polarities so that the sync detector can continue to look for the sync pattern. The apparent sync field ending is seen as a pulse in the data window. This action arms an address mark time out; i.e., after 8 bit cells (or 24 for double density floppy) an address mark must be found or the state machine is reset back to the lowest level of search – looking for 8 apparent zeros.

Address Mark Detector

The detection of various address marks for floppies and hard disks is performed by this section. The type of address mark to be detected is determined by the signals S/D, F/H, FAM_0 and FAM_1 as shown in Table 1. In response to an active AMC, AMF will be asserted to indicate that the desired address mark has been found during a read, or an appropriate address mark has been written during a write. FDDAM is asserted in the FM floppy mode upon detection of a floppy deleted data address mark during a read.

MFM/FM Decoder

The MFM/FM decoder converts the incoming MFM/FM data to NRZ data. If an MFM/FM pulse occurs in a data window, the NRZ data is decoded as a"1." If no pulse occurs in the data window, the NRZ data is decoded as a "0."

SYNC MUX

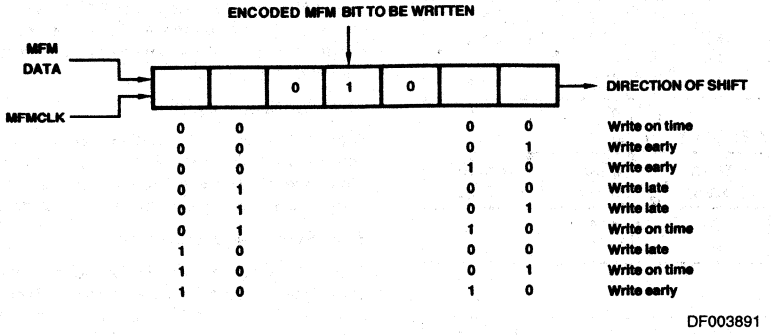
The SYNC MUX is a synchronized multiplexer that outputs either the REFCLK derived from the XTAL oscillator and the reference PLL or the RDCLK derived from disk data. The multiplexer outputs REFCLK while writing and also while reading until a sync field has been detected. Once the sync is detected, it switches over to RDCLK without any glitches. For RLL hard disks, the output is always REFCLK.

XTAL Oscillator, Reference PLL and Divide by N Counter

The XTAL oscillator, the reference PLL and the divide by N counter provide REFCLK to the SYNC MUX. The clock is also used in the data PLL to prevent VCO from drifting off from the center of its tuning range while it is not reading data off the disk. The appropriate divide ratio is selected internally as shown in Table 1.

WRITE PRE-COMPENSATION

Pre-compensation is for the MFM case only. FM does not require pre-compensation. The bit to be written is shifted early or late with respect to a nominal delay. The nominal delay is selected when data is to be written on time. Action to be taken is made by examining a 7-bit register.



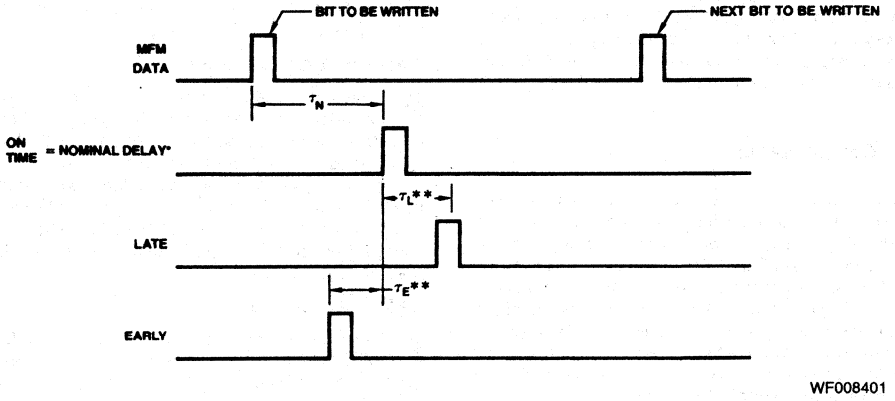


Figure 5.

*Nominal delay = τ_N
**$\tau_L = \tau_E$ = 1% – 20% bit cell time

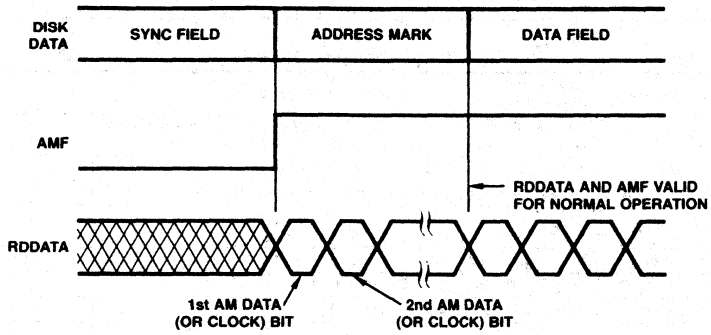
TABLE 1.

F/$\overline{H}$	PCEN/S($\overline{D}$)	FAM_1	FAM_0	Address Mark Selected	Mode	XTAL FREQ
1	1	0	0	Index (IXAM)	FM Floppy	32 x Bit Rate FREQ
1	1	0	1	ID (IDAM)	FM Floppy	
1	1	1	0	Data (DAM)	FM Floppy	
1	1	1	1	Deleted Data (DDAM)	FM Floppy	
1	0	0	0	Index (IXAM)	MFM Floppy	16 x Bit Rate FREQ
1	0	1	0	ID or Data	MFM Floppy	
0	X	0	0	Normal A1 Address Mark	MFM Hard Disk	Bit Rate FREQ
0	X	1	1	Any Apparent Address Mark	Dump Clock** – Hard Disk	Bit Rate FREQ
0	X	1	0	None	RLL Hard Disk	Code Rate FREQ*
0	X	0	1	Any Apparent Address Mark	Dump Data – Hard Disk	Bit Rate FREQ
1	0	0	1	Any Apparent Address Mark	Dump Data – Floppy Disk	16 x Bit Rate FREQ
1	0	1	1	Any Apparent Address Mark	Dump Clock – Floppy Disk	16 x Bit Rate FREQ

*Code Rate Frequency = "2f " frequency of PLL.

**Dump is a mode whereby data (or clock) is passed to the HDC on an apparent START of Address Mark (end of apparent SYNC field). This is to enable the recovery of sectors which could not otherwise be read due to corruption of the Address Mark.

DUMP MODE



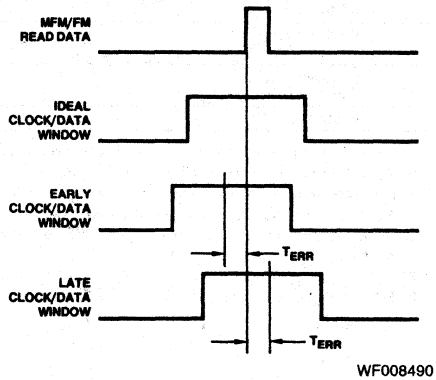
WF008480

TABLE 2.
DATA PHASE LOCKED-LOOP SPECIFICATIONS

Supply Voltage	+5V = 10%
Operating Temperature Range	0 to 70°C (Plastic DIP)
Acquisition Time*	16x1/f_D seconds MAX
Capture Range*	±6% of f_D MIN
Decode Window Error*	±2.5ns or ±2% of 1/f_D, MAX whichever larger

*f_D = bit rate frequency

TABLE 3.



WF008490

DECODE WINDOW ERROR

*Note: f_D = bit rate frequency

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage to Ground Potential Continuous ..... −0.5 to +7.0V
DC Voltage Applied to Outputs for High Output State ..... −0.5V to +V_{CC}
DC Input Voltage ..... −0.5 to +7.0V
DC Output Current into Outputs 30mA
DC Input Current ..... −30 to +5.0mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	V_{SS}
Am9581 DC	0°C to 70°C	5V ±10%	0V
Am9581 LC	0°C to 70°C	5V ±10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

Am9581

ELECTRICAL CHARACTERISTICS

The following conditions apply unless otherwise specified:

COM'L $T_A = 0$ to +70°C $V_{CC} = 5.0V \pm 10\%$ (MIN = 4.50V, MAX = 5.50V)

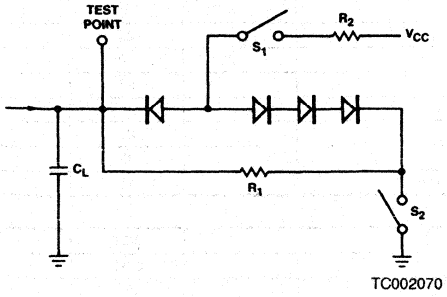
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions (Note 1)		Min	Typ (Note 2)	Max	Units
V_{OH}	Output HIGH Voltage	V_{CC} = MIN, $V_{IN} = V_{IH}$ or V_{IL}	$I_{OH} = -1mA$ (Note 6)	2.4			Volts
V_{OL}	Output LOW Voltage	V_{CC} = MIN, $V_{IN} = V_{IH}$ or V_{IL}	I_{OL} = 4mA (Mil), 8mA (Com'l) (Note 6) I_{OL} = 48mA (Note 5)			0.5	Volts
V_{IH}	Input HIGH Level	Guaranteed Input HIGH Voltage for All Inputs, Note 7		2.0			Volts
V_{IL}	Input LOW Level	Guaranteed Input LOW Voltage for All Inputs, Note 7				0.8	Volts
V_I	Input Clamp Voltage	V_{CC} = MIN, $I_{IN} = -18mA$				−1.2	Volts
I_{IL}	Input LOW Current	V_{CC} = MAX, $V_{IN} = 0.5V$	(Note 7)			−0.4	mA
I_{IH}	Input HIGH Current	V_{CC} = MAX, $V_{IN} = 2.4V$	(Note 7)			20	µA
I_I	Input HIGH Current	V_{CC} = MAX, $V_{IN} = 5.5V$	(Note 7)			1.0	mA
I_O	Off-State (High-Impedance) Output Current	V_{CC} = MAX	$V_O = 0.4V$			−50	µA
			$V_O = 2.4V$			50	
I_{SC}	Output Short Circuit Current (Note 3)	V_{CC} = MAX	(Note 6)	−15		−50	mA
I_{CC}	Power Supply Current (Note 4)	V_{CC} = MAX	0 to +70°C		TBD	TBD	mA
V_{t+}	Positive Going Threshold Voltage		Note 8		1.6		Volts
V_{t-}	Negative Going Threshold Voltage		Note 8		0.8		Volts
V_{tH}	Hysteresis		Note 8		0.8		Volts

Notes: 1. For conditions shown as MIN or MAX, use the appropriate value specified under Operating Ranges for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. All tristate outputs are in the high-impedance state.
5. $\overline{SEL_0} - \overline{SEL_3}$ outputs only. These have resistor-pullups.
6. All outputs except $\overline{SEL_0} - \overline{SEL_3}$.
7. Logic inputs. (Does not include X_1, X_2, PCDLY1, PCDLY2, CFIL, HDLDDLY, $\overline{DACK_0} - \overline{DACK_3}$, $\overline{READY}$, $\overline{FAULT}$, $\overline{SEEKCOMP}$.)
8. Schmitt-trigger inputs only ($\overline{DACK_0} - \overline{DACK_3}$, $\overline{READY}$, $\overline{FAULT}$, $\overline{SEEKCOMP}$).

SWITCHING CHARACTERISTICS

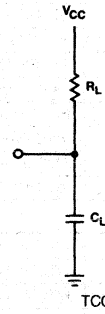
THREE-STATE OUTPUTS



TC002070

C_L, R_1, R_2 to be determined.
All diodes IN916 or IN3064.

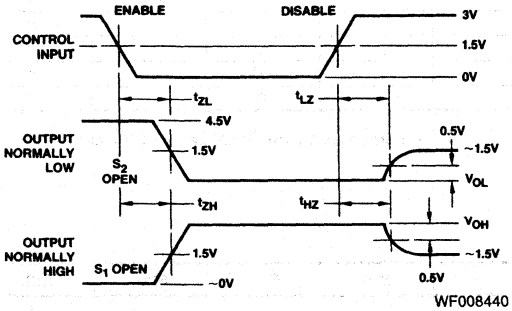
$\overline{SEL_0}$ – $\overline{SEL_3}$ OUTPUTS



TC002080

R_L, C_L to be determined.

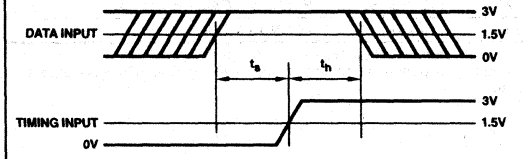
ENABLE AND DISABLE TIMES



WF008440

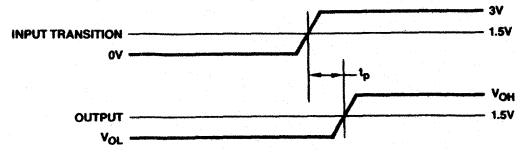
Notes: 1. Diagram shown for input control Enable-Low and input control Disable-High.
2. S_1 and S_2 of load circuit are closed except where shown.

SET-UP AND HOLD TIME MEASUREMENTS



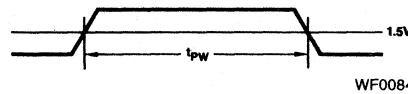
WF008450

PROPAGATION DELAY MEASUREMENTS



WF008460

PULSE WIDTH MEASUREMENTS



WF008470

Am9581 DDS READ/WRITE SECTION TIMING

No.	Parameters	Description	Min	Typ	Max	Units
READ						
1	t_{RRH}	RD/REFCLK HIGH Time*		0.5T		ns
2	t_{RRL}	RD/REFCLK LOW Time*		0.5T		ns
3	t_{RRC}	RD/REFCLK Cycle Time*		T		ns
4	t_{RGDW}	RG ↓ to RG ↑ DWELL Time*		2T		ns
5	t_{RRAMF}	RD/REFCLK ↑ to AMF ↑ Delay		10		ns
6	$t_{AMCAMFL}$	AMC ↓ to AMF ↓ Delay		25		ns
7	t_{RRRD}	RD/REFCLK ↑ to Read Data Delay		15		ns
8	t_{FAMSU}	Valid FAM to AMC ↑ Set-up		20		ns
9	t_{FAMH}	AMC ↑ to Valid FAM Hold		0		ns
10	t_{FDDAMD}	RD/REFCLK ↑ to Valid FDDAM Delay		15		ns
WRITE						
11	t_{WCH}	WTCLK HIGH Time*		0.5T		ns
12	t_{WCL}	WTCLK LOW Time*		0.5T		ns
13	t_{WCC}	WTCLK Cycle Time*		T		ns
14	t_{RGWGDW}	RG ↓ to WG ↑ DWELL*		2T		ns
15	t_{WGDW}	WG ↓ to WG ↑ DWELL*		2T		ns
16	t_{AMCWC}	AMC ↑ to WTCLK ↑ Set-up		20		ns
17	t_{AMCDW}	AMC ↓ to AMC ↑ DWELL*		2T		ns
18	t_{AMFWC}	WTCLK ↑ to AMF Delay		20		ns
19	$t_{AMCAMFL}$	AMC ↓ to AMF ↓ Delay		25		ns
20	t_{WDSU}	WTDATA to WTCLK ↑ Set-up		20		ns
21	t_{WDH}	WTCLK ↑ to WTDATA Hold		0		ns
22	t_{FAMSU}	Valid FAM to AMC ↑ Set-up		20		ns
23	t_{FAMH}	AMC ↑ to Valid FAM Hold		0		ns
24	t_{X1D}	X_1 to RD/REF CLK Delay			TBD	ns
25	t_{WGHWC}	WG ↑ to WTCLK ↑ Set-up		20		ns
26	t_{WGLWC}	WG ↓ to WTCLK ↑ Hold		20		ns
27	t_{PCWCSU}	PCEN to WTCLK ↑ Set-up		−1T		ns
28**	t_{PCWCH}	PCEN to WTCLK ↑ Hold		6T		ns

* T = 1/Nominal Data Rate.
** If the data pattern remaining in the write pipeline after WG goes LOW does not need pre-compensation, PCEN can go LOW at the same time as WG.

Am9581 DDS SELECT TIMING

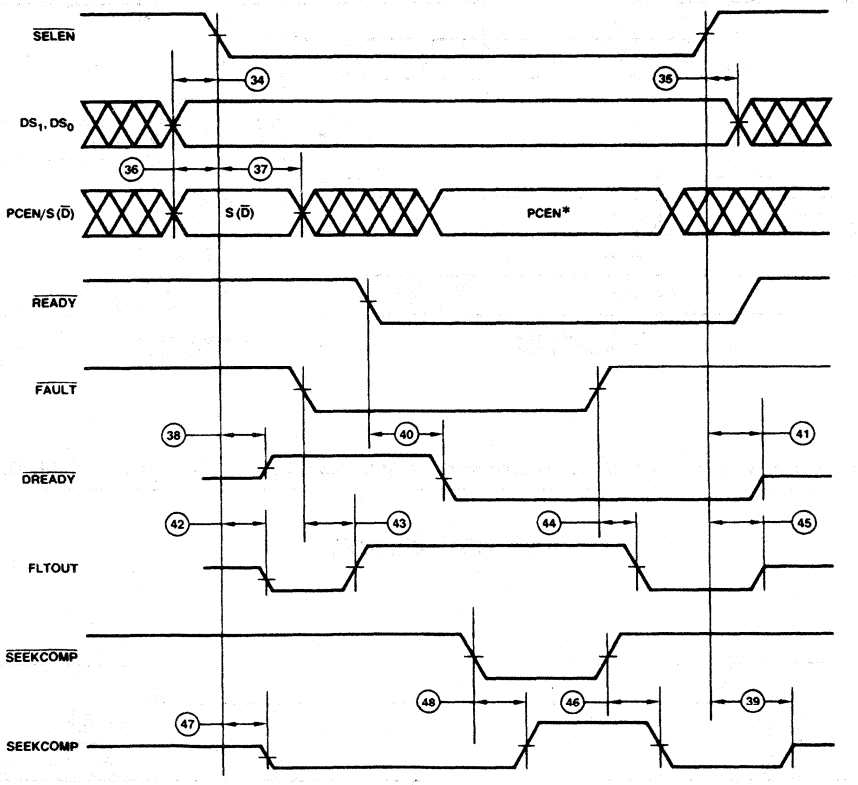
No.	Parameters	Description	Min	Typ	Max	Units
30	t_{SELSU}	$\overline{SEL}$ to SELEN ↓ Set-up		0		ns
31	t_{SEHH}	$\overline{SEL}$ ↑ to $\overline{SELEN}$ ↑ Hold		0		ns
32	t_{SELD}	$\overline{SELEN}$ ↓ to $\overline{SEL}$ ↓ Delay		50		ns
33	t_{SEHD}	$\overline{SELEN}$ ↑ to $\overline{SEL}$ ↑ Delay		50		ns

05309A

Refer to page 7-1 for Essential Information on Military Devices

Am9581 DDS CONTROL SECTION TIMING

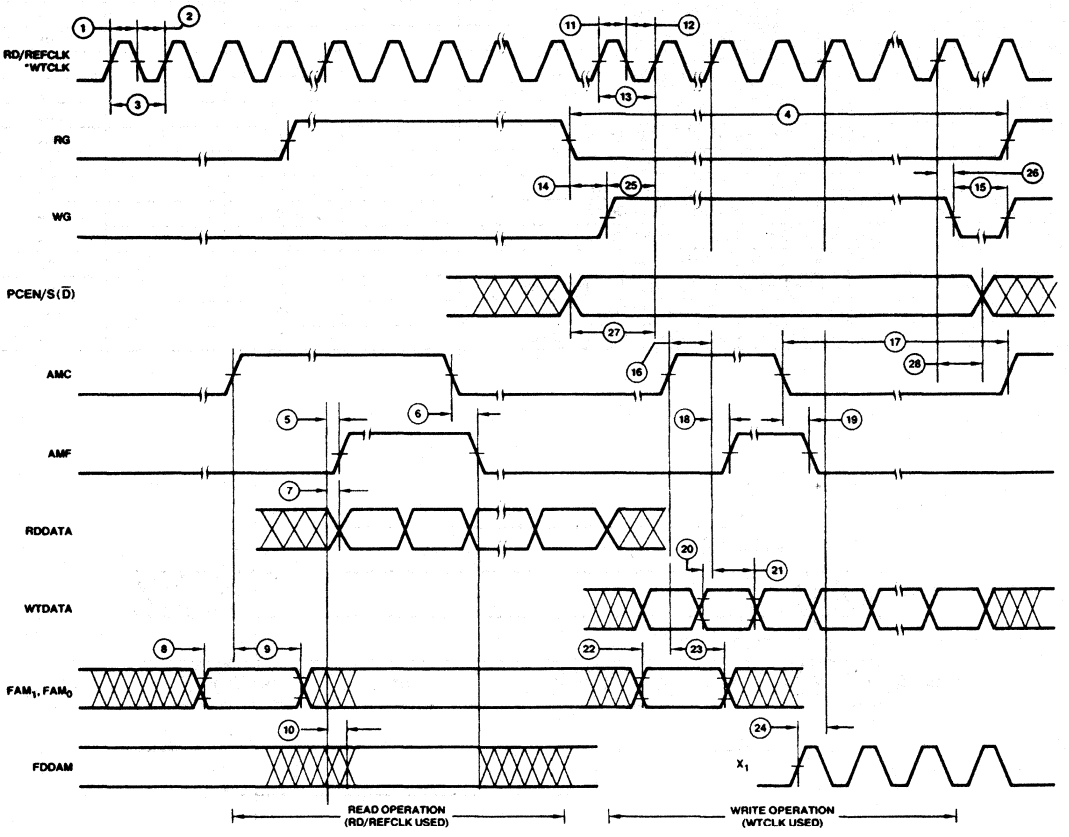
No.	Parameters	Description	Min	Typ	Max	Units
34	t_{DSSU}	DS Valid to $\overline{SELEN}$ LOW Set-up		20		ns
35	t_{DSH}	$\overline{SELEN}$ HIGH to DS Valid Hold		5		ns
36	$t_{S/DSU}$	S/$\overline{D}$ to $\overline{SELEN}$ LOW Set-up		20		ns
37	$t_{S/DH}$	$\overline{SELEN}$ LOW to S/$\overline{D}$ Hold		20		ns
38	t_{SEDR}	$\overline{SELEN}$ LOW to $\overline{DREADY}$ HIGH		50		ns
39	t_{SESCZ}	$\overline{SELEN}$ HIGH to SEEKCOMP High-Z		50		ns
40	t_{DRD}	$\overline{READY}$ LOW to $\overline{DREADY}$ LOW Delay		15		ns
41	t_{SEDRZ}	$\overline{SELEN}$ HIGH to $\overline{DREADY}$ High-Z		50		ns
42	t_{SEFL}	$\overline{SELEN}$ LOW to FLTOUT LOW Delay		50		ns
43	t_{FLFHD}	$\overline{FAULT}$ LOW to FLTOUT HIGH Delay		10		ns
44	t_{FFD}	$\overline{FAULT}$ HIGH to FLTOUT LOW Delay		10		ns
45	t_{SEFZ}	$\overline{SELEN}$ HIGH to FLTOUT High-Z		50		ns
46	t_{SCLD}	$\overline{SEEKCOMP}$ HIGH to SEEKCOMP LOW Delay		10		ns
47	t_{SCD}	$\overline{SELEN}$ LOW to SEEKCOMP LOW		50		ns
48	t_{SCHD}	$\overline{SEEKCOMP}$ LOW to SEEKCOMP HIGH Delay		10		ns



WF008432

*See WRITE section for PCEN timing requirements.

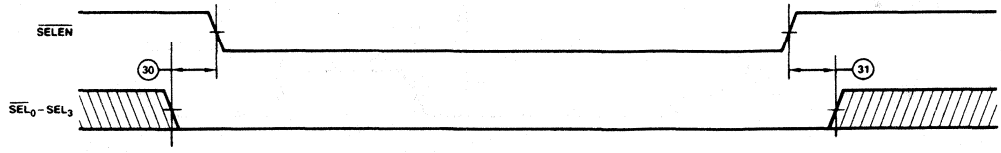
Am9581 DDS READ/WRITE SECTION TIMING



WF008502

*Note: During write operation, WTCLK is used which could be asynchronous relative to the RD/REFCLK.

DDS NOT SELECTED



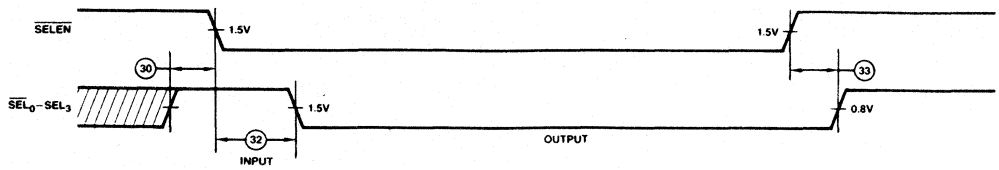
WF008510

Here a $\overline{SEL}$ line is used as an input and is held LOW. Whenever the HDC addresses the drive corresponding to this $\overline{SEL}$ pin, the DDS will tri-state all its output lines on the HDC interface.

05309A

Refer to page 7-1 for Essential Information on Military Devices

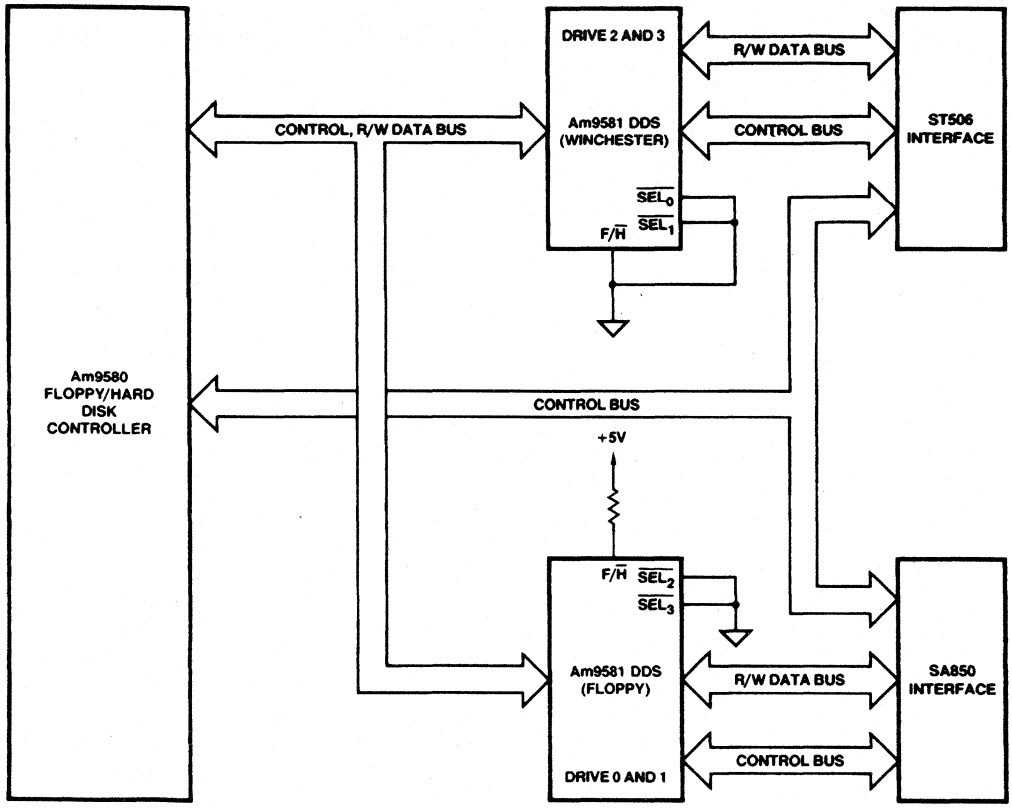
DDS SELECTED



WF008520

Here the DDS causes the $\overline{SEL}$ output corresponding to the drive selected to go LOW.

TYPICAL FLOPPY/WINCHESTER DISK DRIVE SYSTEM



AF003280

iAPX86 Family Index

80186

High Integration 16-Bit Microprocessor
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

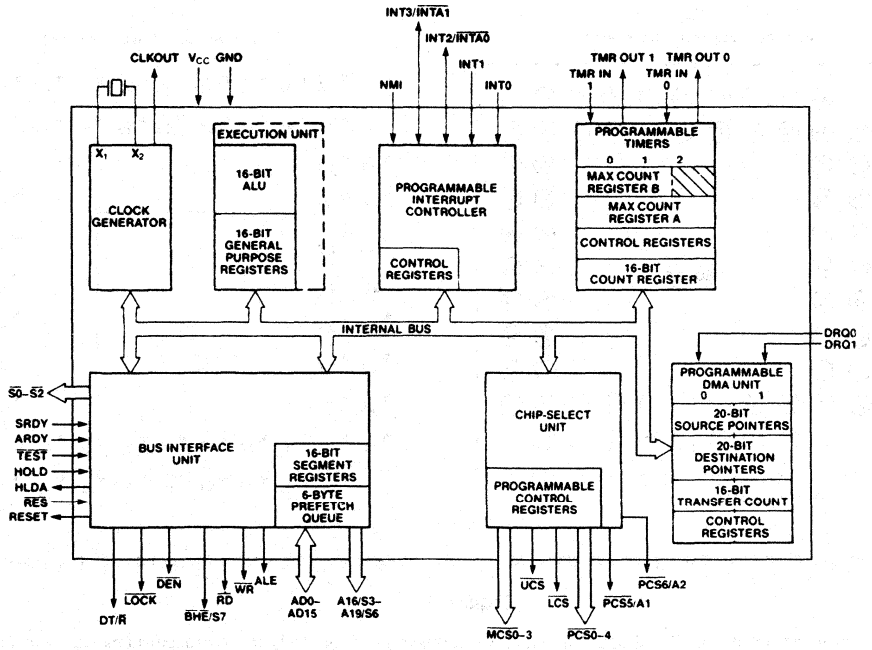
- Integrated feature set
  - Enhanced 10MHz 8086-1 CPU
  - Clock generator
  - Two independent, high-speed DMA channels
  - Programmable interrupt controller
  - Three programmable 16-bit timers
  - Programmable memory and peripheral chip-select logic
  - Programmable wait state generator
  - Local bus controller
- Available in 10MHz (80186-1), 8MHz (80186-3), and 6 MHz (80186-6).
- High performance processor
  - Two times the performance of the standard 8086
  - 4M byte/sec bus bandwidth interface
- Direct addressing capability to 1M byte of memory
- Completely object code compatible with all existing iAPX 86, 88 software
  - Ten new instruction types
  - Compatible with 29843/45, 29833/63, 8284, and 8288 bus support components
- Optional numeric processor extension
- Available in 68-pin Plastic Leaded Chip Carrier (PLCC), Ceramic Leadless Chip Carrier (LCC), and Pin Grid Array (PGA) packages.

GENERAL DESCRIPTION

The 80186 is a highly integrated 16-bit microprocessor. It effectively combines 15-20 of the most common iAPX 86 system components onto one. The 80186 provides two times greater throughput than the standard 5MHz 8086. The 80186 is upward compatible with 8086 and 8088 software and adds 10 new instruction types to the existing set.

The 80186 comes in a 68-pin package and requires a single ±5V power supply.

BLOCK DIAGRAM



BD003560

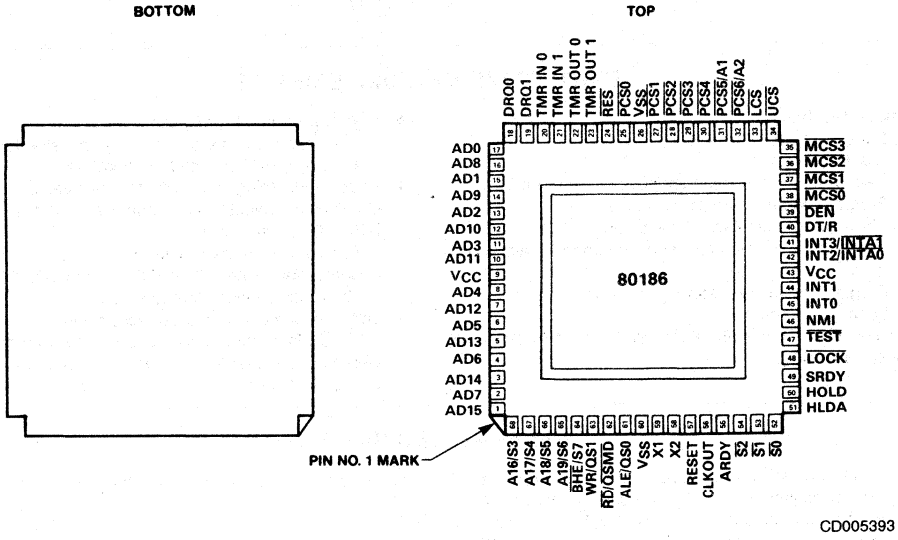
Figure 1.

03551C

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM

68 Pin Ceramic LCC Package

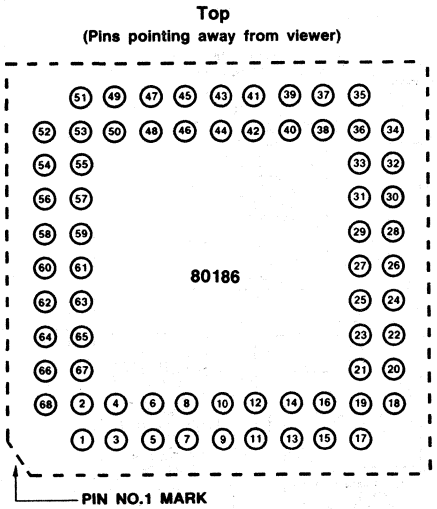


CD005393

The Plastic LCC package has the same connection diagram as the Ceramic LCC package.

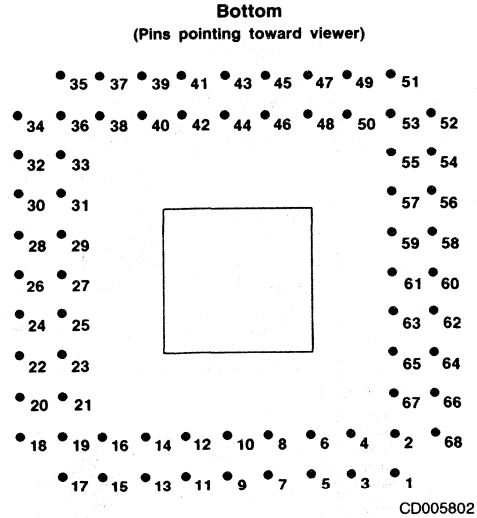
68 Pin Grid Array

Cavity Down Package



CD005792

Pins are not visible from the top of this package.



CD005802

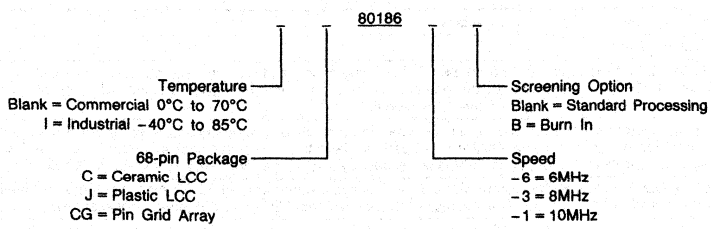
Pins are visible from the bottom of this package.

Figure 2.

Also available in 68 PLCC. Package pin numbers correspond sequentially to device pin numbers. The 80186 is molded upside down and rotated ninety degrees counterclockwise to match the LCC pinout.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
80186-6 80186-3 80186-6B 80186-3B	J, C, CG, IC, IG
80186-1 80186-1B	J, C
MR80186-6 MR80186-3	/C
MG80186-6 MG80186-3	/C

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
9, 43	V_{CC}, V_{CC}	I	System Power: +5 volt power supply.
26, 60	V_{SS}, V_{SS}	I	System Ground.
57	RESET	O	Reset Output indicates that the 80186 CPU is being reset; and can be used as a system reset. It is active HIGH, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the $\overline{RES}$ signal.
59, 58	X1, X2	I	Crystal inputs, X1 and X2, provide an external connection for a fundamental mode parallel resonant crystal for the internal crystal oscillator. X1 can interface to an external clock instead of a crystal. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT).
56	CLKOUT	O	Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT has sufficient MOS drive capabilities for a numeric processor extension.
24	$\overline{RES}$	I	System Reset causes the 80186 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80186 clock. The 80186 begins fetching instructions approximately 7 clock cycles after $\overline{RES}$ is returned HIGH. $\overline{RES}$ is required to be LOW for greater than 4 clock cycles and is internally synchronized. For proper initialization, the LOW-to-HIGH transition of $\overline{RES}$ must occur no sooner than 50 microseconds after power up. This input is provided with a Schmitt-trigger to facilitate power-on $\overline{RES}$ generation via an RC network. When $\overline{RES}$ occurs, the 80186 will drive the status lines to an inactive level for one clock, and then tri-state them.
47	$\overline{TEST}$	I	$\overline{TEST}$ is examined by the WAIT instruction. If the $\overline{TEST}$ input is HIGH when "WAIT" execution begins, instruction execution will suspend. $\overline{TEST}$ will be resampled until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80186 is waiting for $\overline{TEST}$, interrupts will be serviced. This input is synchronized internally.
20 21	TMR in 0, TMR IN1	I I	Timer inputs are used either as clock or control signals, depending upon the programmed timer mode. These inputs are active HIGH (or LOW-to-HIGH transitions are counted) and internally synchronized.
22 23	TMR OUT 0, TMR OUT 1	O O	Timer outputs are used to provide single pulse or continuous waveform generation, depending upon the timer mode selected.
18 19	DRQ0 DRQ1	I I	DMA Request is driven HIGH by an external device when it desires that a DMA channel (Channel 0 or 1) perform a transfer. These signals are active HIGH, level-triggered, and internally synchronized.
46	NMI	I	Non-Maskable Interrupt is an edge-triggered input which causes a type 2 interrupt. NMI is not maskable internally. A transition from a LOW to HIGH initiates the interrupt at the next instruction boundary. NMI is latched internally. An NMI duration of one clock or more will guarantee service. This input is internally synchronized.
45, 44 42 41	INT0, INT1 INT2/$\overline{INTA0}$ INT3/$\overline{INTA1}$	I I/O I/O	Maskable Interrupt Requests can be requested by strobing one of these pins. When configured as inputs, these pins are active HIGH. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured via software to provide active-LOW interrupt-acknowledge output signals. All interrupt inputs may be configured via software to be either edge- or level-triggereed. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowledged. When iRMX mode is selected, the function of these pins changes (see Interrupt Controller section of this data sheet).
65-68	A19/S6, A18/S5, A17/S4, A16/S3	O O O O	Address Bus Outputs (16-19) and Bus Cycle Status (3-6) reflect the four most significant address bits during T_1. These signals are active HIGH. During T_2, T_3, T_W, and T_4, status information is available on these lines as encoded below: S3, S4, and S5 are defined as LOW during T_2-T_4.
10-17 1-8	AD15-AD0	I/O	Address/Data Bus (0-15) signals constitute the time multiplexed memory or I/O address (T_1) and data (T_2, T_3, T_W, and T_4) bus. The bus is active HIGH. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7 through D_0. It is LOW during T_1 when a byte is to be transferred onto the lower portion of the bus in memory or I/O operations.
64	$\overline{BHE}$/S7	O	During T_1 the Bus High Enable signal should be used to determine if data is to be enabled onto the most significant half of the data bus, pins D_{15}-D_8. $\overline{BHE}$ is LOW during T_1 for read, write, an interrupt acknowledge cycles when a byte is to be transferred on the higher half of the bus. The S_7 status information is available during T_2, T_3, and T_4. S_7 is logically equivalent to $\overline{BHE}$. The signal is active LOW, and is tristated OFF during bus HOLD.
61	ALE/QS0	O	Address Latch Enable/Queue Status 0 is provided by the 80186 to latch the address into the 8282/8283 address latches. ALE is active HIGH. Addresses are guaranteed to be valid on the trailing edge of ALE. The ALE rising edge is generated off the rising edge of the CLKOUT immediately preceding T_1 of the associated bus cycle, effectively one-half clock cycle earlier than in the standard 8086. The trailing edge is generated off the CLKOUT rising edge in T_1 as in the 8086. Note that ALE is never floated.

	Low	High
S6	Processor Cycle	DMA Cycle

$\overline{BHE}$ and A0 Encodings

$\overline{BHE}$ Value	A0 Value	Function
0	0	Work Transfer
0	1	Byte Transfer on upper half of data bus (D15-D8)
1	0	Byte Transfer on lower half of data bus (D_7-D_0)
1	1	Reserved

03551C

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
63	$\overline{WR}$/QS1	O	Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. $\overline{WR}$ is active for T_2, T_3, and T_W of any write cycle. It is active LOW, and floats during "HOLD." It is driven HIGH for one clock during Reset, and then floated. When the 80186 is in queue status mode, the ALE/QS0 and $\overline{WR}$/QS1 pins provide information about processor/instruction queue interaction.
62	$\overline{RD}$/$\overline{QSMD}$	O	Read Strobe indicates that the 80186 is performing a memory or I/O read cycle. $\overline{RD}$ is active LOW for T_2, T_3, and T_W of any read cycle. It is guaranteed not to go LOW in T_2 until after the Address Bus is floated. $\overline{RD}$ is active LOW, and floats during "HOLD." $\overline{RD}$ is driven HIGH for one clock during Reset, and then the output driver is floated. A weak internal pull-up mechanism on the $\overline{RD}$ line holds it HIGH when the line is not driven. During RESET the pin is sampled to determine whether the 80186 should provide ALE, $\overline{WR}$, and $\overline{RD}$, or if the Queue-Status should be provided. $\overline{RD}$ should be connected to GND to provide Queue-Status data.
55	ARDY	I	Asynchronous Ready informs the 80186 that the addressed memory space or I/O device will complete a data transfer. The ARDY input pin will accept an asynchronous input, and is active HIGH. Only the rising edge is internally synchronized by the 80186. This means that the falling edge of ARDY must be synchronized to the 80186 clock. If connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active to terminate a bus cycle.
49	SRDY	I	Synchronous Ready must be synchronized externally to the 80186. The use of SRDY provides a relaxed system-timing specification on the Ready input. This is accomplished by eliminating the one-half clock cycle which is required for internally resolving the signal level when using the ARDY input. This line is active HIGH. If this line is connected to V_{CC} no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active before a bus cycle is terminated.
48	$\overline{LOCK}$	O	$\overline{LOCK}$ output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction following the LOCK prefix. It remains active until the completion of the instruction following the LOCK prefix. No prefetches will occur while $\overline{LOCK}$ is asserted. $\overline{LOCK}$ is active LOW, is driven HIGH for one clock during RESET, and then floated.
52-54	$\overline{S0}$, $\overline{S1}$, $\overline{S2}$	O	Bus cycle status $\overline{S0}$-$\overline{S2}$ are encoded to provide bus-transaction information:
50 51	HOLD (input) HLDA (output)	I O	HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. HOLD may be asynchronous with respect to the 80186 clock. The 80186 will issue a HLDA in response to a HOLD request at the end of T_4 or T_1 Simultaneous with the issuance of HLDA, the 80186 will float the local bus and control lines. After HOLD is detected as being LOW, the 80186 will lower HLDA. When the 80186 needs to run another bus cycle, it will again drive the local bus and control lines.
34	$\overline{UCS}$	O	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1K-=256K block) of memory. This line is not floated during bus HOLD. The address range activating $\overline{UCS}$ is software programmable.
33	$\overline{LCS}$	O	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K-256K) of memory. This line is not floated during bus HOLD. The address range activating $\overline{LCS}$ is software programmable.
38, 37, 36, 35	$\overline{MCS0}$-3	O	Mid-Range Memory Chip Select signals are active LOW when a memory reference is made to the defined mid-range portion of memory (8K-512K). These lines are not floated during bus HOLD. The address ranges activating $\overline{MCS0}$-3 are software programmable.
25, 27-30	$\overline{PCS0}$-4	O	Peripheral Chip Select signals 0-4 are active LOW when a reference is made to the defined peripheral area (65K byte I/O space). These lines are not floated during bus HOLD. The address ranges activating $\overline{PCS0}$-4 are software programmable.

QS1	QS0	Queue Operation
0	0	No queue operation
0	1	First opcode byte fetched from the queue
1	1	Subsequent byte fetched from the queue
1	0	Empty the queue

80186 Bus Cycle Status Information			
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Bus Cycle Initiated
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

The status pins float during "HOLD."
$\overline{S2}$ may be used as a logical M/$\overline{IO}$ indicator, and $\overline{S1}$ as a DT/$\overline{R}$ indicator.
The status lines are driven HIGH for one clock during Reset, and then floated until a bus cycle begins.

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
31	$\overline{PCS5}$/A1	O	Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating $\overline{PCS5}$ is software programmable. When programmed to provide latched A1, rather than $\overline{PCS5}$, this pin will retain the previously latched value of A1 during a bus HOLD. A1 is active HIGH.
32	$\overline{PCS6}$/A2	O	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating $\overline{PCS6}$ is software programmable. When programmed to provide latched A2, rather than $\overline{PCS6}$, this pin will retain the previously latched value of A2 during a bus HOLD. A2 is active HIGH.
40	DT/$\overline{R}$	O	Data Transmit/Receive controls the direction of data flow through the external 29833/29863 data bus transceiver. When LOW, data is transferred to the 80186. When HIGH the 80186 places write data on the data bus.
39	$\overline{DEN}$	O	Data Enable is provided as a 29833/29863 data bus transceiver output enable. $\overline{DEN}$ is active LOW during each memory and I/O access. $\overline{DEN}$ is HIGH whenever DT/$\overline{R}$ changes state.

DETAILED DESCRIPTION

Introduction

The following Functional Description describes the base architecture of the 80186. The architecture is common to the 8086, 8088, and 80286 microprocessor families as well. The 80186 is a very high integration 16-bit microprocessor. It combines 15-20 of the most common microprocessor system components onto one chip while providing twice the performance of the standard 8086. The 80186 is object code compatible with the 8086, 8088 microprocessors and adds 10 new instruction types to the existing 8086, 8088 instruction set.

80186 BASE ARCHITECTURE

The 8086, 8088, 80186, and 80286 family all contain the same basic set of registers, instructions, and addressing modes. The 80186 processor is upward compatible with the 8086, 8088, and 80286 CPUs.

Register Set

The 80186 base architecture has fourteen registers as shown in Figures 3a and 3b. These registers are grouped into the following categories.

General Registers

Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers.

Segment Registers

Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers

Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.

Status and Control Registers

Two 16-bit special purpose registers record or alter certain aspects of the 80186 processor state. These are the Instruction Pointer Register, which contains the offset address of the next sequential instruction to be executed, and the Status Word Register, which contains status and control flag bits (see Figures 3a and 3b).

Status Word Description

The Status Word records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80186 within a given operating mode (bits 8, 9, and 10). The Status Word Register is 16-bits wide. The function of the Status Word bits is shown in Table 2.

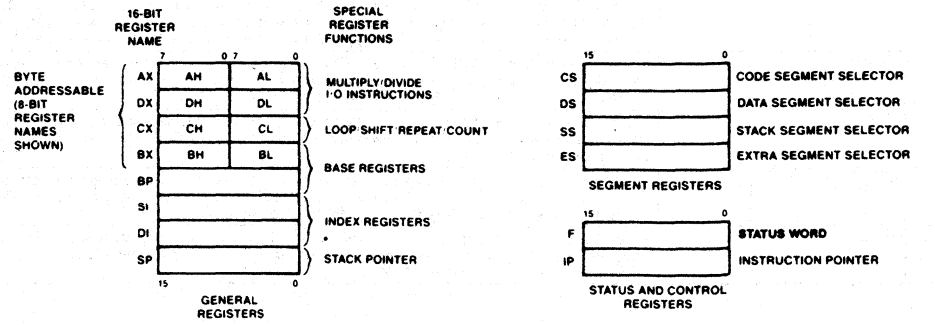


Figure 3a. 80186 General Purpose Register Set

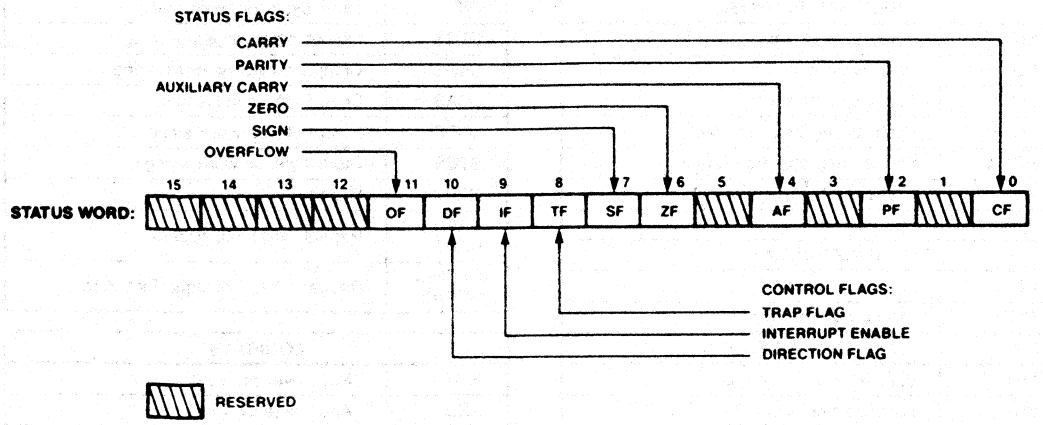


Figure 3b. Status Word Format

Table 2. Status Word Bit Function

Bit Position	Name	Function
0	CF	Carry Flag — Set on high-order bit carry or borrow; cleared otherwise.
2	PF	Parity Flag — Set if low-order 8 bits or result contain an even number of 1-bits; cleared otherwise.
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise.
6	ZF	Zero Flag — Set if result is zero; cleared otherwise.
7	SF	Sign Flag — Set equal to high-order bit of result (0 if positive, 1 if negative).
8	TF	Single Step Flag — Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-enable Flag — When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag — Causes string instructions to auto decrement the appropriate index register when set. Clearing DF causes auto increment.
11	OF	Overflow Flag — Set if the signed result cannot be expressed within the number of bits in the destination operand; cleared otherwise.

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high-level instructions, and processor control. These categories are summarized in Figure 4.

An 80186 instruction can reference anywhere from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this data sheet.

Memory Organization

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit base segment and a 16-bit offset. The 16-bit base values are contained i one of four internal segment registers (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 5). This allows for a 1 MByte physical address size.

All instructions that address operands in memory must specifiy the base segment and the 16-bit offset value. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 3). These rules follow the way programs are written (see Figure 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs.

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword
MOVS	Move byte or word string

INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/ REPZ	Repeat while equal/zero
REPNE/ REPNZ	Repeat while not equal/not zero

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{TEST}}$ pin active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
HIGH LEVEL INSTRUCTIONS	
ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 4. 80186 Instruction Set

CONDITIONAL TRANSFERS		UNCONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**ITERATION CONTROLS**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**INTERRUPTS**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 4. 80186 Instruction Set (continued)

To access operands that do not reside in one of the four immediately available segments, a full 32-bit pointer can be used to reload both the base (segment) and offset values.

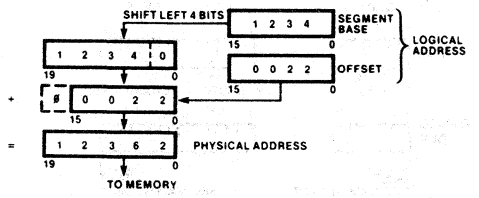


Figure 5. Two Component Address

Table 3. Segment Register Selection Rule

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Instruction prefetch and immediate data.
Stack	Stack (SS)	All stack pushes and pops; any memory references which use BP Register as a base register.
External Data (Global)	Extra (ES)	All string instruction references which use the DI register as an index.
Local Data	Data (DS)	All other data references.

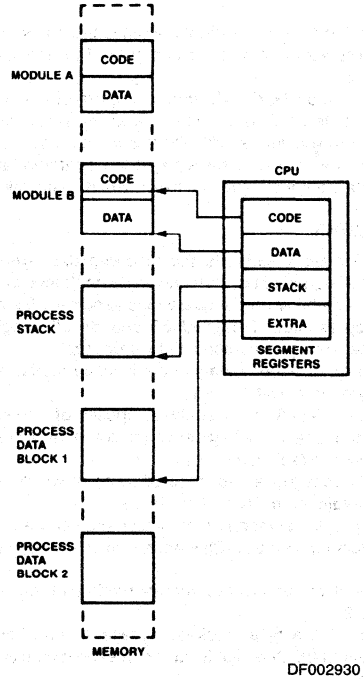


Figure 6. Segmented Memory Helps Structure Software

Addressing Modes

The 80186 provides eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

- *Register Operand Mode:* The operand is located in one of the 8- or 16-bit general registers.
- *Immediate Operand Mode:* The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: a segment base and an offset. The segment base is supplied by a 16-bit segment register either implicity chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

- the *displacement* (an 8- or 16-bit immediate value contained in the instruction);
- the *base* (contents of either the BX or BP base registers); and
- the *index* (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, described below.

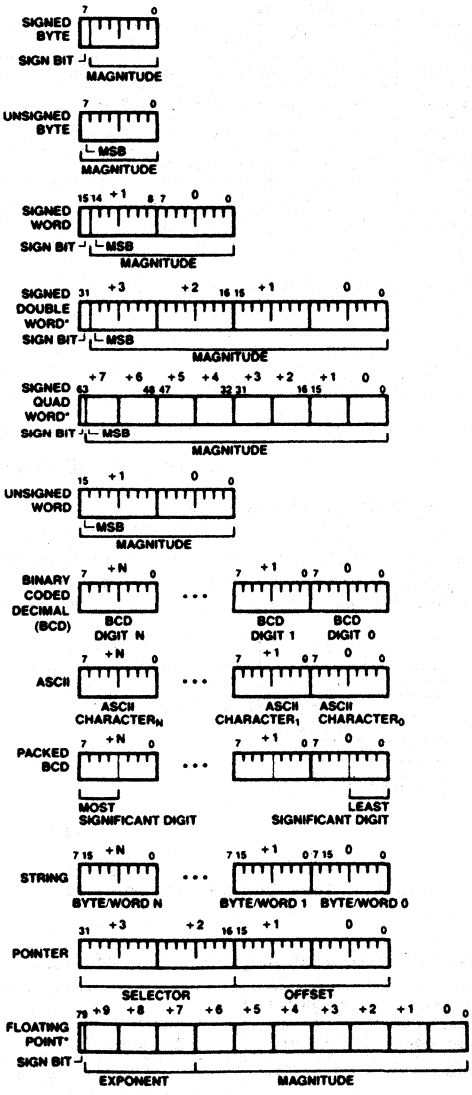
- *Direct Mode:* The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.
- *Register Indirect Mode:*The operand's offset is in one of the registers SI, DI, BX, or BP.
- *Based Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
- *Indexed Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).
- *Based Indexed Mode:* The operand's offset is the sum of the contents of a base register and an index register.
- *Based Indexed Mode with Displacement:* The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80186 directly supports the following data types:

- *Integer:* A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32 and 64 bit integers are supported using a numeric data processor.
- *Ordinal:* An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- *Pointer:* A 16- or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component in addition to a 16-bit offset component.
- *String:* A contiguous sequence of bytes or words. A string may contain from 1K to 64K bytes.
- *ASCII:* A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- *BCD:* A byte (unpacked) representation of the decimal digits 0-9.
- *Packed BCD:* A byte (packed) representation of two decimal digits (0-9). One digit is stored in each nibble (4-bits) of the byte.
- *Floating Point:* A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using a numeric data processor configuration.)

In general, individual data elements must fit within defined segment limits. Figure 7 graphically represents the data types supported by the 80186.



NOTE: *SUPPORTED BY 80186 WITH A NUMERIC DATA PROCESSOR

Figure 7. 80186 Supported Data Types

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. Separate instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register, 8-bit port addresses are zero extended such that A_{15}-A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Status Word) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable.

Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. If the exception was caused by executing an ESC instruction with the ESC trap bit set in the relocation register, the return instruction will point to the ESC instruction, or to the segment override prefix immediately preceding the ESC instruction if the prefix was present. In all other cases, the return address from an exception will point at the instruction immediately following the instruction causing the exception.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0-31, some of which are used for instruction exceptions, are reserved. Table 4 shows the 80186 predefined types and default priority levels. For each interrupt, an 8-bit vector must be supplied to the 80186 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. In addition, internal peripherals and noncascaded external interrupts will generate their own vectors through the internal interrupt controller. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Interrupt Sources

The 80186 can service interrupts generated by software or hardware. The software interrupts are generated by specific instructions (INT, ESC, unused OP, etc.) or the results of conditions specified by instructions (array bounds check, INTO, DIV, IDIV, etc.) All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type (Table 4), multiplied by four. All hardware-generated interrupts are sampled at the end of each instruction. Thus, the software interrupts will begin service first. Once the service routine is entered and interupts are enabled, any hardware source of sufficient priority can interrupt the service routine in progress.

The software generated 80186 interrupts are described below.

DIVIDE ERROR EXCEPTION (TYPE 0)

Generated when a DIV or IDIV instuction quotient cannot be expressed in the number of bits in the destination.

Table 4. 80186 Interrupt Vectors

Interrupt Name	Vector Type	Default Priority	Related Instructions
Divide Error Exception	0	*1	DIV, IDIV
Single Step Interrupt	1	12**2	All
NMI	2	1	All
Breakpoint Interrupt	3	*1	INT
INT0 Detected Overflow Exception	4	*1	INT0
Array Bounds Exception	5	*1	BOUND
Unused-Opcode Exception	6	*1	Undefined Opcodes
ESC Opcode Exception	7	*1***	ESC Opcodes
Timer 0 Interrupt	8	2A****	
Timer 1 Interrupt	16	2B****	
Timer 2 Interrupt	17	2C****	
Reserved	9	3	
DMA 0 Interrupt	10	4	
DMA 1 Interrupt	11	5	
INT0 Interrupt	12	6	
INT1 Interrupt	13	7	
INT2 Interrupt	14	8	
INT3 Interrupt	15	9	

NOTES:

*1. These are generated as the result of an instruction execution.

**2. This is handled as in the 8086.

****3. All three timers constitute one source of request to the interrupt controller. The Timer interrupts all have the same default priority level with respect to all other interrupt sources. However, they have a defined priority ordering amongst themselves. (Priority 2A is higher priority than 2B.) Each Timer interrupt has a separate vector type number.

4. Default priorities for the interrupt sources are used only if the user does not program each source into a unique priority level.

***5. An escape opcode will cause a trap only if the proper bit is set in the peripheral control block relocation register.

SINGLE-STEP INTERRUPT (TYPE 1)

Generated after most instructions if the TF flag is set. Interrupts will not be generated after prefix instructions (e.g., REP), instructions which modify segment registers (e.g., POP DS), or the WAIT instruction.

NON-MASKABLE INTERRUPT-NMI (TYPE 2)

An external interrupt source which cannot be masked.

BREAKPOINT INTERRUPT (TYPE 3)

A one-byte version of the INT instruction. It uses 12 as an index into the service routine address table (because it is a type 3 interrupt).

INTO DETECTED OVERFLOW EXCEPTION (TYPE 4)

Generated during an INTO instruction if the 0F bit is set.

ARRAY BOUNDS EXCEPTION (TYPE 5)

Generated during a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

UNUSED OPCODE EXCEPTION (TYPE 6)

Generated if execution is attempted on undefined opcodes.

ESCAPE OPCODE EXCEPTION (TYPE 7)

Generated if execution is attempted of ESC opcodes (D8H-DFH). This exception will only be generated if a bit in the relocation register is set. The return address of this exception will point to the ESC instruction causing the exception. If a segment override prefix preceded the ESC instruction, the return address will point to the segment override prefix.

Hardware-generated interrupts are divided into two groups: maskable interupts and non-maskable interrupts. The 80186 provides maskable hardware interrupt request pins INTO-INT3. In addition, maskable interrupts may be generated by the 80186 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 4. Software enables these inputs by setting the interrupt flag bit (IF) in the Status Word. The interrupt controller is discussed in the peripheral section of this data sheet.

Further maskable interrupts are disabled while servicing an interrupt because the IF bit is reset as part of the response to an interrupt or exception. The saved Status Word will reflect the enable status of the processor prior to the interrupt. The interrupt flag will remain zero unless specifically set. The interrupt return instruction restores the Status Word, thereby restoring the original status of IF bit. If the interrupt return re-enables interrupts, and another interrupt is pending, the 80186 will immediately service the highest-priority interrupt pending, i.e., no instructions of the main line program will be executed.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt (NMI) is also provided. This interrupt is serviced regardless of the state of the IF bit. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of an NMI interrupt to prevent maskable interrupts from being serviced.

Single-Step Interrupt

The 80186 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single-step interrupt and is controlled by the single-step flag bit (TF) in the Status Word. Once this bit is set, an internal single-step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single-stepped.

Initialization and Processor Reset

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin LOW. $\overline{RES}$ forces the 80186 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{RES}$ is active. After $\overline{RES}$ becomes inactive and an internal processing interval elapses, the 80186 begins execution with the instruction at physical location FFFF0(H). $\overline{RES}$ also sets some registers to predefined values as shown in Table 5.

Table 5. 80186 Initial Register State after RESET

Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

80186 CLOCK GENERATOR

The 80186 provides an on-chip clock generator for both internal and external clock generation. The clock generator features a crystal oscillator, a divide-by-two counter, synchronous and asynchronous ready inputs, and reset circuitry.

Oscillator

The oscillator circuit of the 80186 is designed to be used with a parallel resonant fundamental mode crystal. This is used as the time base for the 80186. The crystal frequency selected will be double the CPU clock frequency. Use of an LC or RC circuit is not recommended with this oscillator. If an external oscialltor is used, it can be connected directly to input pin X1 in lieu of a crystal. The output of the oscillator is not directly available outside the 80186. The recommended crystal configuration is shown in Figure 8.

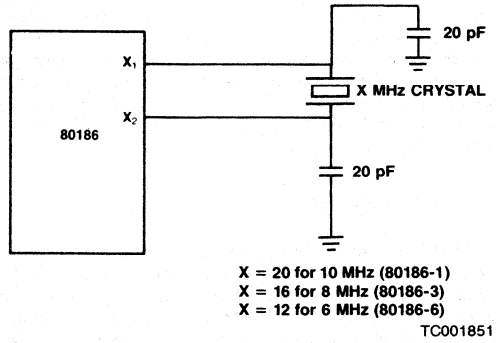


Figure 8. Recommended 80186 Crystal Configuration

Clock Generator

The 80186 clock generator provides the 50% duty cycle processor clock for the 80186. It does this by dividing the oscillator output by 2 forming the symmetrical clock. If an external oscillator is used, the state of the clock generator will change on the falling edge of the oscillator signal. The CLKOUT pin provides the processor clock signal for use outside the 80186. This may be used to drive other system components. All timings are referenced to the output clock.

READY Synchronization

The 80186 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of T_2, T_3 and again in the middle of each T_W until ARDY is sampled HIGH. One-half CLKOUT cycle of resolution time is used. Full synchronization is performed only on the rising edge of ARDY, i.e., the falling edge of ARDY must be synchronized to the CLKOUT signal if it will occur during T_2 or T_W. HIGH-to-LOW transitions of ARDY must be performed synchronously to the CPU clock.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of T_2 and again at the end of each T_W until it is sampled HIGH. By using this input rather than the asynchronous ready input, the half-clock cycle resolution time penalty is eliminated.

This input must satisfy set-up and hold times to guarantee proper operation of the circuit.

In addition, the 80186, as part of the integrated chip-select logic, has the capability to program WAIT states for memory and peripheral blocks. This is discussed in the Chip Select/ Ready Logic description.

RESET Logic

The 80186 provides both a $\overline{RES}$ input pin and a synchronized RESET pin for use with other system components. The $\overline{RES}$ input pin on the 80186 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET is guaranteed to remain active for at least five clocks given a $\overline{RES}$ input of at least six clocks. RESET may be delayed up to two and one-half clocks behind $\overline{RES}$.

Multiple 80186 processors may be synchronized through the $\overline{RES}$ input pin, since this input resets both the processor and divide-by-two internal counter in the clock generator. In order to insure that the divide-by-two counters all begin counting at the same time, the active going edge of $\overline{RES}$ must satisfy a 25 ns setup time before the falling edge of the 80186 clock input. In addition, in order to insure that all CPUs begin executing in the same clock cycle, the reset must satisfy a 25 ns setup time before the rising edge of the CLKOUT signal of all the processors.

LOCAL BUS CONTROLLER

The 80186 provides a local bus controller to generate the local bus control signals. In addition, it employs a HOLD/HLDA protocol for relinquishing the local bus to other bus masters. It also provides control lines that can be used to enable external buffers and to direct the flow of data on and off the local bus.

Memory/Peripheral Control

The 80186 provides ALE, $\overline{RD}$, and $\overline{WR}$ bus control signals. The $\overline{RD}$ and $\overline{WR}$ signals are used to strobe data from memory to the 80186 or to strobe data from the 80186 to memory. The ALE line provides a strobe to address latches for the multiplexed address/data bus. The 80186 local bus controller does not provide a memory/$\overline{I/O}$ signal. If this is required, the user will have to use the $\overline{S2}$ signal (which will require external latching), make the memory and I/O spaces nonoverlapping, or use only the integrated chip-select circuitry.

Transceiver Control

The 80186 generates two control signals to be connected to 29833/29863 transceiver chips. This capability allows the addition of transceivers for extra buffering without adding external logic. These control lines, DT/$\overline{R}$ and $\overline{DEN}$, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 6.

Table 6. Transceiver Control Signals Description

Pin Name	Function
$\overline{DEN}$ (Data Enable)	Enables the output drivers of the transceivers. It is active LOW during memory, I/O, or INTA cycles.
DT/$\overline{R}$ (Data Transmit/ Receive)	Determines the direction of travel through the transceivers. A HIGH level directs data away from the processor during write operations, while a LOW level directs data toward the processor during a read operation.

Local Bus Arbitration

The 80186 uses a HOLD/HLDA system of local bus exchange. This provides an asynchronous bus exchange mechanism. This means multiple masters utilizing the same bus can operate at separate clock frequencies. The 80186 provides a single HOLD/HLDA pari through which all other bus masters may gain cotnrol of the local bus. This requires external circuitry to arbitrate whcih external device will gain control of the bus from the 80186 when there is more than one alternate local bus master. When the 80186 relinquishes control of the local bus, it floats $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S0}$-$\overline{S2}$, $\overline{LOCK}$, AD0-AD15, A16-A19, $\overline{BHE}$, and DT/$\overline{R}$ to allow another master to drive these lines directly.

The 80186 HOLD latency time, i.e., the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is the highest-priority activity request which the processor may receive: higher than instruction fetching or internal DMA cycles. However, if a DMA cycle is in progress, the 80186 will complete the transfer before relinquishing the bus. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency time can be as great as 4 bus cycles. This will occur if a DMA word transfer operation is taking place from an odd address to an odd address. This is a total of 16 clocks or more, if WAIT states are required. In addition, if locked transfers are performed, the HOLD latency time will be increased by the length of the locked transfer.

Local Bus Controller and Reset

Upon receipt of a RESET pulse from the $\overline{RES}$ input, the local bus controller will perform the following actions:

- Drive $\overline{DEN}$, $\overline{RD}$, and $\overline{WR}$ HIGH for one clock cycle, then float.

  NOTE: $\overline{RE}$ is also provided with an internal pull-up device to prevent the processor from inadvertently entering Queue Status mode during reset.
- Drive $\overline{S0}$-$\overline{S2}$ to the passive state (all HIGH) and then float.
- Drive $\overline{LOCK}$ HIGH and then float.
- Tristate AD0-15, A16-19, $\overline{BHE}$, DT/$\overline{R}$.
- Drive ALE LOW (ALE is never floated).
- Drive HLDA LOW.

INTERNAL PERIPHERAL INTERFACE

All the 80186 integrated peripherals are controlled via 16-bit registers contained within an internal 256-byte control block. This control block may be mapped into either memory or I/O space. Internal logic will recognize the address and respond to

the bus cycle. During bus cycles to internal registers, the bus controller will signal the operation externally (i.e., the $\overline{RD}$, $\overline{WR}$, status, address, data, etc., lines will be driven as in a normal bus cycle), but D_{15-0}, SRDY, and ARDY will be ignored. The base address of the control block must be on an even 256-byte boundary (i.e., the lower 8 bits of the base address are all zeros). All of the defined registers within this control block may be read or written by the 80186 CPU at any time. The location of any register contained within the 256-byte control block is determined by the current base address of the control block.

The control block base address is programmed via a 16-bit relocation register contained within the control block at offset FEH from the base address of the control block (see Figure 9). It provides the upper 12 bits of the base address of the control block. Note that mapping the control register block into an address range corresponding to a chip-select range is not recommended (the chip select circuitry is discussed later in this date sheet). In addition, bit 12 of this register determines whether the control block will be mapped into I/O or memory space. If this bit is 1, the control block will be located in memory space, whereas if the bit is 0, the control block will be located in I/O space. If the control register block is mapped into I/O space, the upper 4 bits of the base address must be programmed as 0 (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the relocation register contains bits which place the interrupt controller into iRMX mode, and cause the CPU to interrupt upon encountering ESC instructions. At RESET, the relocation register is set to 20FFH. This causes the control block to start at FF00H in I/O space. An offset map of the 256-byte control register block is shown in Figure 10.

The integrated 80186 peripherals operate semiautonomously from the CPU. Access to them for the most part is via software read/write of the control and data locations in the control block. Most of these registers can be both read and written. A few dedicated lines, such as interrupts and DMA request provide real-time communication between the CPU and peripherals as in a more conventional system utilizing discrete peripheral blocks. The overall interaction and function of the peripheral blocks has not substantially changed.

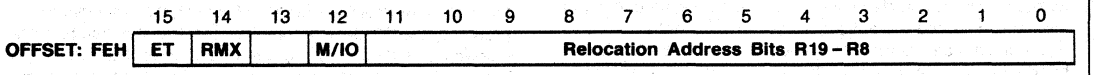
CHIP-SELECT/READY GENERATION LOGIC

The 80186 contains logic which provides programmable chip-select generation for both memories and peripherals. In addition, it can be programmed to provide READY (or WAIT state) generation. It can also provide latched address bits A1 and A2. The chip-select lines are active for all memory and I/O cycles in their programmed areas, whether they be generated by the CPU or by the integrated DMA unit.

Memory Chip Selects

The 80186 provides 6 memory chip select outputs for 3 address areas: upper memory, lower memory, and midrange memory. One each is provided for upper memory and lower memory, while four are provided for midrange memory.

The range for each chip select is user-programmable and can be set to 2K, 4K, 8K, 16K, 32K, 64K, 128K (plus 1K and 256K for upper and lower chip selects). In addition, the beginning or base address of the midrange memory chip select may also be selected. Only one chip select may be programmed to be active for any memory location at a time. All chip select sizes are in bytes, whereas 80186 memory is arranged in words. This means that if, for example, 16 64K x 1 memories are used, the memory block size will be 128K, not 64K.



ET = ESC Trap/No ESC Trap (1/0)
M/IO = Register block located in Memory/ I/O Space (1/0)
RMX = Normal Interrupt Controller mode/ IRMX compatible Interrupt Controller mode (0/1)

Figure 9. Relocation Register

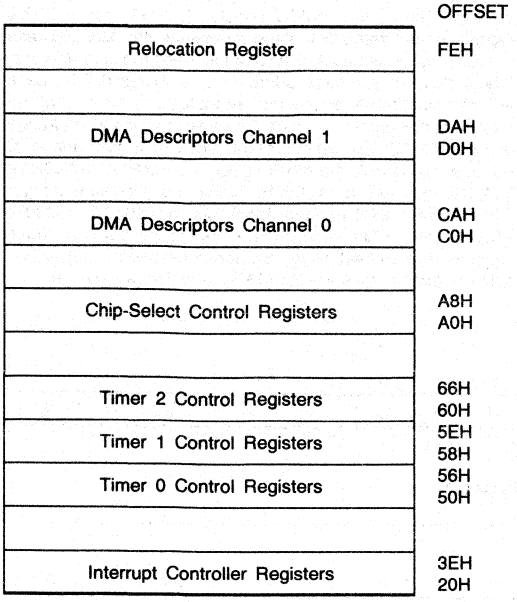


Figure 10. Internal Register Map

Upper Memory $\overline{CS}$

The 80186 provides a chip select, called $\overline{UCS}$, for the top of memory. The top of memory is usually used as the system memory because after reset the 80186 begins executing at memory location FFFF0H.

The upper limit of memory defined by this chip select is always FFFFFH, while the lower limit is programmable. By programming the lower limit, the size of the select block is also defined. Table 7 shows the relationship between the base address selected and the size of the memory block obtained.

Table 7. UMCS Programming Values

Starting Address (Base Address)	Memory Block Size	UMCS Value (Assuming R0 = R1 = R2 = 0)
FFC00	1K	FFF8H
FF800	2K	FFB8H
FF000	4K	FF38H
FE000	8K	FE38H
FC000	16K	FC38H
F8000	32K	F838H
F0000	64K	F038H
E0000	128K	E038H
C0000	256K	C038H

The lower limit of this memory block is defined in the UMCS register (see Figure 11). This register is at offset A0H in the internal control block. The legal values for bits 6-13 and the resulting starting address and memory block sizes are given in Table 7. Any combination of bits 6-13 not shown in Table 7 will result in undefined operation. After reset, the UMCS register is programmed for a 1K area. It must be reprogrammed if a larger upper memory area is desired.

Any internally generally 20-bit address whose upper 16-bits are greater than or equal to UMCS (with bits 0-5 "0") will cause UCS to be activated. UMCS bits R2-R0 are used to specify READY mode for the area of memory defined by this chip-select register, as explained below.

Lower Memory $\overline{CS}$

The 80186 provides a chip select for low memory called LCS. The bottom of memory contains the interrupt vector table, starting at location 00000H.

The lower limit of memory defined by this chip select is always 0H, while the upper limit is programmable. By programming the upper limit, the size of the memory block is also defined. Table 8 shows the relationship between the upper address selected and the size of the memory block obtained.

Table 8. LMCS Programming Values

Upper Address	Memory Block Size	LMCS Value (Assuming R0 = R1 = R2 = 0)
003FFH	1K	0038H
007FFH	2K	0078H
00FFFH	4K	00F8H
01FFFH	8K	01F8H
03FFFH	16K	03F8H
07FFFH	32K	07F8H
0FFFFH	64K	0FF8H
1FFFFH	128K	1FF8H
3FFFFH	256K	3FF8H

The upper limit of this memory block is defined in the LMCS register (see Figure 12). This register is at offset A2H in the internal control block. The legal values for bits 6-15 and the resulting upper address and memory block sizes are given in Table 8. Any combination of bits 6-15 not shown in Table 8 will result in undefined operation. After reset, the LMCS register value is undefined. However, the $\overline{LCS}$ chip-select line will not become active until the LMCS register is accessed.

Any internally generated 20-bit address whose upper 16 bits are less than or equal to LMCS (with bits 0-5 "1") will cause $\overline{LCS}$ to be active. LMCS register bits R2-R0 are used to specify the READY mode for the area of memory defined by this chip-select register.

Mid-Range Memory $\overline{CS}$

The 80186 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located anywhere within the 80186 1M byte memory address space exclusive of the areas defined by $\overline{UCS}$ and $\overline{LCS}$. Both the base address and size of this memory block are programmable.

The size of the memory block defined by the midrange select lines, as shown in Table 9, is determined by bits 8-14 of the MPCS register (see Figure 13). This register is at location A8H in the internal control block. One and only one of bits 8-14 must be set at a time. Unpredictable operation of the $\overline{MCS}$ lines will otherwise occur. Each of the four chip-select lines is active for one of the four equal contiguous divisions of the mid-range block. Thus, if the total block size is 23K, each chip select is active for 8K of memory with $\overline{MCS0}$ being active for the first range and $\overline{MCS3}$ being active for the last range.

The EX and MS in MPCS relate to peripheral functionally as described in a later section.

Table 9. MMCS Programming Values

Total Block Size	Individual Select Size	MMCS Bits 14 – 8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

The base address of the mid-range memory block is defined by bits 15-9 of the MMCS register (see Figure 14). This register is at offset A6H in the internal control block. These bits correspond to bits A19-A13 of the 20-bit memory address. Bits A12-A0 of the base address area always 0. The base address may be set at any integer multiple of the size of the total memory block selected. For example, if the midrange block size is 32K (or the size of the block for which each $\overline{MCS}$ line is active is 8K), the block could be located at 10000H or 18000H, but not at 14000H, since the first few integer multiples of a 32K memory block are 0H, 8000H, 10000H, 18000H, etc. After reset, the contents of both of these registers is undefined. However, none of the $\overline{MCS}$ lines will be active until both the MMCS and MPCS registers are accessed.

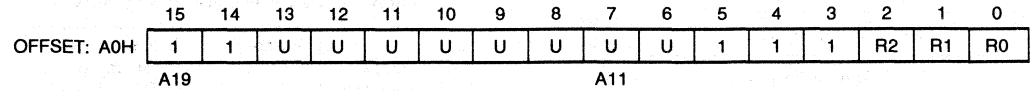


Figure 11. UMCS Register

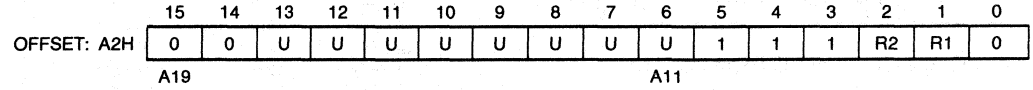


Figure 12. LMCS Register

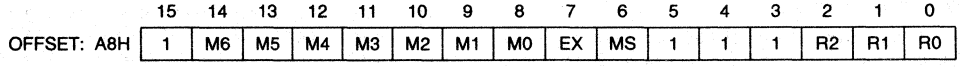


Figure 13. MPCS Register

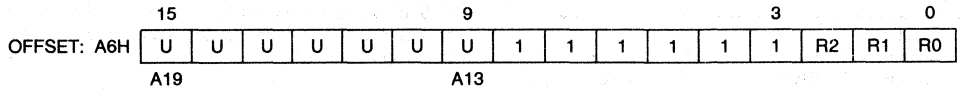


Figure 14. MMCS Register

MMCS bits R2-R0 specify READY mode of operation for all mid-range chip selects. All devices in mid-range memory must use the same number of WAIT states.

The 512K block size for the mid-range memory chip selects is a special case. When using 512K, the base address would have to be at either locations 00000H or 80000H. If it were to be programmed at 00000H when the $\overline{LCS}$ line was programmed, there would be an internal conflict between the $\overline{LCS}$ ready generation logic and the $\overline{MCS}$ ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the $\overline{UCS}$ ready generation logic. Since the $\overline{LCS}$ chip-select line does not become active until programmed, while the $\overline{UCS}$ line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the $\overline{LCS}$ range must not be programmed.

Peripheral Chip Selects

The 80186 can generate chip selects for up to seven peripheral devices. These chip selects are active for seven contiguous blocks of 128 bytes above a programmable base address. This base address may be located in either memory or I/O space.

Seven $\overline{CS}$ lines called $\overline{PCS0}$-6 are generated by the 80186. The base address is user-programmable; however it can only be a multiple of 1K bytes, i.e., the least significant 10 bits of the starting address are always 0.

$\overline{PCS5}$ and $\overline{PCS6}$ can also be programmed to provide latched address bits A1, A2. If so programmed, they cannot be used as peripheral selects. These outputs can be connected directly to the A0, A1 pins used for selecting internal registers of 8-bit peripheral chips. This scheme simplifies the hardware interface because the 8-bit registers of peripherals are simply

treated as 16-bit registers located on even boundaries in I/O space or memory space where only the lower 8-bits of the register are significant: the upper 8-bits are "don't cares."

The starting address of the peripheral chip-select block is defined by the PACS register (see Figure 15). This register is located at offset A4H in the internal control block. Bits 15-6 of this register correspond to bits 19-10 of the 20-bit Programmable Base Address (PBA) of the peripheral chip-select block. Bits 9-0 of the PBA of the peripheral chip-select block are all zeros. If the chip-select block is located in I/O space, bits 12-15 must be programmed zero, since the I/O address is only 16 bits wide. Table 10 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

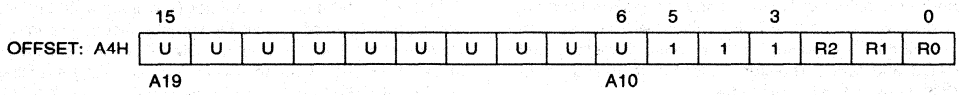


Figure 15. PACS Register

The user should program bits 15-6 to correspond to the desired peripheral base location. PACS bits 0-2 are used to specify READY mode for $\overline{PCS0}$-$\overline{PCS3}$.

Table 10. PCS Address Ranges

PCS Line	Active between Locations
PCS0	PBA — PBA + 127
PCS1	PBA + 128 — PBA + 255
PCS2	PBA + 256 — PBA + 383
PCS3	PBA + 384 — PBA + 511
PCS4	PBA + 512 — PBA + 639
PCS5	PBA + 640 — PBA + 767
PCS6	PBA + 768 — PBA + 895

The mode of operation of the peripheral chip selects is defined by the MPCS register (which is also used to set the size of the mid-range memory chip-select block, see Figure 16). This register is located at offset A8H in the internal control block. Bit 7 is used to select the function of $\overline{PCS5}$ and $\overline{PCS6}$, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 11 describes the programming of these bits. After reset, the contents of both the MPCS and the PACS registers are undefined, however none of the PCS lines will be active until both of the MPCS and PACS registers are accessed.

Table 11. MS, EX Programming Values

Bit	Description
MS	1 = Peripherals mapped into memory space. 0 = Peripherals mapped into I/O space.
EX	0 = 5 $\overline{PCS}$ lines. A1, A2 provided. 1 = 7 $\overline{PCS}$ lines. A1, A2 are not provided.

MPCS bits 0-2 are used to specify READY mode for $\overline{PCS4}$-$\overline{PCS6}$ as outlined below.

READY Generation Logic

The 80186 can generate a "READY" signal internally for each of the memory or peripheral $\overline{CS}$ lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 0-3 wait states for all accesses to the area for which the chip select is active. In addition, the 80186 may be programmed to either ignore external READY for each chip- select range individually or to factor external READY with the integrated ready generator.

READY control consists of 3 bits for each $\overline{CS}$ line or group of lines generated by the 80186. The interpretation of the ready bits is shown in Table 12.

Table 12. READY Bits Programming

R2	R1	R0	Number of WAIT States Generated
0	0	0	0 wait states, external RDY also used.
0	0	1	1 wait state inserted, external RDY also used.
0	1	0	2 wait states inserted, external RDY also used.
0	1	1	3 wait states inserted, external RDY also used.
1	0	0	0 wait states, external RDY ignored.
1	0	1	1 wait state inserted, external RDY ignored.
1	1	0	2 wait states inserted, external RDY ignored.
1	1	1	3 wait states inserted, external RDY ignored.

The internal ready generator operates in parallel with external READY, not in series if the external READY is used (R2 = 0). This means, for example, if the internal generator is set to insert two wait states, but activity on the external READY lines will insert four wait states, the processor will only insert four wait states, not six. This is because the two wait states generated by the internal generator overlapped the first two wait states generated by the external ready signal. Note that the external ARDY and SRDY lines are always ignored during cycles accessing internal peripherals.

R2-R0 of each control word specifies the READY mode for the corresponding block, with the exception of the peripheral chip selects: R2-R0 of PACS set the $\overline{PCS0}$-3 READY mode, R2-R0 of MPCS set the $\overline{PCS4}$-6 READY mode.

Chip Select/Ready Logic and Reset

Upon reset, the Chip-Select/Ready Logic will perform the following actions:

- All chip-select outputs will be driven HIGH.
- Upon leaving RESET, the UCS line will be programmed to provide chip selects to a 1K block with the accompanying READY control bits set at 011 to allow the maximum number of internal wait states in conjunction with external Ready consideration (i.e. UMCS resets to FFFBH).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET: A8H	1	M6	M5	M4	M3	M2	M1	M0	EX	MS	1	1	1	R2	R1	R0

Figure 16. MPCS Register

- No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the $\overline{PCS}$ lines will become active.

DMA CHANNELS

The 80186 DMA controller provides two independent high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., Memory to I/O) or within the same space (e.g., Memory to Memory or I/O to I/O). Data can be transferred either in bytes (8 bits) or in words (16 bits) to or from even or odd addresses. Each DMA channel maintains both a 20-bit source and destination pointer which can be optionally incremented or decremented after each data transfer (by one or two depending on byte or word transfers). Each data transfer consumes 2 bus cycles (a minimum of 8 clocks), one cycle to fetch data and the other to store data. This provides a maximum data transfer rate of one Mword/sec or 2 MBytes/sec.

DMA Operation

Each channel has six registers in the control block which define each channel's specific operation. The control registers consist of a 20-bit Source pointer (2 words), a 20-bit Destination pointer (2 words), a 16-bit Transfer Counter, and a 16-bit Control Word. The format of the DMA Control Blocks is shown in Table 13. The Transfer Count Register (TC) specifies the number of DMA transfers to be performed. Up to 64K byte or word transfers can be performed with automatic termination. The Control Word defines the channel's operation (see Figure 18). All registers may be modified or altered during any DMA activity. Anyl changes made to these registers will be reflected immediately in DMA operation.

Table 13. DMA Control Block Format

Register Name	Register Address	
	Ch. 0	Ch. 1
Control Word	CAH	DAH
Transfer Count	C8H	D8H
Destination Pointer (upper 4 bits)	C6H	D6H
Destination Pointer	C4H	D4H
Source Pointer (upper 4 bits)	C2H	D2H
Source Pointer	C0H	D0H

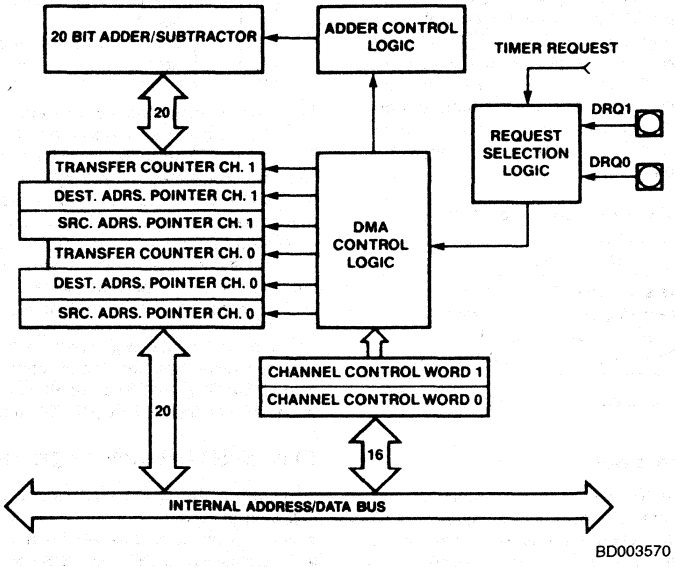


Figure 17. DMA Unit Block Diagram

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M/$\overline{IO}$	DEST DEC	INC	M/$\overline{IO}$	SRC DEC	INC	TC	INT	SYN		P	TDRQ	X	CHG/$\overline{NOCHG}$	ST/$\overline{STOP}$	$\overline{B}$/W

X = DON'T CARE

Figure 18. DMA Control Register

DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 80186 DMA channel. This register specifies:

- the mode of synchronization;
- whether bytes or words will be transferred;
- whether interrupts will be generated after the last transfer;
- whether DMA activity will cease after a programmed number of DMA cycles;
- the relative priority of the DMA channel with respect to the other DMA channel;
- whether the source pointer will be incremented, decremented, or maintained constant after each transfer;
- whether the source pointer addresses memory or I/O space;
- whether the destination pointer will be incremented, decremented, or maintained constant after each transfer; and
- whether the destination pointer will address memory or I/O space.

The DMA channel control registers may be changed while the channel is operating. However, any changes made during operation will affect the current DMA transfer.

DMA Control Word Bit Descriptions

$\overline{B}$/W: Byte/Word (0/1) Transfers.

ST/$\overline{STOP}$: Start/stop (1/0) Channel.

CHG/$\overline{NOCHG}$: Change/Do not change (1/0) ST/$\overline{STOP}$ bit. If this bit is set when writing to the control word, the ST/$\overline{STOP}$ bit will be programmed by the write to the control word. If this bit is cleared when writing the control word, the ST/$\overline{STOP}$ bit will not be altered. This bit is not stored; it will always be a 0 on read.

INT: Enable Interrupts to CPU on byte count termination.

TC: If set, DMA will terminate when the contents of the Transfer Count register reach zero. The ST/$\overline{STOP}$ bit will also be reset at this point if TC is set. If this bit is cleared, the DMA unit will decrement the transfer count register for each DMA cycle, but the DMA transfer will not stop when the contents of the TC register reach zero.

SYN: (2 bits)
- 00 No synchronization
  NOTE: The ST bit will be cleared automatically when the contents of the TC register reach zero regardless of the state of the bit.
- 01 Source synchronization.
- 10 Destination synchronization.
- 11 Unused.

SOURCE: INC Increment source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Source pointer is in M/IO space (1/0).

DEC Decrement source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

DEST: INC Increment destination pointer by 1 or 2 ($\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Destination pointer is in M/IO space (1/0).

DEC Decrement destination pointer by 1 or 2 (depending on $\overline{B}$/W) after each transfer.

P Channel priority-relative to other channel.
0 low priority.

TDRQ
0: Disable DMA requests from timer 2.
1: Enable DMA requests from timer 2.

Bit 3 Bit 3 is not used.

If both INC and DEC are specified for the same pointer, the pointer will remain constant after each cycle.

DMA Destination and Source Pointer Registers

Each DMA channel maintains a 20-bit source and a 20-bit destination pointer. Each of these pointers takes up two full 16-bit registers in the peripheral control block. The lower four bits of the upper register contain the upper four bits of the 20-bit physical address (see Figure 18a). These pointers may be individually incremented or decremented after each transfer. If word transfers are performed the pointer is incremented or

decremented by two. Each pointer may point into either memory or I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the pointer registers. Higher transfer rates can be obtained if all word transfers are performed to even addresses, since this will allow data to be accessed in a single memory access.

DMA Transfer Count Register

Each DMA channel maintains a 16-bit transfer count register (TC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control Register. If the TC bit in the DMA control word is set, however, DMA activity will terminate when the transfer count register reaches zero.

DMA Requests

Data transfers may be either source or destination synchronized, that is either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized; that is, the transfer will take place continually until the correct number of transfers has occurred. When source or unsynchronized transfers are performed, the DMA channel may begin another transfer immediately after the end of a previous DMA transfer. This allows complete transfer to take place every 2 bus cycles or eight clock cycles (assuming no wait states). No prefetching occurs when destination synchronization is performed, however. Data will not be fetched from the source address until the destination device signatls that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller will relinquish control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle will begin after two processor clocks. This is done to allow the destination device time to remove its request if another transfer is not desired. Since the DMA controller will relinquish the bus, the CPU can initiate a bus cycle. As a result, a complete bus cycle will often be inserted between destination synchronized transfers. These lead to the maximum DMA transfer rates shown in Table 14.

Table 14. Maximum DMA Transfer Rates

Type of Synchronization Selected	CPU Running	CPU Halted
Unsynchronized	2MBytes/sec	2MBytes/sec
Source Synch	2MBytes/sec	2MBytes/sec
Destination Synch	1.3MBytes/sec	1.5MBytes/sec

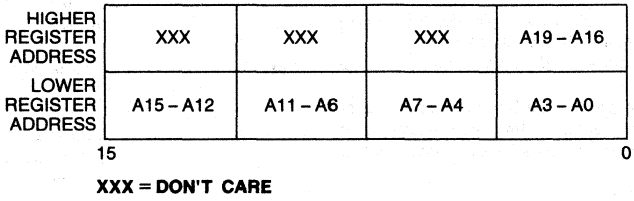


Figure 18a. DMA Memory Pointer Register Format

DMA Acknowledge

No explicit DMA acknowledge pulse is provided. Since both source and destination pointers are maintained, a read from a requesting source, or a write to a requesting destination, should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA pointers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

DMA Priority

The DMA channels may be programmed such that one channel is always given priority over the other, or they may be programmed such as to alternate cycles when both have DMA requests pending. DMA cycles always have priority over internal CPU cycles except between locked memory accesses or word accesses the odd memory locations; however, an external bus hold takes priority over an internal DMA cycle. Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time will suffer during sequences of continuous DMA cycles. An NMI request, however, will cause all internal DMA activity to halt. This allows the CPU to quickly respond to the NMI request.

DMA Programming

DMA cycles will occur whenever the ST/STOP bit of the Control Register is set. If synchronized transfers are programmed, a DRQ must also have been generated. Therefore, the source and destination transfer pointers, and the transfer count register (if used) must be programmed before this bit is set.

Each DMA register may be modified while the channel is operating. If the CHG/NOCHG bit is cleared when the control register is written, the ST/STOP bit of the control register will not be modified by the write. If multiple channel registers are modified, it is recommended that a LOCKED string transfer be used to prevent a DMA transfer from occurring between updates to the channel registers.

DMA Channels and Reset

Upon RESET, the DMA channels will perform the following actions:

- The Start/Stop bit for each channel will be rest to STOP.
- Any transfer in progress is aborted.

TIMERS

The 80186 provides three internal 16-bit programmable timers (see Figure 19). Two of these are highly flexible and are connected to four external pins (2 per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. The third timer is not connected to any external pins, and is useful for real-time coding and time delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source.

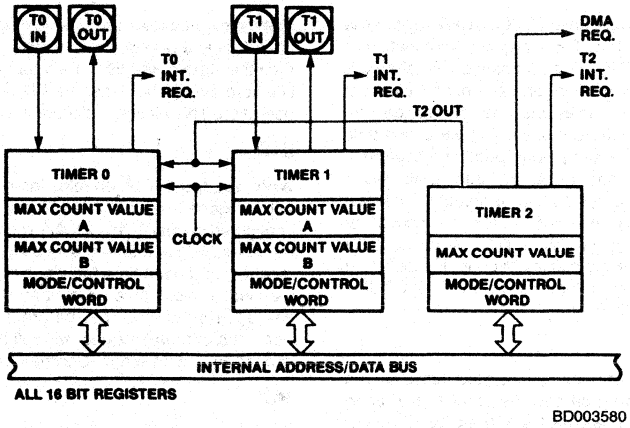


Figure 19. Timer Block Diagram

Timer Operation

The timers are controlled by 11 16-bit registers in the internal peripheral control block. The configuration of these registers is shown in Table 15. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of this register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to zero during that same clock, i.e., the maximum count value is never stored in the count register itself. Timers 0 and 1 are, in addition, equipped with a second MAX COUNT register, which enables the timers to alternate their count between two different MAX COUNT values programmed by the user. If a single MAX COUNT register is used, the timer output pin will switch LOW for a single clock, 2 clocks after the maximum count value has been reached. In the dual MAX COUNT register mode, the output pin will indicate which MAX COUNT register is currently in use, thus allowing nearly complete freedom in selecting waveform duty cycles. For the timers with two MAX COUNT registers, the RIU bit in the control register determines which is used for the comparison.

Each timer gets serviced every fourth CPU-clock cycle, and thus can operate at speeds up to one-quarter the internal clock frequency (one-eighth the crystal rate). External clocking of the timers may be done at up to a rate of one-quarter of the internal CPU-clock rate (2 MHz for an 8 MHz CPU clock). Due to internal synchronization and pipelining of the timer circuitry, a timer output may take up to 6 clocks to respond to any individual clock or gate input.

Since the count registers and the maximum count registers are all 16 bits wide, 16 bits of resolution are provided. Any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle. However, this is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

The timers have several programmable options.

- All three timers can be set to halt or continue on a terminal count.
- Timers 0 and 1 can select between internal and external clocks, alternate between MAX COUNT registers and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

These options are selectable via the timer mode/control word.

Timer Mode/Control Register

The mode/control register (see Figure 20) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

Table 15. Timer Control Block format

Register Name	Register Offset		
	Tmr. 0	Tmr. 1	Tmr. 2
Mode/Control Word	56H	5EH	66H
Max Count B	54H	5CH	not present
Max Count A	52H	5AH	62H
Count Register	50H	58H	60H

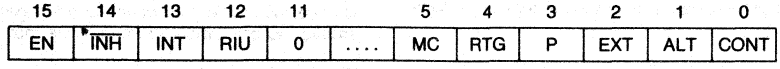


Figure 20. Timer Mode/Control Register

ALT:

The ALT bit determines which of two MAX COUNT registers is used for count comparison. If ALT = 0, register A for that timer is always used, while if ALT = 1, the comparison will alternate between register A and register B when each maximum count is reached. This alternation allows the user to change one MAX COUNT register while the other is being used, and thus provides a method of generating nonrepetitive waveforms. Square waves and pulse outputs of any duty cycle are a subset of available signals obtained by not changing the final count registers. The ALT bit also determines the function of the timer output pin. If ALT is zero, the output pin will go LOW for one clock, the clock after the maximum count is reached, If ALT is one, the output pin will reflect the current MAX COUNT register being used (0/1 for B/A).

CONT:

Setting the CONT bit causes the associated timer to run continuously, while resetting it causes the timer to halt upon maximum count. If CONT = 0 and ALT = 1, the timer will count to the MAX COUNT register A value, reset, count to the register B value, reset, and halt.

EXT:

The external bit selects between internal and external clocking for the timer. The external signal may be asynchronous with respect to the 80186 clock. If this bit is set, the timer will count LOW-to-HIGH transitions on the input pin. If cleared, it will count an internal clock while using the input pin for control. In this mode, the function of the external pin is defined by the RTG bit. The maximum input to output transition latency time may be as much as 6 clocks. However, clock inputs may be pipelined as closely together as every 4 clocks without losing clock pulses.

P:

The prescaler bit is ignored unless internal clocking has been selected (EXT = 0). If the P bit is a zero, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a one, the output of timer 2 will be used as a clock for the timer. Note that the user must initialize and start timer 2 to obtain the prescaled clock.

RTG:

Retrigger bit is only active for internal clocking (EXT = 0). In this case it determines the control function provided by the input pin.

If RTG = 0, the input level gates the internal clock on and off. If the input pin is HIGH, the timer will count; if the input pin is LOW, the timer will hold its value. As indicated previously, the input signal may be asynchronous with respect to the 80186 clock.

When RTG = 1, the input pin detects LOW-to-HIGH transitions. The first such transition starts the timer running, clearing the timer value to zero on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to zero, from which it will start counting up again. If CONT = 0, when the timer has reached maximum count, the EN bit will be cleared, inhibiting further timer activity.

EN:

The enable bit provides programmer control over the timer's RUN/HALT status. When set, the timer is enabled to increment subject to the input pin constraints in the internal clock mode (discussed previously). When cleared, the timer will be inhibited from counting. All input pin transitions during the time EN is zero will be ignored. If CONT is zero, the EN bit is automatically cleared upon maximum count.

INH:

The inhibit bit allows for selective updating of the enable (EN) bit. If INH is a one during the write to the mode/control word, then the state of the EN bit will be modified by the write. If INH is a zero during the write, the EN bit will be unaffected by the operation. This bit is not stored; it will always be a 0 on a read.

INT:

When set, the INT bit enables interrupts from the timer, which will be generated on every terminal count. If the timer is configured in dual MAX COUNT register mode, an interrupt will be generated each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. If this enable bit is cleared after the interrupt request has been generated, but before a pending interrupt is serviced, the interrupt request will still be in force. (The request is latched in the Interrupt Controller.)

MC:

The Maximum Count bit is set whenever the timer reaches its final maximum count value. If the timer is configured in dual MAX COUNT register mode, this bit will be set each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. This bit is set regardless of the timer's interrupt-enable bit. The MC bit gives the user the ability to monitor timer status through software instead of through interrupts.

RIU:

The Register In Use bit indicates which MAX COUNT register is currently being used for comparison to the timer count value. A zero value indicates register A. The RIU bit cannot be written, i.e., its value is not affected when the control register is written. It is always cleared when the ALT bit is zero.

Not all mode bits are provided for timer 2. Certain bits are hardwired as indicated below:

ALT = 0, EXT = 0, P = 0, RTG = 0, RIU = 0

Count Registers

Each of the three timers has a 16-bit count register. The current contents of this register may be read or written by the processor at any time. If the register is written into while the timer is counting, the new value will take effect in the current count cycle.

Max Count Registers

Timers 0 and 1 have two MAX COUNT registers, while timer 2 has a single MAX COUNT register. These contain the number of events the timer will count. In timers 0 and 1, the MAX COUNT register used can alternate between the two max count values whenever the current maximum count is reached. The condition which causes a timer to reset is equivalent between the current count value and the max count being used. This means that if the count is changed to be above the max count value, or if the max count value is changed to be below the current value, the timer will not reset to zero, but rather will count to its maximum value, "wrap around" to zero, then count until the max count is reached.

Timers and Reset

Upon RESET, the Timers will perform the following actions:

- All EN (Enable) bits are reset preventing timer counting.
- All SEL (Select) bits are reset to zero. This selects MAX COUNT register A, resulting in the Timer Out pins going HIGH upon RESET.

INTERRUPT CONTROLLER

The 80186 can receive interrupts from a number of sources, both internal and external. The internal interrupt controller serves to merge these requests on a priority basis, for individual service by the CPU. Internal interrupt sources (Timers and DMA channels) can be disabled by their own control registers or by mask bits within the interrupt controller. The 80186 interrupt controller has its own control registers that set the mode of operation for the controller.

The interrupt controller will resolve priority among requests that are pending simultaneously. Nesting is provided so interrupt service routines for lower priority interrupts may themselves be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 21.

The interrupt controller has a special iRMX 86 compatibility mode that allows the use of the 80186 within the iRMX 86 operating system interrupt structure. The controller is set in this mode by setting bit 14 in the peripheral control block relocation register (see iRMX 86 Compatibility Mode section). In this mode, the internal 80186 interrupt controller functions as a "slave" controller to an external "master" controller. Special initialization software must be included to properly set up the 80186 interrupt controller in iRMX 86 mode.

MASTER (NON-iRMX) MODE OPERATION

Interrupt Controller External Interface

For external interrupt sources, five dedicated pins are provided. One of these pins is dedicated to NMI, non-maskable interrupt. This is typically used for power-fail interrupts, etc. The other four pins may function either as four interrupt input lines with internally generated interrupt vectors, as an interrupt line and an interrupt acknowledge line (called the "cascade mode") along with two other input lines with internally generated interrupt vectors, or as two interrupt input lines and two dedicated interrupt acknowledge ouput lines. When the interrupt lines are configured in cascade mode, the 80186 interrupt controller will not generate internal interrupt vectors.

External sources in the cascade mode use externally generated interrupt vectors. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the vector is read into the 80186 on the second cycle. The capability to interface to external 8259A programmable interrupt controllers is thus provided when the inputs are configured in cascade mode.

Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in non-iRMX mode are similar to the 8259A. The interrupt controller responds identically to internal interrupts in all three modes: the difference is only in the interpretation of function of the four external interrupt pins. The interrupt controller is set into one of these three modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are as follows:

Fully Nested Mode

When in the fully nested mode four pins are used as direct interrupt requests. The vectors for these four inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set, no interrupt will be generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the inservice bit is set, no interrupt will be generated by the interrupt controller. This allows interrupt service routines to operate with interrupts enabled without being themselves interrupted by lower-priority interrupts. Since interrupts are enabled, higher-priority interrupts will be serviced.

When a service routine is completed, the proper IS bit must be reset by writing the proper pattern to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. An EOI command is issued at the end of the service routine just before the issuance of the return from interrupt instruction. If the fully nested structure has been upheld, the next highest-priority source with its IS bit set is then serviced.

Cascade Mode

The 80186 has four interrupt pins and two of them have dual functions. In the fully nested mode the four pins are used as direct interrupt inputs and the corresponding vectors are generated internally. In the cascade mode, the four pins are configured into interrupt input-dedicated acknowledge signal pairs. The interconnection is shown in Figure 22. INT0 is an interrupt input interfaced to an 8259A, while INT2/$\overline{\text{INTA0}}$ serves as the dedicated interrupt acknowledge signal to that peripheral. The same is true for INT1 and INT3/$\overline{\text{INTA1}}$. Each pair can selectively be placed in the cascade or non-cascade mode by programming the proper value into INT0 and INT1 control registers. The use of the dedicated acknowledge signals eliminates the need for the use of external logic to generate $\overline{\text{INTA}}$ and device select signals.

The primary cascade mode allows the capability to serve up to 128 external interrupt sources through the use of external master and slave 8259As. Three levels of priority are created, requiring priority resolution in the 80186 interrupt controller, the master 8259As, and the slave 8259As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt commands must be issued by the programmer.

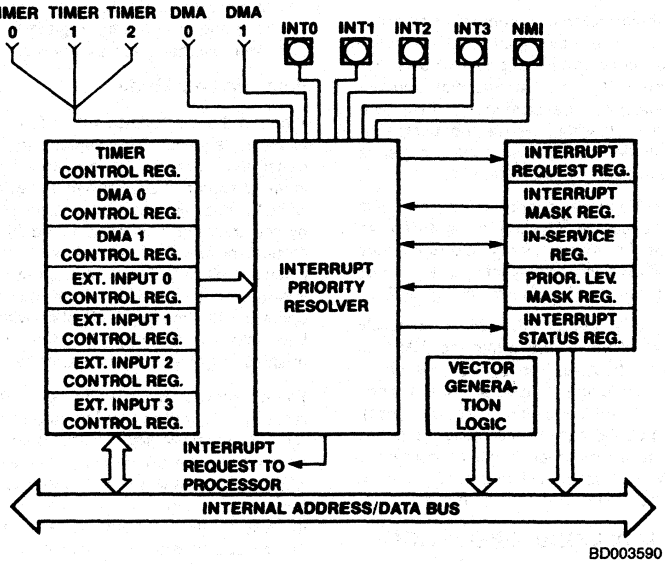


Figure 21. Interrupt Controller Block Diagram

Special Fully Nested Mode

This mode is entered by setting the SFNM bit in INT0 or INT1 control register. It enables complete nestability with external 8259A masters. Normally, an interrupt request from an interrupt source will not be recognized unless the in-service bit for that source is reset. If more than one interrupt source is connected to an external interrupt controller, all of the interrupts will be funneled through the same 80186 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80186 controller until the 80186 in-service bit is reset. In special fully nested mode, the 80186 interrupt controller will allow interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit will continue to be set, however, to inhibit interrupts from other lower-priority 80186 interrupt sources.

Special procedures should be followed when resetting IS bits at the end of interrupt service routines. Software polling of the external master's IS register is required to determine if there is more than one bit set. If so, the IS bit in the 80186 remains active and the next interrupt service routine is entered.

Operation in a Polled Environment

The controller may be used in a polled mode if interrupts are undesirable. When polling, the processor disables interrupts and then polls the interrupt controller whenever it is convenient. Polling the interrupt controller is accomplished by reading the Poll Word (Figure 9). bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Reading the Poll Word causes the In-Service bit of the highest-priority source to be set.

It is desirable to be able to read the Poll Word information without guaranteeing service of any pending interrupt, i.e., not set the indicated in-service bit. The 80186 provides a Poll Status Word in addition to the conventional Poll Word to allow this to be done. Poll Word information is duplicated in the Poll Status Word, but reading the Poll Status Word does not set the associated in-service bit. These words are located in two adjacent memory locations in the register file.

Master Mode Features

Programmable Priority

The user can program the interrupt sources into any of eight different priority levels. The programming is done by placing a 3-bit priority level (0–7) in the control register of each interrupt source. (A source with a priority level of 4 has higher priority over all priority levels from 5 to 7. Priority registers containing values lower than 4 have greater priority.) All interrupt sources have preprogrammed default priority levels (see Table 4).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 4 is used. If the serviced interrupt routine reenables interrupts, it allows other requests to be serviced.

End-of-Interrupt Command

The end-of-interrupt (EOI) command is used by the programmer to reset the In-Service (IS) bit when an interrupt service routine is completed. The EOI command is issued by writing the proper pattern to the EOI register. There are two types of EOI commands, specific and nonspecific. The nonspecific command does not specify which IS bit is reset. When issued, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. A specific EOI command requires that the programmer send the interrupt vector type to the interrupt controller indicating which source's IS bit is to be reset. This command is used when the fully nested structure has been disturbed or the highest priority IS bit that was set does not belong to the service routine in progress.

Trigger Mode

The four external interrupt pins can be programmed in either edge- or level-trigger mode. The control register for each external source has a level-trigger mode (LTM) bit. All interrupt inputs are active HIGH. In the edge sense mode or the level-trigger mode the interrupt request must remain active (HIGH) until the interrupt request is acknowledged by the 80186 CPU. In the edge-sense mode, if the level remains high after the interrupt is acknowledged, the input is disabled and no further requests will be generated. The input level must go LOW for at least one clock cycle to reenable the input. In the level-trigger mode, no such provision is made: holding the interrupt input HIGH will cause continuous interrupt requests.

Interrupt Vectoring

The 80186 Interrupt Controller will generate interrupt vectors for the integrated DMA channels and the integrated Timers. In addition, the Interrupt Controller will generate interrupt vectors for the external interrupt lines if they are not configured in Cascade or Special Fully Nested Mode. The interrupt vectors generated are fixed and cannot be changed (see Table 4).

Interrupt Controller Registers

The Interrupt Controller register model is shown in Figure 23. It contains 15 registers. All registers can both be read or written unless specified otherwise.

In-Service Register

This register can be read from or written into. The format is shown in Figure 24. It contains the In-Service bit for each of the interrupt sources. The In-Service bit for each of the interrupt sources. The In-Service bit is set to indicate that a source's service routine is in progress. When an In-Service bit is set, the interrupt controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the In-Service bit for all three timers; the D0 and D1 bits are the In-Service bits for the two DMA channels; the 10–13 are the In-Service bits for the external interrupt pins. The IS bit is set when the processor acknowledges an interrupt request either by an interrupt acknowledge or by reading the poll register. The IS bit is reset at the end of the interrupt service routine by an end-of-interrupt command issued by the CPU.

Interrupt Request Register

The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 24. A read from this register yields the status of these bits. The TMR bit is the logical OR of all timer interrupt requests. D0 and D1 are the interrupt request bits for the DMA channels.

The state of the external interrupt input pins is also indicated. The state of the external interrupt pins is not a stored condition inside the interrupt controller, therefore the external interrupt bits cannot be written. The external interrupt request bits show exactly when an interrupt request is given to the interrupt controller, so if edge-triggered mode is selected, the bit in the register will be HIGH only after an inactive-to-active transition. For internal interrupt sources, the register bits are set when a request arrives and are reset when the processor acknowledges the requests.

Mask Register

This is a 16-bit register that contains a mask bit for each interrupt source. The format for this register is shown in Figure 24. A one in a bit position corresponding to a particular source serves to mask the source from generating interrupts. These mask bits are the exact same bits which are used in the individual control registers; programming a mask bit using the mask register will also change this bit in the individual control registers, and vice versa.

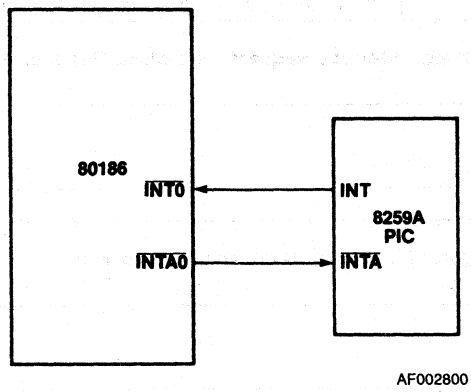


Figure 22. Cascade Mode Interrupt Connection

	OFFSET
INT3 CONTROL REGISTER	3EH
INT2 CONTROL REGISTER	3CH
INT1 CONTROL REGISTER	3AH
INT0 CONTROL REGISTER	38H
DMA 1 CONTROL REGISTER	36H
DMA 0 CONTROL REGISTER	34H
TIMER CONTROL REGISTER	32H
INTERRUPT CONTROLLER STATUS REGISTER	30H
INTERRUPT REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY MASK REGISTER	2AH
MASK REGISTER	28H
POLL STATUS REGISTER	26H
POLL REGISTER	24H
EOI REGISTER	22H

Figure 23. Interrupt Controller Registers (Non-IRMX 86 Mode)

Priority Mask Register

This register is used to mask all interrupts below particular interrupt priority levels. The format of this register is shown in Figure 25. The code in the lower three bits of this register inhibits interrupts of priority lower (a higher priority number) than the code specified. For example, 100 written into this register masks interrupts of level five (101), six (110), and seven (111). The register is reset to seven (111) upon RESET so all interrupts are unmasked.

Interrupt Status Register

This register contains general interrupt controller status information. The format of this register is shown in Figure 26. The bits in the status register have the following functions:

DHLT: DMA Halt Transfer; setting this bit halts all DMA transfers. It is automatically set whenever a non-maskable interrupt occurs, and it is reset when an IRET instruction is executed. The purpose of this bit is to allow prompt service of all non-maskable interrupts. This bit may also be set by the CPU.

IRTx: These three bits represent the individual timer interrupt request bits. These bits are used to differentiate the timer interrupts, since the timer IR bit in the interrupt request register is the "OR" function of all timer interrupt requests. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

15	14	. . .	10	9	8	7	6	5	4	3	2	1	0
0	0	. . .	0	0	0	13	12	11	10	D1	D0	0	TMR

Figure 24. In-Service, Interrupt Request, and Mask Register Formats

15	14	. . .	3	2	1	0
0	0	. . .	0	PRM2	PRM1	PRM0

Figure 25. Priority Mask Register Format

15	14	. . .	7	6	5	4	3	2	1	0
DHLT	0	. . .	0	0	0	0	0	IRT2	IRT1	IRT0

Figure 26. Interrupt Status Register Format

Timer, DMA 0, 1; Control Registers

These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 27. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as are in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

INT0–INT3 Control Registers

These registers are the control words for the four external input pins. Figure 28 shows the format of the INT0 and INT1 Control registers; Figure 29 shows the format of the INT2 and INT3 Control registers. In cascade mode or special fully nested mode, the control words for INT2 and INT3 are not used.

The bits in the various control registers are encoded as follows:

PRO-2: Priority programming information. Highest priority = 000, lowest priority = 111.

LTM: Level-trigger mode bit. 1 = level-triggered; 0 = edge-triggered. Interrupt Input levels are active high. In level-triggered mode, an interrupt is generated whenever the external line is high. In edge-triggered mode, an interrupt will be generated only when this level is preceded by an inactive-to-active transition on the line. In both cases, the level must remain active until the interrupt is acknowledged.

MSK: Mask bit, 1 = mask; 0 = nonmask.

C: Cascade mode bit, 1 = cascade; 0 = direct

SFNM: Special fully nested mode bit, 1 = SFNM; 0 = normal nested mode.

EOI Register

The end of the interrupt register is a command register which can only be written into. The format of this register is shown in Figure 30. It initiates an EOI command when written to by the 80186 CPU.

The bits in the EOI register are encoded as follows:

S_X: Encoded information that specifies an interrupt source vector type as shown in Table 4. For example, to reset the In-Service bit for DMA channel 0, these bits should be set to 01010, since the vector type for DMA channel 0 is 10. Note that to reset the single In-Service bit for any of the three timers, the vector type for timer 0(8) should be written in this register.

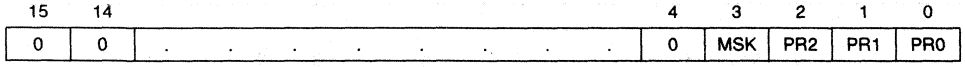


Figure 27. Timer/DMA Control Register Formats

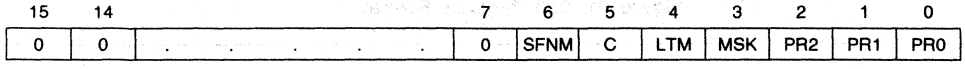


Figure 28. INT0/INT1 Control Register Formats

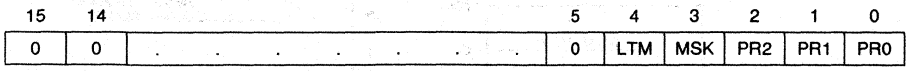


Figure 29. INT2/INT3 Control Register Formats

NSPEC/: A bit that determines the type of EOI command.

SPEC Nonspecific = 1, Specific = 0.

Poll and Poll Status Registers

These registers contain polling information. The format of these registers is shown in Figure 31. They can only be read. Reading the Poll register constitutes a software poll. This will set the IS bit of the highest priority pending interrupt. Reading the poll status register will not set the IS bit of the highest priority pending interrupt; only the status of pending interrupts will be provided.

Encoding of the Poll and Poll Status register bits are as follows:

S_X: Encoded information that indicates the vector type of the highest priority interrupting source. Valid only when INTREQ = 1.

INTREQ: This bit determines if an interrupt request is present. Interrupt Request = 1; no Interrupt Request = 0.

iRMX 86 COMPATIBILITY MODE

This mode allows iRMX 86–80186 compatibility. The interrupt model of iRMX 86 requires one master and multiple slave 8259As in cascaded fashion. When iRMX mode is used, the internal 80186 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80186 resources will be monitored through the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80186 interrupt controller will be in the non-iRMX 86 mode of operation. To set the controller in the iRMX 86 mode, bit 14 of the Relocation Register should be set.

Because of pin limitations caused by the need to interface to an external 8259A master, the internal interrupt controller will no longer accept external inputs. There are however, enough 80186 interrupt controller inputs (internally) to dedicate one to each timer. In this mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The iRMX 86 operating system requires peripherals to be assigned fixed priority levels. This is incompatible with the normal operation of the 80186 interrupt controller. Therefore, the initialization software must program the proper priority levels for each source. The required priority levels for the internal interrupt sources in iRMX mode are shown in Table 16.

Table 16. Internal Source Priority Level

Priority Level	Interrupt Source
0	Timer 0
1	(reserved)
2	DMA 0
3	DMA 1
4	Timer 1
5	Timer 2

These level assignments must remain fixed in the iRMX 86 mode of operation.

iRMX 86 Mode External Interface

The configuration of the 80186 with respect to an external 8259A master is shown in Figure 32. The INT0 input is used as the 80186 CPU interrupt input. INT3 functions as an output to send the 80186 slave-interrupt-request to one of the 8 master-PIC-inputs.

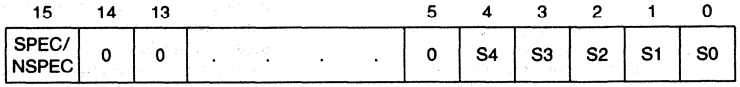


Figure 30. EOI Register Format

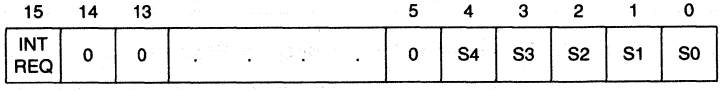


Figure 31. Poll Register Format

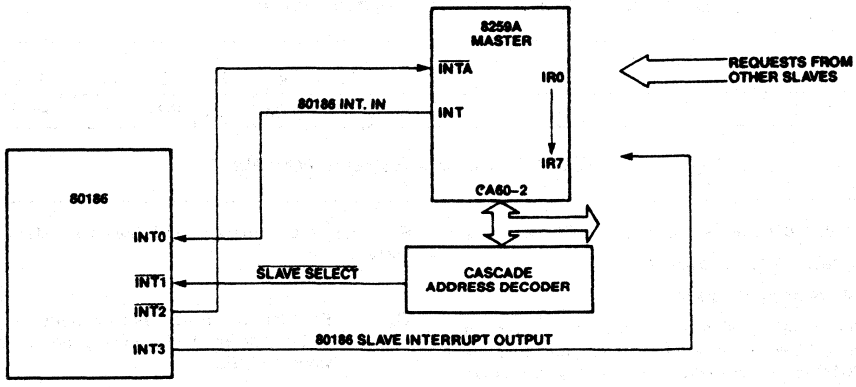


Figure 32. iRMX 86 Interrupt Controller Interconnection

Correct master-slave interface requires decoding of the slave addresses (CAS0–2). Slave 8259As do this internally. Because of pin limitations, the 80186 salve address will have to be decoded externally. $\overline{INT1}$ is used as a slave-select input. Note that the slave vector address is transferred internally, but the READY input must be supplied externally.

$\overline{INT2}$ is used as an acknowledge output, suitable to drive the $\overline{INTA}$ input of an 8259A.

Interrupt Nesting

iRMX 86 mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

Vector Generation in the iRMX 86 MODE

Vector generation in iRMX mode is exactly like that of an 8259A slave. The interrupt controller generates an 8-bit vector which the CPU multiplies by four and uses as an address into a vector table. The significant five bits of the vector are user-programmable while the lower three bits are generated by the priority logic. These bits represent the encoding of the priority level requesting service. The significant five bits of the vector are programmed by writing to the Interrupt Vector register at offset 20H.

Specific End-of-Interrupt

In iRMX mode the specific EOI command operates to reset an in-service bit of a specific priority. The user supplies a 3-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22H.

Interrupt Controller Registers in the iRMX 86 Mode

All control and command registers are located inside the internal peripheral control block. Figure 33 shows the offsets of these registers.

End-of-Interrupt Register

The end-of-interrupt register is a command register which can only be written. The format of this register is shown in Figure 34. It initiates an EOI command when written by the 80186 CPU.

The bits in the EOI register are encoded as follows:

L_x: Encoded value indicating the priority of the IS bit to be reset.

In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the internal sources. The format for this register is shown in Figure 35. Bit positions 2 and 3 correspond to the DMA channels; positions 0, 4, and 5 correspond to the integral timers. The source's IS bit is set when the processor acknowledges its interrupt request.

Interrupt Request Register

This register indicates which internal peripherals have interrupt requests pending. The format of this register is shown in Figure 35. The interrupt request bits are set when a request arrives from an internal source, and are rest when the processor acknowledges the request.

Mask Register

This register contains a mask bit for each interrupt source. The format for this register is shown in Figure 35. If the bit in this register corresponding to a particular interrupt source is set, any interrupts from that source will be masked. These mask bits are exactly the same bits which are used in the individual control registers, i.e., changing the state of a mask bit in this register will also change the state of the mask bit in the individual interrupt control register corresponding to the bit.

Control Registers

These registers are the control words for all the internal interrupt sources. The format of these registers is shown in Figure 36. Each of the timers and both of the DMA channels have their own Control Register.

The bits of the Control Registers are encoded as follows:

pr_x: 3-bit encoded field indicating a priority level for the source; note that each source must be programmed at specified levels.

msk: mask bit for the priority level indicated by pr_x bits.

	OFFSET
LEVEL 5 CONTROL REGISTER (TIMER 2)	3AH
LEVEL 4 CONTROL REGISTER (TIMER 1)	38H
LEVEL 3 CONTROL REGISTER (DMA 1)	36H
LEVEL 2 CONTROL REGISTER (DMA 0)	34H
LEVEL 0 CONTROL REGISTER (TIMER 0)	32H
INTERRUPT-REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY-LEVEL MASK REGISTER	2AH
MASK REGISTER	28H
SPECIFIC EOI REGISTER	22H
INTERRUPT VECTOR REGISTER	20H

Figure 33. Interrupt Controller Registers (iRMX86 Mode)

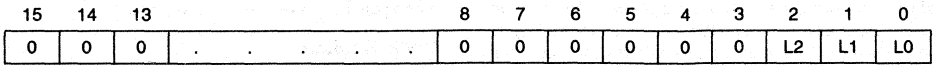


Figure 34. Specific EOI Register Format

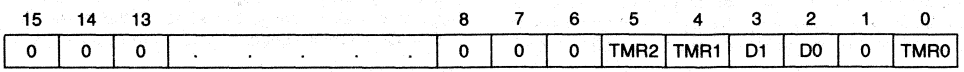


Figure 35. In-Service, Interrupt Request, and Mask Register Format

Interrupt Vector Register

This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 37. The interrupt controller itself provides the lower three bits of the interrupt vector as determined by the priority level of the interrupt request.

The format of the bits in this register is:

t_x: 5-bit field indicating the upper five bits of the vector address.

Priority-Level Mask Register

This register indicates the lowest priority-level interrupt which will be serviced.

The encoding of the bits in this register is:

m_x: 3-bit encoded field indication priority-level value. All levels of lower priority will be masked.

Interrupt Controller and Reset

Upon RESET, the interrupt controller will perform the following actions:

- All SFNM bits reset to 0, implying Fully Nested Mode.
- All PR bits in the various control registers set to 1. Tis places all sources at lowest priority (level 111).
- All LTM bits reset to 0, resulting in edge-sense mode.
- All Interrupt Service bits reset to 0.
- All Interrupt Request bits reset to 0.
- All MSK (Interrupt Mask) bits set to 1 (mask).
- All C (Cascade) bits reset to 0 (non-cascade).
- All PRM (Priority Mask) bits set to 1, implying no levels masked.
- Initialized to non-iRMX 86 mode.

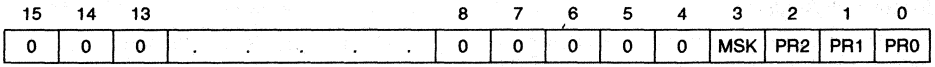


Figure 36. Control Word Format

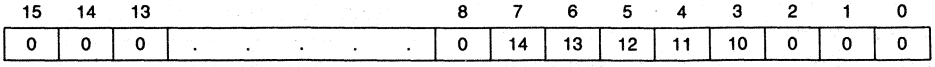


Figure 37. Interrupt Vector Register Format

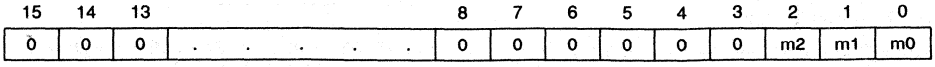


Figure 38. Priority Level Mask Register

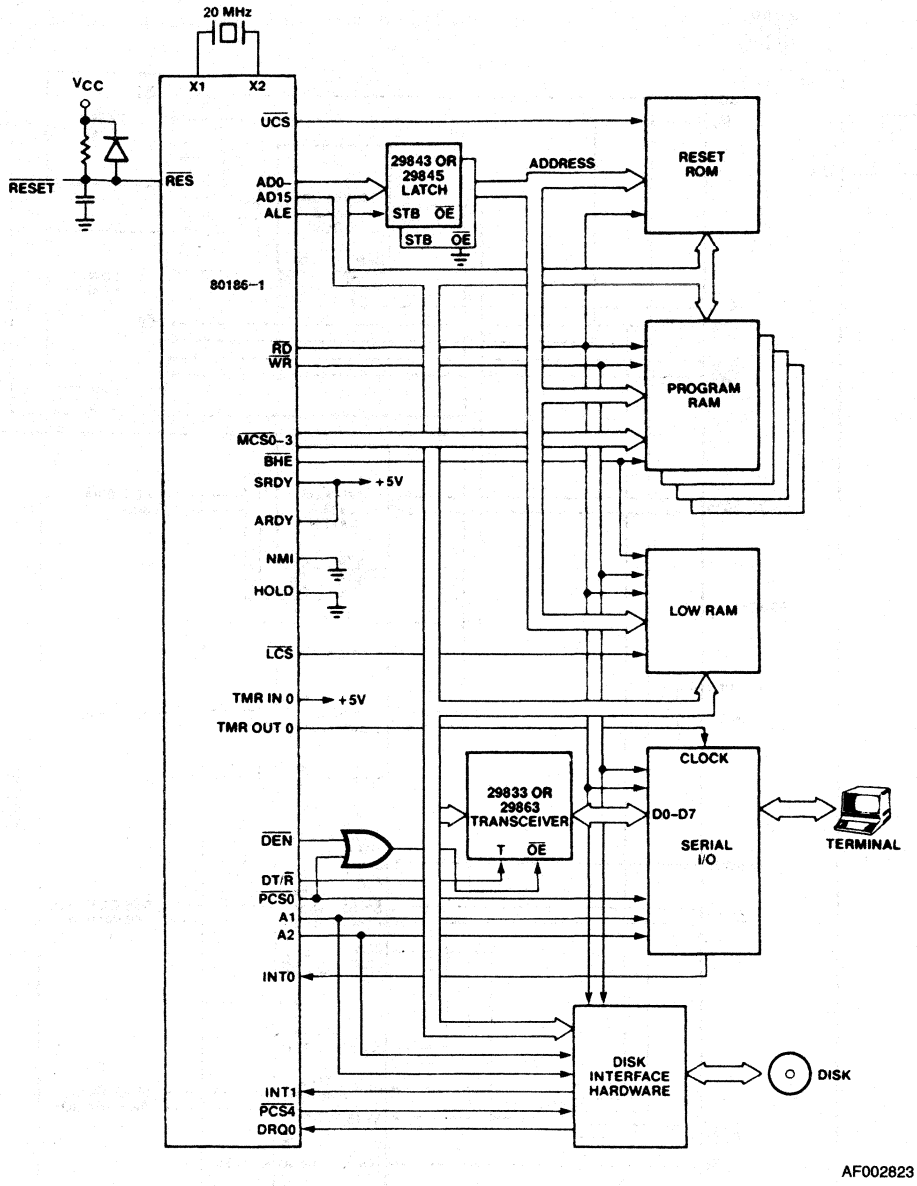


Figure 39. Typical 80186 Computer

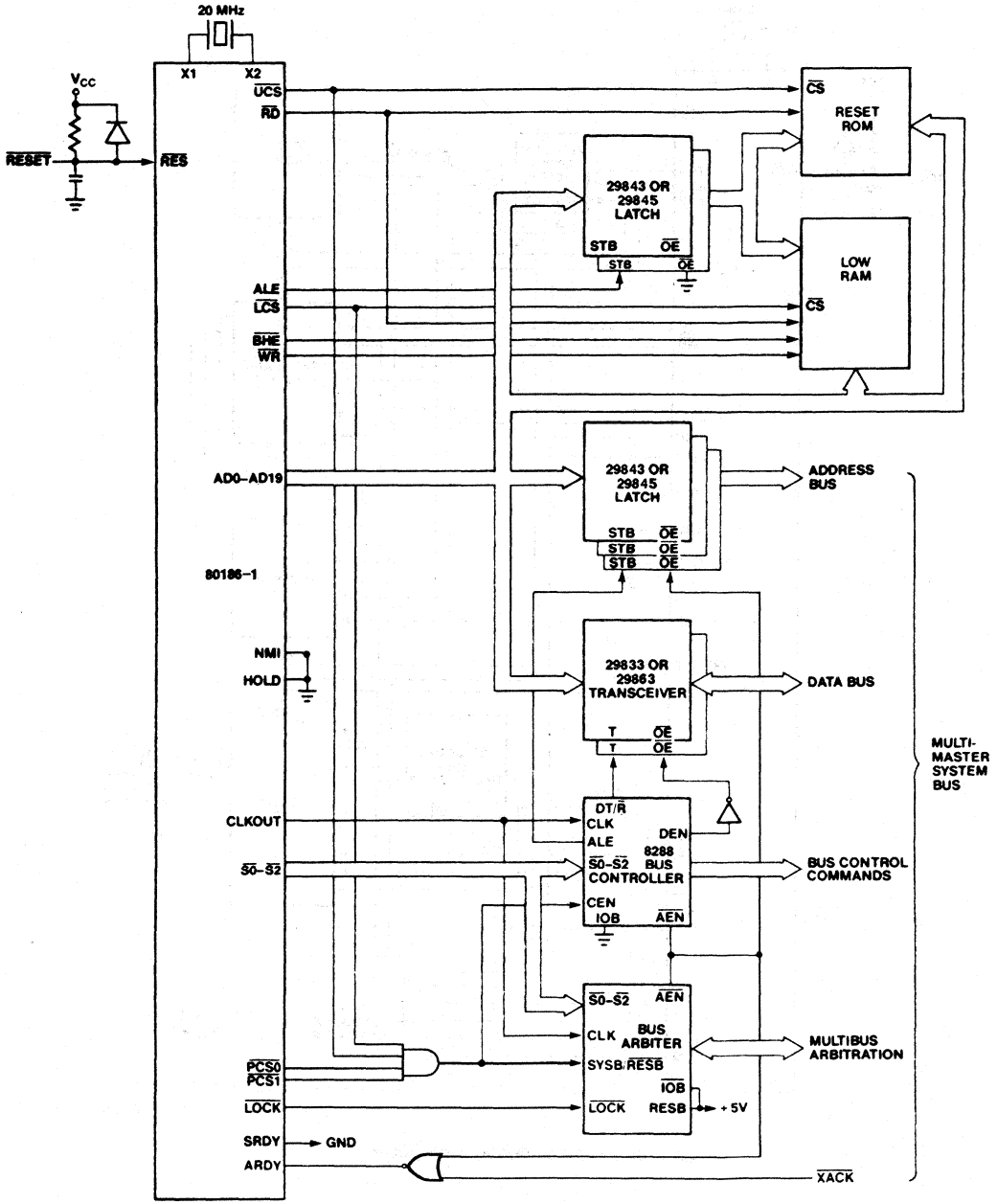


Figure 40. Typical 80186 Multi-Master Bus Interface

PACKAGE

The 80186 is housed in a 68-pin, leadless JEDEC type A Option 2 Ceramic Leadless Chip Carrier. Figure 41 illustrates the package dimensions.

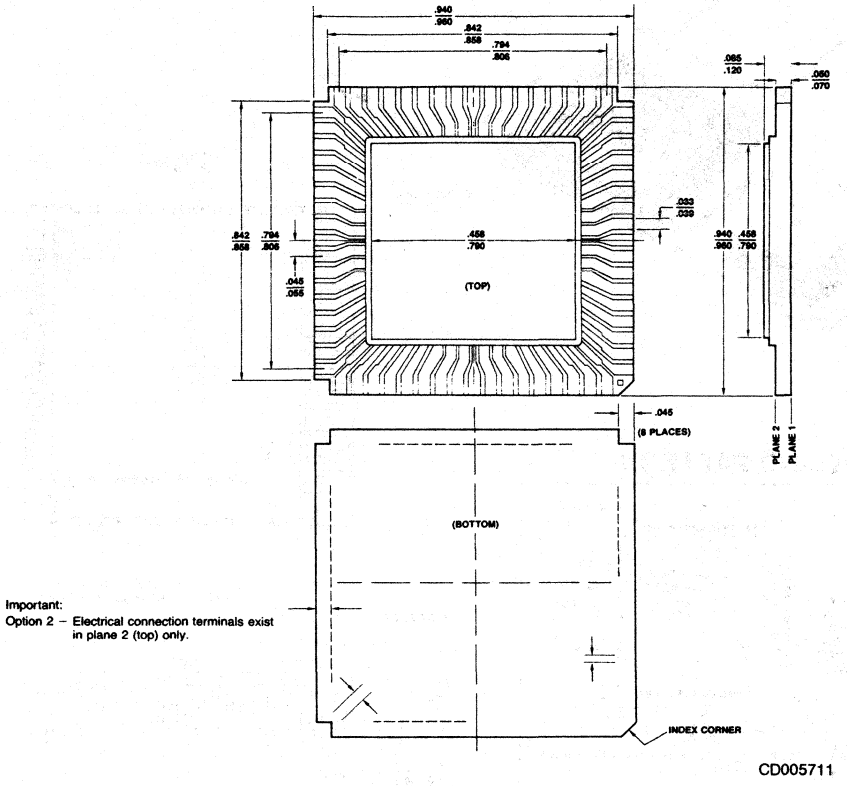


Figure 41. 80186 JEDEC Type A Option 2 Ceramic LCC Package

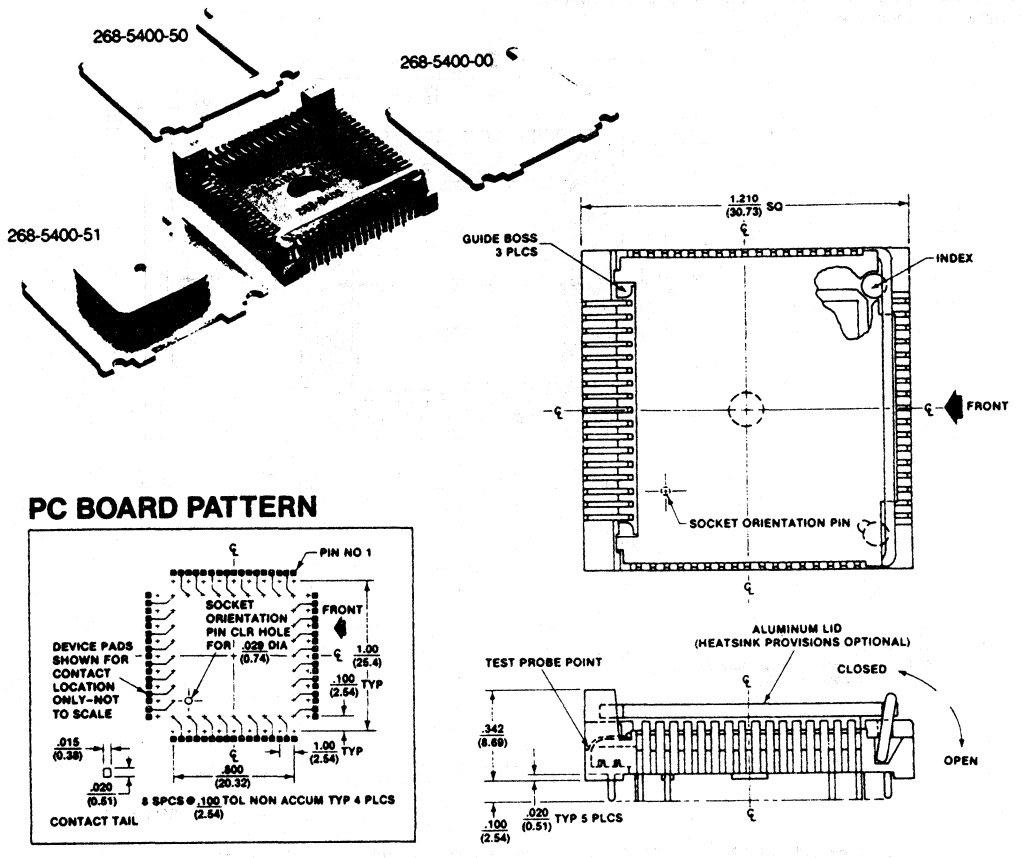


Figure 42. Textool 68 Lead Chip Carrier Socket

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground..... −1.0V to +7V
Power Dissipation .....3 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	T_A	V_{CC}
80186-1 (10MHz)	0 to 70°C	5.0V ±5%
80186-3 (8MHz) 80186-6 (6MHz)	0 to 70°C	5.0V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	Volts
V_{IH}	Input High Voltage (All except X1 and $\overline{RES}$)		2.0	V_{CC} + 0.5	Volts
V_{IH1}	Input High Voltage ($\overline{RES}$)		3.0	V_{CC} + 0.5	Volts
V_{OL}	Output Low Voltage	I_a = 2.5mA for $\overline{S0}$ − $\overline{S2}$ I_a = 2.0mA for all other outputs		0.45	Volts
V_{OH}	Output High Voltage	I_{oa} = −400μA	2.4		Volts
I_{CC}	Power Supply Current	T_A = 0°C		550	mA
		T_A = 70°C		450	
I_{LI}	Input Leakage Current	0V < V_{IN} < V_{CC}		±10	μA
I_{LO}	Output Leakage Current	0.45V < V_{OUT} < V_{CC}		±10	μA
V_{CLO}	Clock Output Low	I_a = 4.0mA		0.6	Volts
V_{CHO}	Clock Output High	I_{oa} = −200μA	4.0		Volts
V_{CLI}	Clock Input Low Voltage		−0.5	0.6	Volts
V_{CHI}	Clock Input High Voltage		3.9	V_{CC} + 1.0	Volts
C_{IN}	Input Capacitance			10	pF
C_{IO}	I/O Capacitance			20	pF

SWITCHING CHARACTERISTICS

PIN TIMING

80186 Timing Requirements All Timings Measured At 1.5 Volts Unless Otherwise Noted

			80186-1 (10MHz)		80186-3 (8MHz) 80186-6 (6MHz)		
Parameters	**Description**	**Test Conditions**	**Min**	**Max**	**Min**	**Max**	**Units**
TDVCL	Data in Setup (A/D)		15		20		ns
TCLDX	Data in Hold (A/D)		8		10		ns
TARYHCH	Asynchronous Ready (AREADY) active setup time*		15		20		ns
TARYLCL	AREADY inactive setup time		25		35		ns
TCHARYX	AREADY hold time		12		15		ns
TSRYCL	Synchronous Ready (SREADY) transition setup time		20		20		ns
TCLSRY	SREADY transition hold time		12		15		ns
THVCL	HOLD Setup*		20		25		ns
TINVCH	INTR, NMI, $\overline{TEST}$, TIMERIN, Setup*		20		25		ns
TINVCL	DRQ0, DRQ1, Setup*		20		25		ns

*To guarantee recognition at next clock.

SWITCHING CHARACTERISTICS (Continued)

80186 Master Interface Timing Responses

Parameters	Description	Test Conditions	80186-1 (10MHz)		80186-3 (8MHz)		80186-6 (6MHz)		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	C_L = 20 – 200 pF all outputs	5	40	10	55	5	63	ns
TCLAX	Address Hold		5		10		10		ns
TCLAZ	Address Float Delay		TCLAX	30	TCLAX	35	TCLAX	44	ns
TCHCZ	Command Lines Float Delay			35		45		56	ns
TCHCV	Command Lines Valid Delay (after float)			45		55		76	ns
TLHLL	ALE Width		TCLCL-25		TCLCL-35		TCLCL-35		ns
TCHLH	ALE Active Delay			30		35		44	ns
TCHLL	ALE Inactive Delay			30		35		44	ns
TLLAX	Address Hold to ALE Inactive		TCHCL-20		TCHCL-25		TCHCL-30		ns
TCLDV	Data Valid Delay		5	35	10	44	10	55	ns
TCLDOX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold after WR		TCLCL-30		TCLCL-40		TCLCL-50		ns
TCVCTV	Control Active Delay1		10	55	10	70	10	87	ns
TCHCTV	Control Active Delay2		10	45	10	55	10	76	ns
TCVCTX	Control Inactive Delay		5	45	5	55	5	76	ns
TCVDEX	$\overline{DEN}$ Inactive Delay (Non-Write Cycle)			55	10	70	10	87	ns
TAZRL	Address Float to $\overline{RD}$ Active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	55	10	70	10	87	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	45	10	55	10	76	ns
TRHAV	$\overline{RD}$ Inactive to Address Active		TCLCL-30		TCLCL-40		TCLCL-50		ns
TCLHAV	HLDA Valid Delay		5	40	5	50	5	67	ns
TRLRH	$\overline{RD}$ Width		2TCLCL-40		2TCLCL-50		2TCLCL-50		ns
TWLWH	$\overline{WR}$ Width		2TCLCL-30		2TCLCL-40		2TCLCL-40		ns
TAVAL	Address Valid to ALE Low		TCLCH-20		TCLCH-25		TCLCH-45		ns
TCHSV	Status Active Delay		10	45	10	55	10	76	ns
TCLSH	Status Inactive Delay		10	45	10	55	10	76	ns
TCLTMV	Timer Output Delay	100 pF max		45		60		75	ns
TCLRO	Reset Delay			45		60		75	ns
TCHQSV	Queue Status Delay			30		35		44	ns
TCHDX	Status Hold Time		10		10		10		ns
TAVCH	Address Valid to Clock High		10		10		10		ns

80186 Chip-Select Timing Responses

Parameters	Description	Test Conditions	80186-1 (10MHz)		80186-3 (8MHz)		80186-6 (6MHz)		Units
			Min	Max	Min	Max	Min	Max	
TCLCSV	Chip-Select Active Delay			50		66		80	ns
TCXCSX	Chip-Select Hold from Command Inactive		25		35		35		ns
TCHCSX	Chip-Select Inactive Delay		10	30	5	35	5	47	ns

SWITCHING CHARACTERISTICS (Continued)

80186 CLKIN Requirements

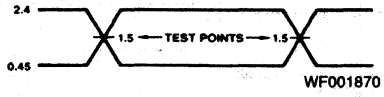
Parameters	Description	Test Conditions	80186-1 (10MHz)		80186-3 (8MHz)		80186-6 (6MHz)		Units
			Min	Max	Min	Max	Min	Max	
TCKIN	CLKIN Period		50	250	62.5	250	83	250	ns
TCKHL	CLKIN Fall Time	3.5 to 1.0 volts		8		10		10	ns
TCKLH	CLKIN Rise Time	1.0 to 3.5 volts		8		10		10	ns
TCLCK	CLKIN Low Time	1.5 volts	20		25		33		ns
TCHCK	CLKIN High Time	1.5 volts	20		25		33		ns

80186 CLKOUT Timing (200 pF load)

Parameters	Description	Test Conditions	Min	Max	Min	Max	Min	Max	Units
TCICO	CLKIN to CLKOUT Skew			40		50		62.5	ns
TCLCL	CLKOUT Period		100	400	125	500	167	500	ns
TCLCH	CLKOUT Low Time	1.5 volts	45		½TCLCL-7.5		½TCLCL-7.5		ns
TCHCL	CLKOUT High Time	1.5 volts	45		½TCLCL-7.5		½TCLCL-7.5		ns
TCH1CH2	CLKOUT Rise Time	1.0 to 3.5 volts		12		15		15	ns
TCL2CL1	CLKOUT Fall Time	3.5 to 1 volts		12		15		15	ns

All timings measured at 1.5 volts unless otherwise noted.

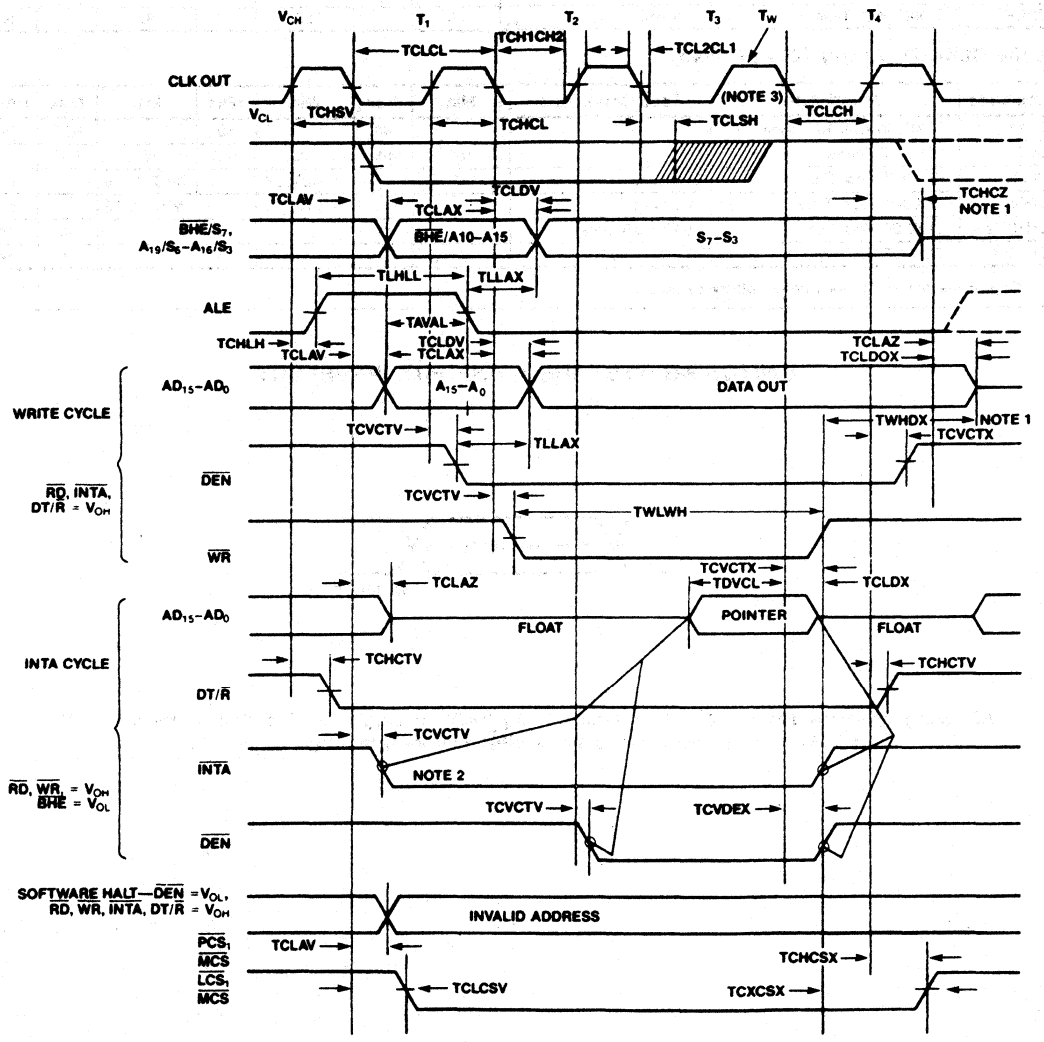
SWITCHING TEST INPUT/OUTPUT WAVEFORM



AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." The clock is driven at 4.3V and 0.25V. Timing measurements are made at 1.5V for both a logic "1" and "0."

SWITCHING WAVEFORMS

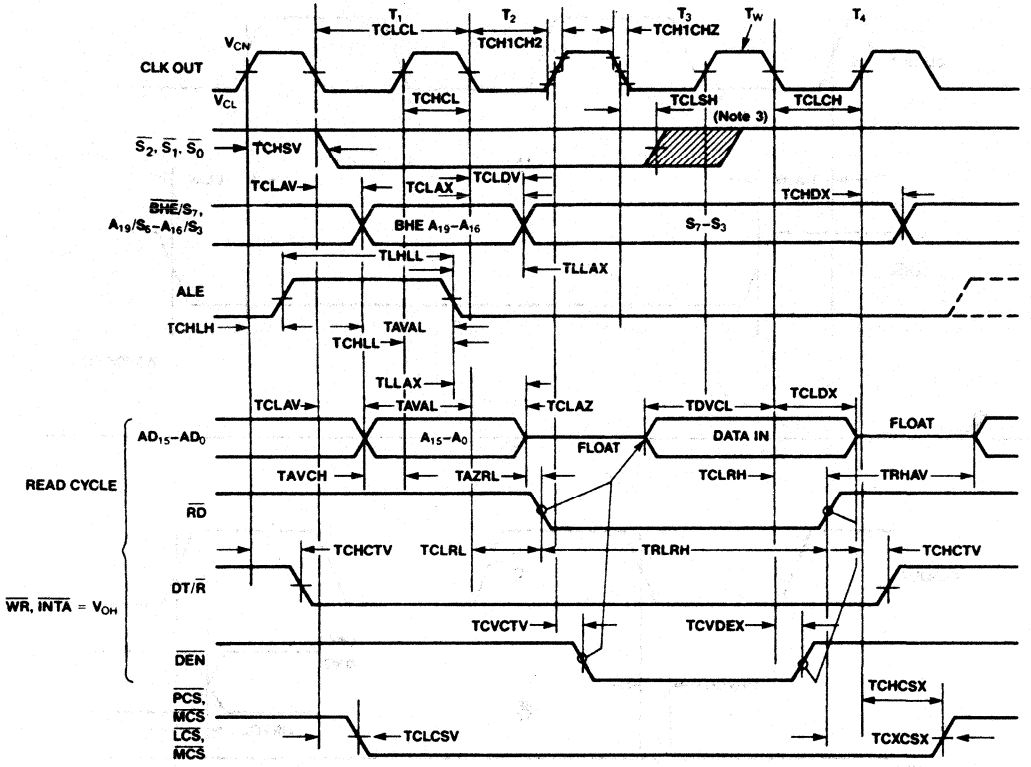
MAJOR CYCLE TIMING



WF006210

SWITCHING WAVEFORMS (Continued)

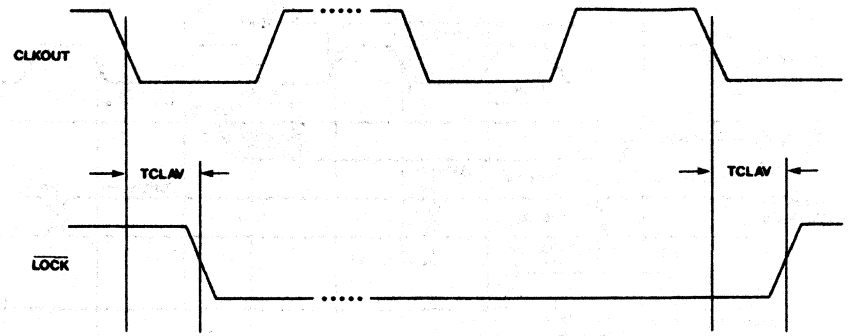
MAJOR CYCLE TIMING (Continued)



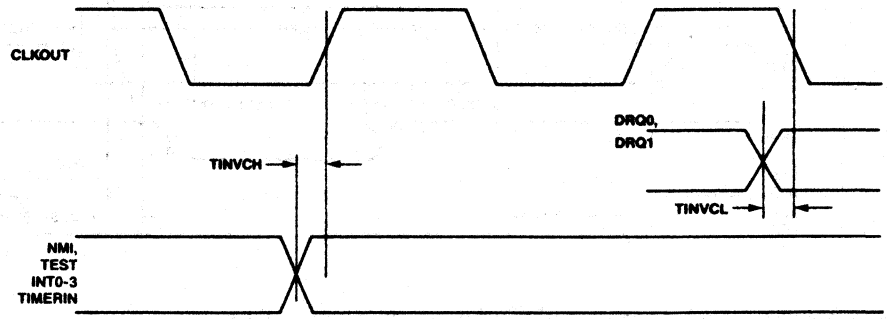
WF006221

Notes:
1. Following a Write cycle, the Local Bus is floated by the 80186 only when the 80186 enters a "Hold Acknowledge" state.
2. INTA occurs one clock later in RMX-mode.
3. Status inactive just prior to T_4.

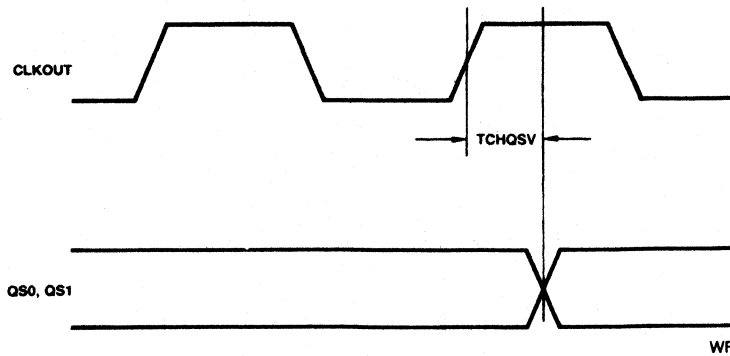
SWITCHING WAVEFORMS (Continued)



WF006230



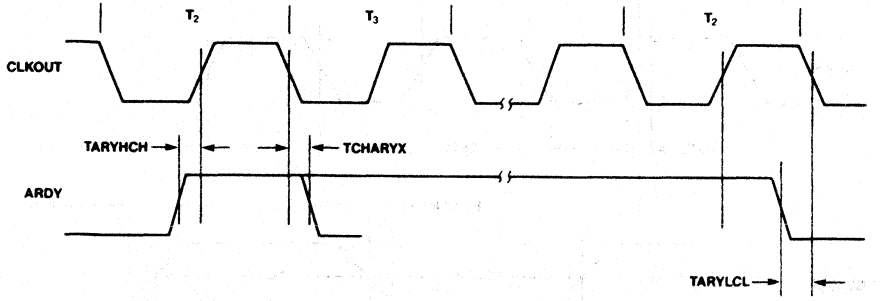
WF006240



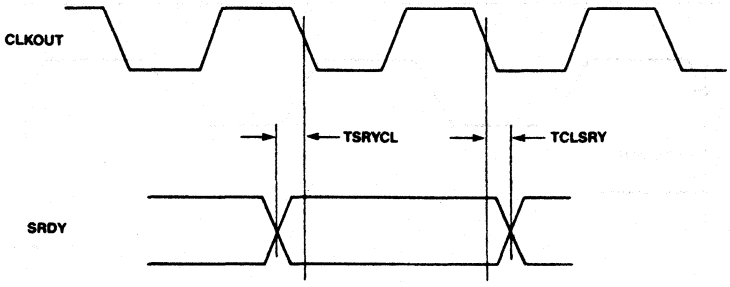
WF006250

SWITCHING WAVEFORMS (Continued)

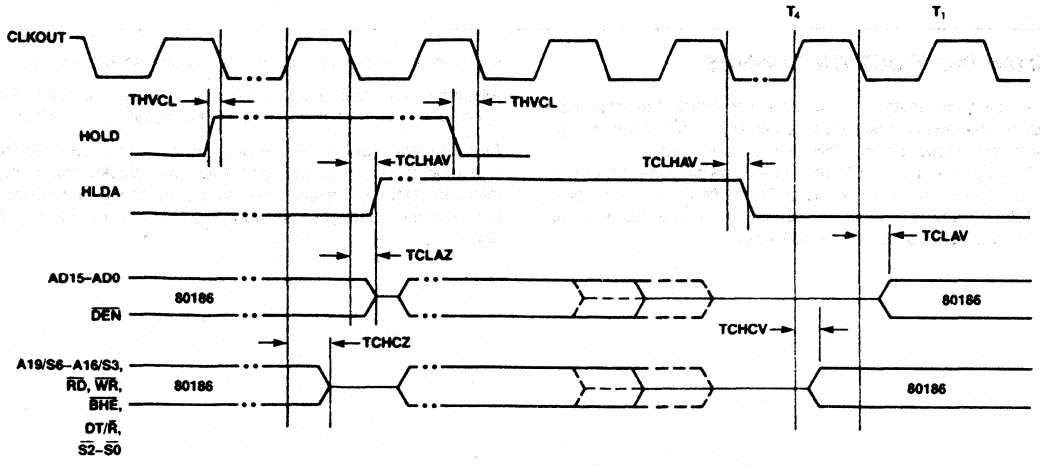
HOLD-HLDA TIMING



WF006260



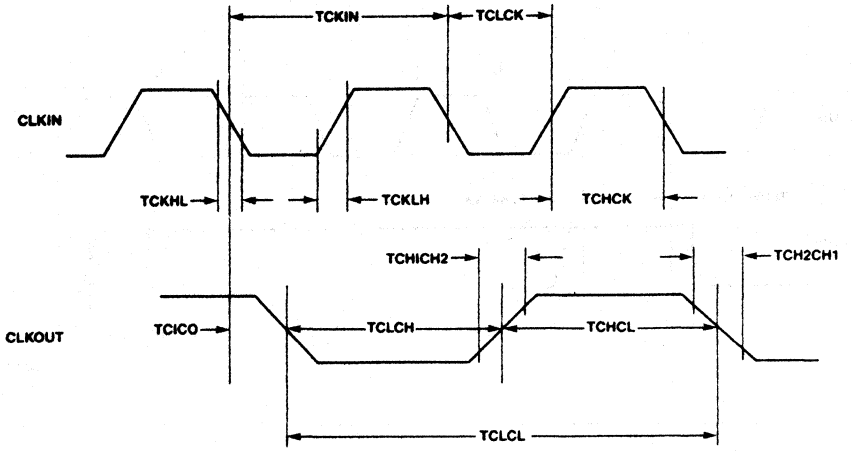
WF006270



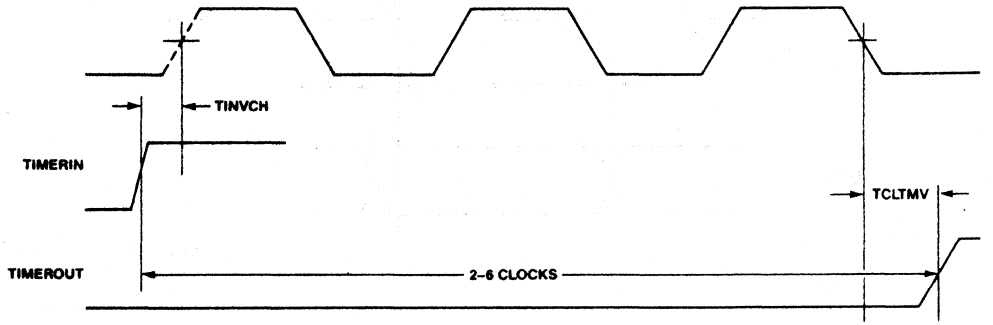
WF006280

SWITCHING WAVEFORMS (Continued)

TIMER ON 80186



WF006290



WF006300

80186 INSTRUCTION TIMINGS

The following instruction timings represent the minimum execution time in clock cycles for each instruction. The timings given are based on the following assumptions:

- The opcode, along with any data or displacement required for execution of a particular instruction, has been pre-fetched and resides in the queue at the time it is needed.
- No wait states or bus HOLDS occur.
- All word-data is located on even-address boundaries.

All jumps and calls include the time required to fetch the opcode of the next instruction at the destination address.

All instructions which involve memory reference can require one (and in some cases, two) additional clocks above the minimum timings shown. This is due to the asynchronous nature of the handshake between the BIU and the Execution unit.

INSTRUCTION SET SUMMARY

FUNCTION	FORMAT				Clock Cycles	Comments
DATA TRANSFER						
MOV = Move:						
Register to Register/Memory	1 0 0 0 1 0 0 w	mod reg r/m			2/12	
Register/memory to register	1 0 0 0 1 0 1 w	mod reg r/m			2/9	
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	12 – 13	8/16-bit
Immediate to register	1 0 1 1 w reg	data	data if w = 1		3 – 4	8/16-bit
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high		9	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high		8	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m			2/9	
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m			2/11	
PUSH = Push:						
Memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m			16	
Register	0 1 0 1 0 reg				10	
Segment register	0 0 0 reg 1 1 0				9	
Immediate	0 1 1 0 1 0 s 1	data	data if s = 0		10	
PUSHA = Push All	0 1 1 0 0 0 0 0				36	
POP = Pop:						
Memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m			20	
Register	0 1 0 1 1 reg				10	
Segment register	0 0 0 reg 1 1 1	(reg ≠ 01)			8	
POPA = Pop All	0 1 1 0 0 0 0 1				51	
XCHG = Exchange:						
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m			4/17	
Register with accumulator	1 0 0 1 0 reg				3	
IN = Input from:						
Fixed port	1 1 1 0 0 1 0 w	port			10	
Variable port	1 1 1 0 1 1 0 w				8	
OUT = Output to:						
Fixed port	1 1 1 0 0 1 1 w	port			9	
Variable port	1 1 1 0 1 1 1 w				7	
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1				11	
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m			6	
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m	(mod ≠ 11)		18	
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m	(mod ≠ 11)		18	
LAHF = Load AH with flags	1 0 0 1 1 1 1 1				2	
SAHF = Store AH into flags	1 0 0 1 1 1 1 0				3	
PUSHF = Push flags	1 0 0 1 1 1 0 0				9	
POPF = Pop flags	1 0 0 1 1 1 0 1				8	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT				Clock Cycles	Comments
ARITHMETIC						
ADD = Add:						
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m			3/10	
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s w = 01	4/16	
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1		3/4	8/16-bit
ADC = Add with carry:						
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m			3/10	
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s w = 01	4/16	
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1		3/4	8/16-bit
INC = Increment:						
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m			3/15	
Register	0 1 0 0 0 reg				3	
SUB = Subtract:						
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m			3/10	
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s w = 01	4/16	
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1		3/4	8/16-bit
SBB = Subtract with borrow:						
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m			3/10	
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s w = 01	4/16	
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1		3/4	8/16-bit
DEC = Decrement:						
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m			3/15	
Register	0 1 0 0 1 reg				3	
CMP = Compare:						
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m			3/10	
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m			3/10	
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s w = 01	3/10	
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1		3/4	8/16-bit
NEG = Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m			3	
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1				8	
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1				4	
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1				7	
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1				4	
MUL = Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m				
Register-Byte					26 – 28	
Register-Word					35 – 37	
Memory-Byte					32 – 34	
Memory-Word					41 – 43	
IMUL = Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m				
Register-Byte					25 – 28	
Register-Word					34 – 37	
Memory-Byte					31 – 34	
Memory-Word					40 – 43	
IMUL = Integer immediate multiply (signed)	0 1 1 0 1 0 s 1	mod reg r/m	data	data if s = 0	22 – 25/29 – 32	
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m				
Register-Byte					29	
Register-Word					38	
Memory-Byte					35	
Memory-Word					44	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC (Continued):			
IDIV = Integer divide (signed):	1 1 1 1 0 1 1 w | mod 1 1 1 r/m		
Register-Byte		44 – 52	
Register-Word		53 – 61	
Memory-Byte		50 – 58	
Memory-Word		59 – 67	
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	19	
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	15	
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	
CWD = Convert word to double word	1 0 0 1 1 0 0 1	4	
LOGIC			
Shift/Rotate Instructions:			
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2/15	
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m		
Register/Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m | count	5 + n/17 + n	

TTT	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

AND = **And:**					
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m			3/10
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1		3/4 8/16-bit
TEST = **And function to flags, no result:**					
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m			3/10
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	4/10
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1		3/4 8/16-bit
OR = **Or:**					
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m			3/10
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1		3/4 8/16-bit
XOR = **Exclusive or:**					
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m			3/10
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1		3/4 8/16-bit
NOT = Invert register/memory	1 1 1 1 0 1 1 w	mod 0 1 0 r/m			3
STRING MANIPULATION:					
MOVS = Move byte/word	1 0 1 0 0 1 0 w				8 + 8n
CMPS = Compare byte/word	1 0 1 0 0 1 1 w				5 + 22n
SCAS = Scan byte/word	1 0 1 0 1 1 1 w				5 + 15n
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w				6 + 11n
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w				6 + 9n
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w				8
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w				7

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
STRING MANIPULATION (Continued):			
Repeated by count in CX			
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	14	
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	22	
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	15	
LODS = Load string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 1 0 w	12	
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	10	
INS = Input string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 0 w	8 + 8n/14	Repeated/ Not Repeated
OUTS = Output string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 1 w	8 + 8n/14	Repeated/ Not Repeated
CONTROL TRANSFER			
CALL = **Call:**			
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	14	
Register memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	13/19	
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset | segment selector	23	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m | (mod ≠ 11)	38	
JMP = **Unconditional jump:**			
Short/long	1 1 1 0 1 0 1 1 | disp-low	13	
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	13	
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	11/17	
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset | segment selector	13	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m | (mod ≠ 11)	26	
RET = **Return from CALL:**			
Within segment	1 1 0 0 0 0 1 1	16	
Within seg adding immed to SP	1 1 0 0 0 0 1 0 | data-low | data-high	18	
Intersegment	1 1 0 0 1 0 1 1	22	
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0 | data-low | data-high	25	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT				Clock Cycles	Comments
CONTROL TRANSFER (Continued):						
JE/JZ = Jump on equal zero	0 1 1 1 0 1 0 0	disp			4/13	13 if JMP taken 4 if JMP not taken
JL/JNGE = Jump on less not greater or equal	0 1 1 1 1 1 0 0	disp			4/13	
JLE/JNG = Jump on less or equal not greater	0 1 1 1 1 1 1 0	disp			4/13	
JB/JNAE = Jump on below not above or equal	0 1 1 1 0 0 1 0	disp			4/13	
JBE/JNA = Jump on below or equal not above	0 1 1 1 0 1 1 0	disp			4/13	
JP/JPE = Jump on parity parity even	0 1 1 1 1 0 1 0	disp			4/13	
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp			4/13	
JS = Jump on sign	0 1 1 1 1 0 0 0	disp			4/13	
JNE/JNZ = Jump on not equal not zero	0 1 1 1 0 1 0 1	disp			4/13	
JNL/JGE = Jump on not less greater or equal	0 1 1 1 1 1 0 1	disp			4/13	
JNLE/JG = Jump on not less or equal greater	0 1 1 1 1 1 1 1	disp			4/13	
JNB/JAE = Jump on not below above or equal	0 1 1 1 0 0 1 1	disp			4/13	
JNBE/JA = Jump on not below or equal above	0 1 1 1 0 1 1 1	disp			4/13	
JNP/JPO = Jump on not par / par odd	0 1 1 1 1 0 1 1	disp			4/13	
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp			4/13	
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp			4/13	
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp			5/15	
LOOPZ/LOOPE = Loop while zero equal	1 1 1 0 0 0 0 1	disp			6/16	
LOOPNZ/LOOPNE = Loop while not zero equal	1 1 1 0 0 0 0 0	disp			6/16	
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp			16 5	JMP taken/ JMP not taken
ENTER = Enter Procedure	1 1 0 0 1 0 0 0	data-low	data-high	L		
L = 0					15	
L = 1					25	
L > 1					22 + 16(n - 1)	
LEAVE = Leave Procedure	1 1 0 0 1 0 0 1				8	
INT = **Interrupt:**						
Type specified	1 1 0 0 1 1 0 1	type			47	
Type 3	1 1 0 0 1 1 0 0				45	if INT. taken/
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0				48/4	if INT. not taken
IRET = Interrupt return	1 1 0 0 1 1 1 1				28	
BOUND = Detect value out of range	0 1 1 0 0 0 1 0	mod reg r/m			33 - 35	

Shaded areas indicate new 80186 instructions not available in 8086 or 8088 microprocessors.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
PROCESSOR CONTROL			
CLC = Clear carry	1 1 1 1 1 0 0 0	2	
CMC = Complement carry	1 1 1 1 0 1 0 1	2	
STC = Set carry	1 1 1 1 1 0 0 1	2	
CLD = Clear direction	1 1 1 1 1 1 0 0	2	
STD = Set direction	1 1 1 1 1 1 0 1	2	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	2	
STI = Set interrupt	1 1 1 1 1 0 1 1	2	
HLT = Halt	1 1 1 1 0 1 0 0	2	
WAIT = Wait	1 0 0 1 1 0 1 1	6	if $\overline{test}$ = 0
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	2	
ESC = Processor Extension Escape	1 0 0 1 1 T T T | mod LLL r/m (TTT LLL are opcode to processor extension)	6	

FOOTNOTES

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)
000 AX	000 AL
001 CX	001 CL
010 DX	010 DL
011 BX	011 BL
100 SP	100 AH
101 BP	101 CH
110 SI	110 DH
111 DI	111 BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed bythe DI register) are computed using the ES segment, which may not be overridden.

80286

High-Performance Microprocessor with
Memory Management and Protection

DISTINCTIVE CHARACTERISTICS

- High performance 6 and 8MHz processor (up to six times iAPX 86)
- Large address space
  - 16 megabytes physical
  - 1 gigabyte virtual memory per task
- Integrated memory management, four-level memory protection and support for virtual memory and operating systems
- Two iAPX 86 upward compatible operating modes
  - iAPX 86 real address mode
  - Protected virtual address mode
- High bandwidth bus interface (8 megabyte/sec)
- Range of clock rates
  - 8MHz 80286-8
  - 6MHz 80286-6

GENERAL DESCRIPTION

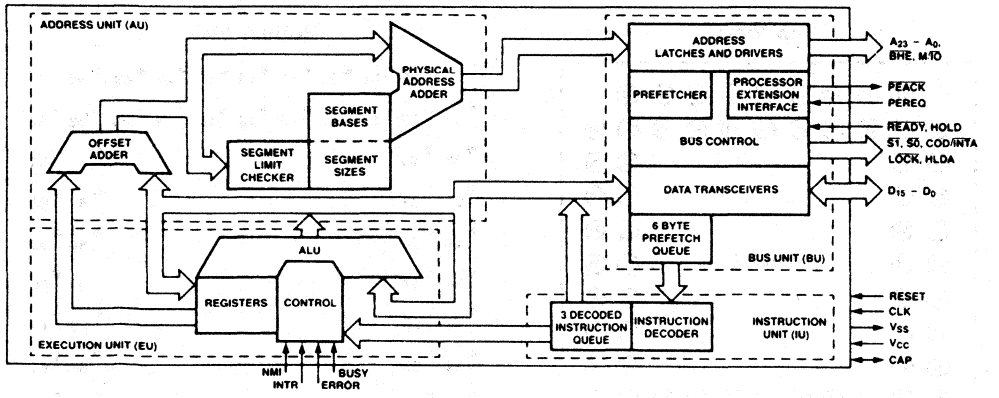
The 80286 is an advanced, high performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. The 80286 has built-in memory protection that supports operating system and task isolation as well as program and data privacy within tasks. An 8MHz 80286 provides up to six times greater throughput than the standard 5MHz 8086. The 80286 includes memory management capabilities that map up to 2^{30} bytes (one gigabyte) of virtual address space per task into 2^{24} bytes (16 megabytes) of physical memory.

The 80286 is upward compatible with iAPX 86 and 88 software. Using iAPX 86 real address mode, the 80286 is object code compatible with existing iAPX 86, 88 software.

In protected virtual address mode, the 80286 is source code compatible with iAPX 86, 88 software and may require upgrading to use virtual addresses supported by the 80286's integrated memory management and protection mechanism. Both modes operate at full 80286 performance and execute a superset of the iAPX 86 and 88 instructions.

The 80286 provides special operations to support the efficient implementation and execution of operating systems. For example, one instruction can end execution of one task, save it state, switch to a new task, load its state, and start execution of the new task. The 80286 also supports virtual memory systems by providing a segment-not-present exception and restartable instructions.

BLOCK DIAGRAM



BD003960

Figure 1. 80286 Internal Block Diagram

03552B

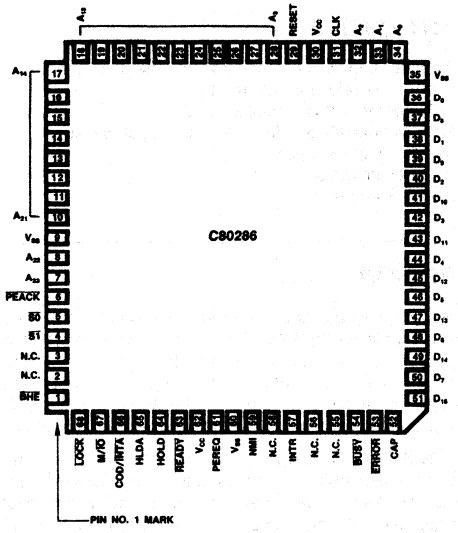
Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM

Ceramic Leadless Chip Carrier

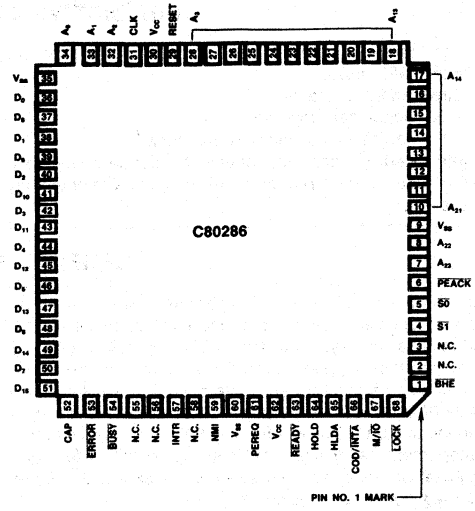
68-LCC

Top View



CD005612

Bottom View



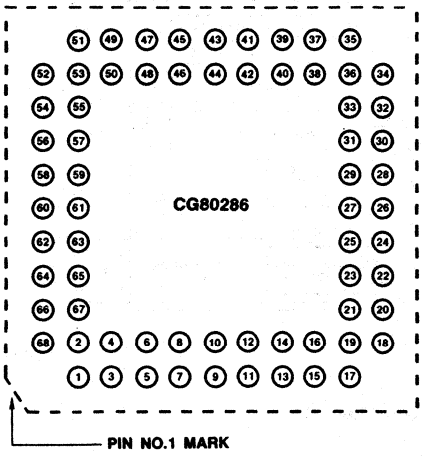
CD005901

There are no electrical connections on the bottom of this package.

68 Pin Grid Array Package

68-PGA

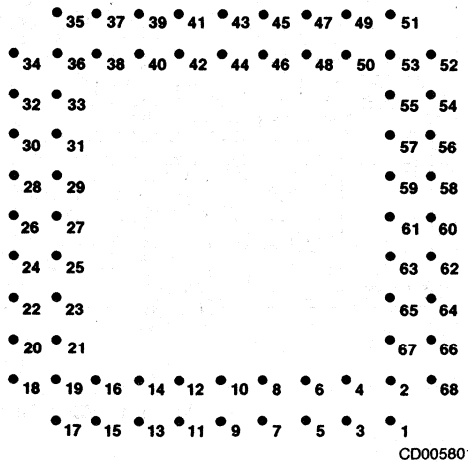
Top View



CD005793

Pins pointing away from viewer

Bottom View

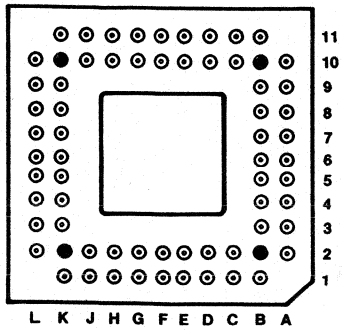


CD005801

Pins pointing toward viewer

68-PGA (continued)

Bottom View



CD005910

NAME	PAD	PIN
$\overline{BHE}$	1	B1
NC	2	B2
NC	3	C1
$\overline{ST}$	4	C2
$\overline{SO}$	5	D1
$\overline{PEACK}$	6	D2
A23	7	E1
A22	8	E2
VSS	9	F1
A21	10	F2
A20	11	G1
A19	12	G2
A18	13	H1
A17	14	H2
A16	15	J1
A15	16	J2
A14	17	K1
A13	18	L2
A12	19	K2
A11	20	L3
A10	21	K3
A9	22	L4
A8	23	K4
A7	24	L5
A6	25	K5
A5	26	L6
A4	27	K6
A3	28	L7
RESET	29	K7
VCC	30	L8
CLK	31	K8
A2	32	L9
A1	33	K9
A0	34	L10

NAME	PAD	PIN
VSS	35	K11
D0	36	K10
D8	37	J11
D1	38	J10
D9	39	H11
D2	40	H10
D10	41	G11
D3	42	G10
D11	43	F11
D4	44	F10
D12	45	E11
D5	46	E10
D13	47	D11
D6	48	D10
D14	49	C11
D7	50	C10
D15	51	B11
CAP	52	A10
$\overline{ERROR}$	53	B10
$\overline{BUSY}$	54	A9
NC	55	B9
NC	56	A8
INTR	57	B8
NC	58	A7
NMI	59	B7
VSS	60	A6
PEREQ	61	B6
VCC	62	A5
$\overline{READY}$	63	B5
HOLD	64	A4
HLDA	65	B4
COD/$\overline{INTA}$	66	A3
M/$\overline{IO}$	67	B3
$\overline{LOCK}$	68	A2

Plastic Leaded Chip Carrier
68-PLCC

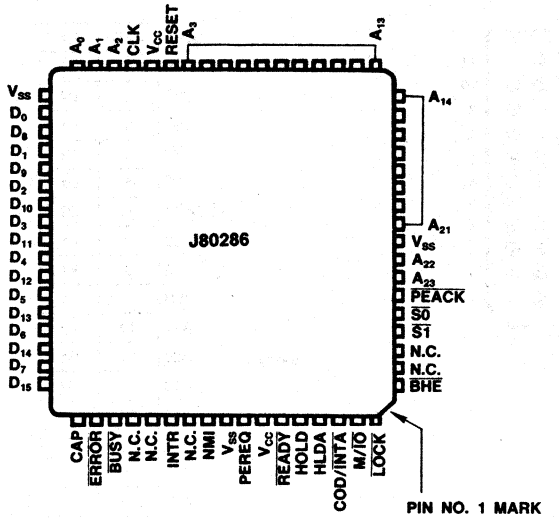
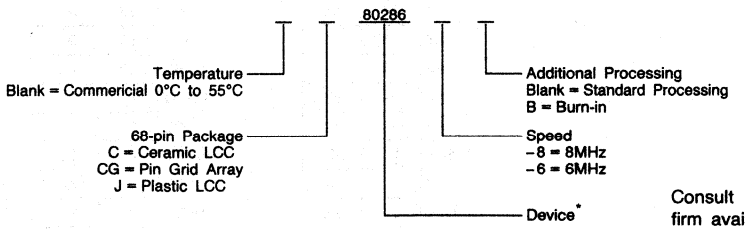


Figure 2. 80286 Pin Configurations

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations	
80286-8	C, CG
80286-6	

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
31	CLK	I	System Clock provides the fundamental timing for 80286 systems. It is a 16 MHz signal divided by two inside the 80286 to generate the 8 MHz processor clock. The internal divide-by-two circuitry can be synchronized to an external clock generator by a LOW-to-HIGH transition on the RESET input.
36-51	$D_0 - D_{15}$	I/O	Data Bus inputs data during memory, I/O, and interrupt acknowledge read cycles; outputs data during memory and I/O write cycles. The data bus is active HIGH and floats to 3-state OFF during bus hold acknowledge.
7, 8, 10-28, 32-34	$A_{23} - A_0$	O	Address Bus outputs physical memory and I/O port addresses. A_0 is LOW when data is to be transferred on pins D_{7-0}. $A_{23} - A_{16}$ are LOW during I/O transfers. The address bus is active HIGH and floats to 3-state OFF during bus hold acknowledge.
1	$\overline{BHE}$	O	Bus High Enable indicates transfer of data on the upper byte of the data bus D_{15-8}. Eight-bit oriented devices assigned to the upper byte of the data bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is active LOW and floats to 3-state OFF during bus hold acknowledge.
4, 5	$\overline{S1}$, $\overline{S0}$	O	Bus Cycle Status indicates initiation of a bus cycle and, along with M/$\overline{IO}$ and COD/$\overline{INTA}$, defines the type of bus cycle. The bus is in a T_S state whenever one or both are LOW. $\overline{S1}$ and $\overline{S0}$ are active LOW and float to 3-state OFF during bus hold acknowledge.
67	M/$\overline{IO}$	O	Memory/IO Select distinguishes memory access from I/O access. If HIGH during T_S, a memory cycle or a halt/shutdown cycle is in progress. If LOW, an I/O cycle or an interrupt acknowledge cycle is in progress. M/$\overline{IO}$ floats to 3-state OFF during bus hold acknowledge.
66	COD/$\overline{INTA}$	O	Code/Interrupt Acknowledge distinguishes instruction fetch cycles from memory data read cycles. Also distinguishes interrupt acknowledge cycles from I/O cycles. COD/$\overline{INTA}$ floats to 3-state OFF during bus hold acknowledge.
68	$\overline{LOCK}$	O	Bus Lock indicates that other system bus masters are not to gain control of the system bus following the current bus cycle. The $\overline{LOCK}$ signal may be activated explicitly by the "LOCK" instruction prefix or automatically by 80286 hardware during memory XCHG instructions, interrupt acknowledge, or descriptor table access. $\overline{LOCK}$ is active LOW and floats to 3-state OFF during bus hold acknowledge.
63	$\overline{READY}$	I	Bus Ready terminates a bus cycle. Bus cycles are extended without limit until terminated by $\overline{READY}$ LOW. $\overline{READY}$ is an active LOW synchronous input requiring set-up and hold times relative to the system clock be met for correct operation. $\overline{READY}$ is ignored during bus hold acknowledge.
64, 65	HOLD, HLDA	I O	Bus Hold Request and Hold Acknowledge control ownership of the 80286 local bus. The HOLD input allows another local bus master to request control of the local bus. When control is granted, the 80286 will float its bus drivers to 3-state OFF and then active HLDA, thus entering the bus hold acknowledge condition. The local bus will remain granted to the requesting master until HOLD becomes inactive which results in the 80286 deactivating HLDA and regaining control of the local bus. This terminates the bus hold acknowledge condition. HOLD may be asynchronous to the system clock. These signals are active HIGH.

$\overline{BHE}$ and A_0 Encodings

$\overline{BHE}$ Value	A_0 Value	Function
0	0	Word Transfer
0	1	Byte transfer on upper half of data bus (D_{15-8})
1	0	Byte transfer on lower half of data bus (D_{7-0})
1	1	Reserved

80286 Bus Cycle Status Definition

COD/$\overline{INTA}$	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Bus cycle initiated
0 (LOW)	0	0	0	Interrupt acknowledge
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	None; not a status cycle
0	1	0	0	IF $A_1 = 1$ then halt; else shutdown
0	1	0	1	Memory data read
0	1	1	0	Memory data write
0	1	1	1	None; not a status cycle
1 (HIGH)	0	0	0	Reserved
1	0	0	1	I/O read
1	0	1	0	I/O write
1	0	1	1	None; not a status cycle
1	1	0	0	Reserved
1	1	0	1	Memory instruction read
1	1	1	0	Reserved
1	1	1	1	None; not a status cycle

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
57	INTR	I	Interrupt Request requests the 80286 to suspend its current program execution and service a pending external request. Interrupt requests are masked whenever the interrupt enable bit in the flag word is cleared. When the 80286 responds to an interrupt request, it performs two interrupt acknowledge bus cycles to read an 8-bit interrupt vector that identifies the source of the interrupt. To assure program interruption, INTR must remain active until the first interrupt acknowledge cycle is completed. INTR is sampled at the beginning of each processor cycle and must be active HIGH at least two processor cycles before the current instruction ends in order to interrupt before the next instruction. INTR is level sensitive, active HIGH, and may be asynchronous to the system clock.
59	NMI	I	Non-maskable Interrupt Request interrupts the 80286 with an internally supplied vector value of 2. No interrupt acknowledge cycles are performed. The interrupt enable bit in the 80286 flag word does not affect this input. The NMI input is active HIGH, may be asynchronous to the system clock, and is edge triggered after internal synchronization. For proper recognition, the input must have been previously LOW for at least four system clock cycles and remain HIGH for at least four system clock cycles.
61, 6	PEREQ, $\overline{PEACK}$	I O	Processor Extension Operand Request and Acknowledge extended the memory management and protection capabilities of the 80286 to processor extensions. The PEREQ input requests the 80286 to perform a data operand transfer for a processor extension. The $\overline{PEACK}$ output signals the processor extension when the requested operand is being transferred. PEREQ is active HIGH and may be asynchronous to the system clock. $\overline{PEACK}$ is active LOW.
54, 53	$\overline{BUSY}$, $\overline{ERROR}$	I I	Processor Extension Busy and Error indicate the operating condition of a processor extension to the 80286. An active $\overline{BUSY}$ input stops 80286 program execution on WAIT and some ESC instructions until BUSY becomes inactive (HIGH). The 80286 may be interrupted while waiting for BUSY to become inactive. An active ERROR input causes the 80286 to perform a processor extension interrupt when executing WAIT or some ESC instructions. These inputs are active LOW and may be asynchronous to the system clock.
29	RESET	I	System Reset clears the internal logic of the 80286 and is active HIGH. The 80286 may be reinitialized at any time with a LOW-to-HIGH transition on RESET which remains active for more than 16 system clock cycles. During RESET active, the output pins of the 80286 enter the state shown below: **80286 Pin State During Reset** **Pin Value** / **Pin Names** 1 (HIGH) / $\overline{S0}$, $\overline{ST}$, $\overline{PEACK}$, A23 – A0, $\overline{BHE}$, $\overline{LOCK}$ 0 (LOW) / M/$\overline{IO}$, COD/$\overline{INTA}$, HLDA 3-state OFF / $D_{15} - D_0$ Operation of the 80286 begins after a HIGH-to-LOW transition on RESET. The HIGH-to-LOW transition of RESET must be synchronous to the system clock. Approximately 50 system clock cycles are required by the 80286 for internal initializations before the first bus cycle to fetch code from the power-on execution address is performed. A LOW-to-HIGH transition of RESET synchronous to the system clock, will begin a new processor cycle at the next HIGH-to-LOW transition of the system clock. The LOW-to-HIGH transition of RESET may be asynchronous to the system clock; however, in this case it cannot be predetermined which phase of the processor clock will occur during the next system clock period. Synchronous LOW-to-HIGH transitions of RESET are only required for systems where the processor clock must be phase synchronous to another clock.
9, 35, 60	V_{SS}	I	System Ground: 0 VOLTS.
30, 62	V_{CC}	I	System Power: +5 Volt Power Supply.
52	CAP	I	Substrate Filter Capacitor: a 0.047μf ±20% 12V capacitor must be connected between this pin and ground. This capacitor filters the output of the internal substrate bias generator. A maximum DC leakage current of 1 μa is allowed through the capacitor. For correct operation of the 80286, the substrate bias generator must charge this capacitor to its operating voltage. The capacitor charge-up time is 5 milliseconds (max.) after V_{CC} and CLK reach their specified AC and DC parameters. RESET may be applied to prevent spurious activity by the CPU during this time. After this time, the 80286 processor clock can be phase synchronized to another clock by pulsing RESET LOW synchronous to the system clock.

DETAILED DESCRIPTION

Introduction

The 80286 is an advanced, high-performance microprocessor with specially optimized capabilities for multiple user and multi-tasking systems. Depending on the application, the 80286's performance is up to six times faster than the standard 5 MHz 8086's, while providing complete upward software compatibility with AMD's iAPX 86, 88, and 186 family of CPU's.

The 80286 operates in two modes: iAPX 86 real address mode and protected virtual address mode. Both modes execute a superset of the iAPX 86 and 88 instruction set.

In iAPX 86 real address mode programs use real addresses with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

The following pages describe first, the base 80286 architecture common to both modes; second, iAPX 86 real address mode; and third, protected mode.

80286 Base Architecture

The iAPX 86, 88, 186, and 286 CPU family all contain the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPU's.

Register Set

The 80286 base architecture has fifteen registers as shown in Figure 3. These registers are grouped into the following four categories:

General Registers: Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used either in their entirety as 16-bit words or split into pairs of separate 8-bit registers.

Segment Registers: Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers: Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode determines the specific registers used for operand address calculations.

Status and Control Registers: Three 16-bit special purpose registers record or control certain aspects of the 80286 processor state. These include the Instruction Pointer, which contains the offset address of the next sequential instruction to be executed.

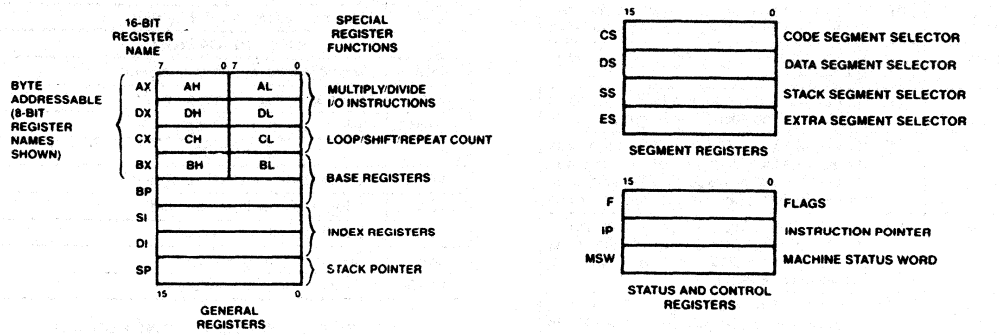


Figure 3. Register Set

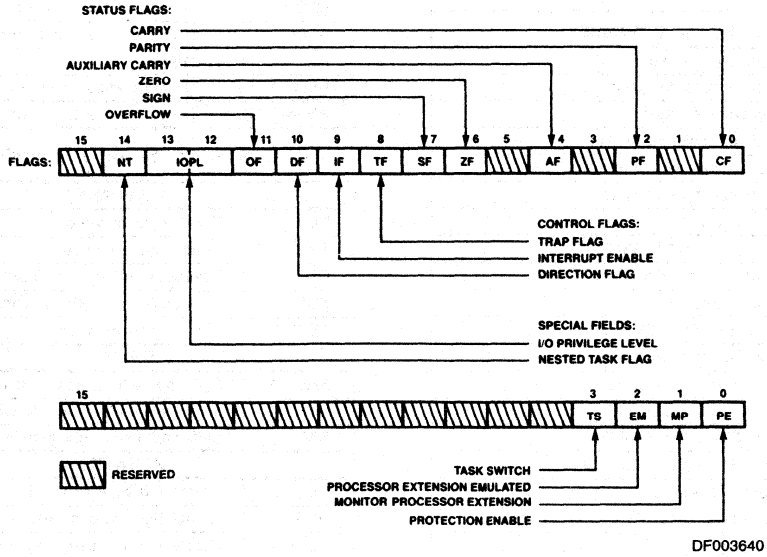


Figure 3a. Status and Control Register Bit Functions

Flags Word Description

The Flags word (Flags) records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80286 within a given operating mode (bits 8 and 9). Flags is a 16-bit register. The function of the flag bits is given in Table 2.

Table 2. Flags Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag — Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag — Set if low-order 8 bits of result contain an even number of 1-bits; cleared otherwise
4	AF	Set on carry from or borrow to the low-order four bits of AL; cleared otherwise
6	ZF	Zero Flag — Set if result is zero; cleared otherwise
7	SF	Sign Flag — Set equal to high-order bit of result (0 if positive, 1 if negative)
11	OF	Overflow Flag — Set if result is a too-large large positive number or a too-small negative number (excluding sign-bit) to fit in destination operand; cleared otherwise
8	TF	Single Step Flag — Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt.
9	IF	Interrupt-Enable Flag — When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location
10	DF	Direction Flag — Causes string instructions to auto decrement the appropriate index registers when set. Clearing DF causes auto increment.

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

Figure 4a. Data Transfer Instructions

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword

Figure 4b. Arithmetic Instructions

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

Figure 4c. String Instructions

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

Figure 4d. Shift/Rotate/Logical Instructions

CONDITIONAL TRANSFERS		UNCONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal	CALL	Call procedure
JAE/JNB	Jump if above or equal/not below	RET	Return from procedure
JB/JNAE	Jump if below/not above nor equal	JMP	Jump
JBE/JNA	Jump if below or equal/not above		
JC	Jump if carry	**ITERATION CONTROLS**	
JE/JZ	Jump if equal/zero		
JG/JNLE	Jump if greater/not less nor equal	LOOP	Loop
JGE/JNL	Jump if greater or equal/not less	LOOPE/LOOPZ	Loop if equal/zero
JL/JNGE	Jump if less/not greater nor equal	LOOPNE/LOOPNZ	Loop if not equal/not zero
JLE/JNG	Jump if less or equal/not greater	JCXZ	Jump if register CX = 0
JNC	Jump if not carry		
JNE/JNZ	Jump if not equal/not zero	**INTERRUPTS**	
JNO	Jump if not overflow		
JNP/JPO	Jump if not parity/parity odd	INT	Interrupt
JNS	Jump if not sign	INTO	Interrupt if overflow
JO	Jump if overflow	IRET	Interrupt return
JP/JPE	Jump if parity/parity even		
JS	Jump if sign		

Figure 4e. Program Transfer Instructions

FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{\text{BUSY}}$ not active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
EXECUTION ENVIRONMENT CONTROL	
LMSW	Load machine status word
SMSW	Store machine status word

Figure 4f. Processor Control Instructions

ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

Figure 4g. High Level Instructions

Memory Organization

Memory is organized as sets of variable length segments. Each segment is a linear contiguous sequence of up to 64K(2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit segment selector and a 16-bit offset. The segment selector indicates the desired segment in memory. The offset component indicates the desired byte address within the segment.

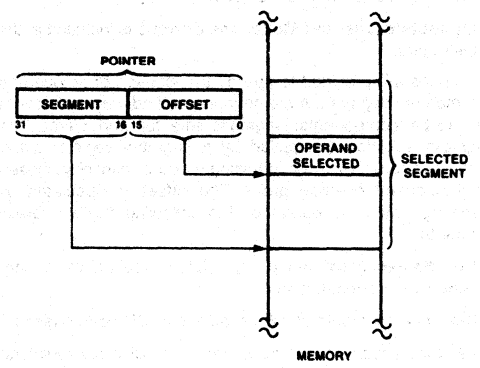


Figure 5. Two Component Address

Table 3. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Automatic with instruction prefetch
Stack	Stack (SS)	All stack pushes and pops. Any memory reference which uses BP as a base register.
Local Data	Data (DS)	All data references except when relative to stack or string destination
External (Global) Data	Extra (ES)	Alternate data segment and destination of string operation

All instructions that address operands in memory must specify the segment and the offset. For speed and compact instruction encoding, segment selectors are usually stored in the high speed segment registers. An instruction need specify only the desired segment register and an offset to address a memory operand.

Most instructions need not explicitly specify which segment register is used. The correct segment register is automatically chosen according to the rules of Table 3. These rules follow the way programs are written (see Figure 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs. To access operands that do not reside in one of the four immediately available segments, either a full 32-bit pointer can be used or a new segment selector must be loaded.

Addressing Modes

The 80286 provides a total of eight addressing modes for instructions to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8 or 16-bit general registers.

Immediate Operand Mode. The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: segment selector and offset. The segment selector is supplied by a segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset is calculated by summing any combination of the following three address elements:

the displacement (an 8 or 16-bit immediate value contained in the instruction)

the base (contents of either the BX or BP base registers)

the index (contents of either the SI or DI index registers)

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, here described.

Direct Mode: The operand's offset is contained in the instruction as an 8 or 16-bit displacement element.

Register Indirect Mode: The operand's offset is in one of the registers SI, DI, BX, or BP.

Based Mode: The operand's offset is the sum of an 8 or 16-bit displacement and the contents of a base register (BX or BP).

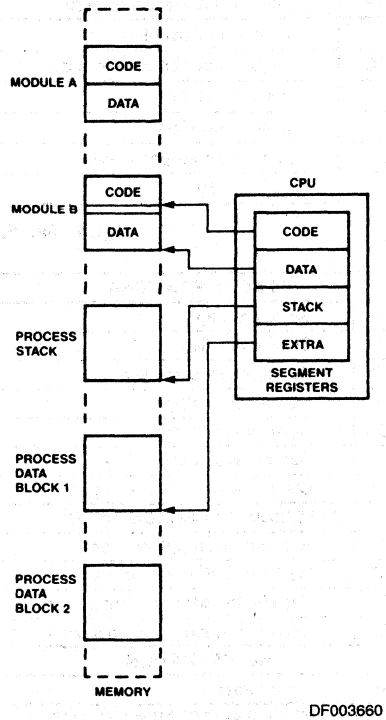


Figure 6. Segmented Memory Helps Structure Software

Indexed Mode: The operand's offset is the sum of an 8 or 16-bit displacement and the contents of an index register (SI or DI).

Based Indexed Mode: The operand's offset is the sum of the contents of a base register and an index register.

Based Indexed Mode with Displacement: The operand's offset is the sum of a base register's contents, an index register's contents, and an 8 or 16-bit displacement.

Data Types

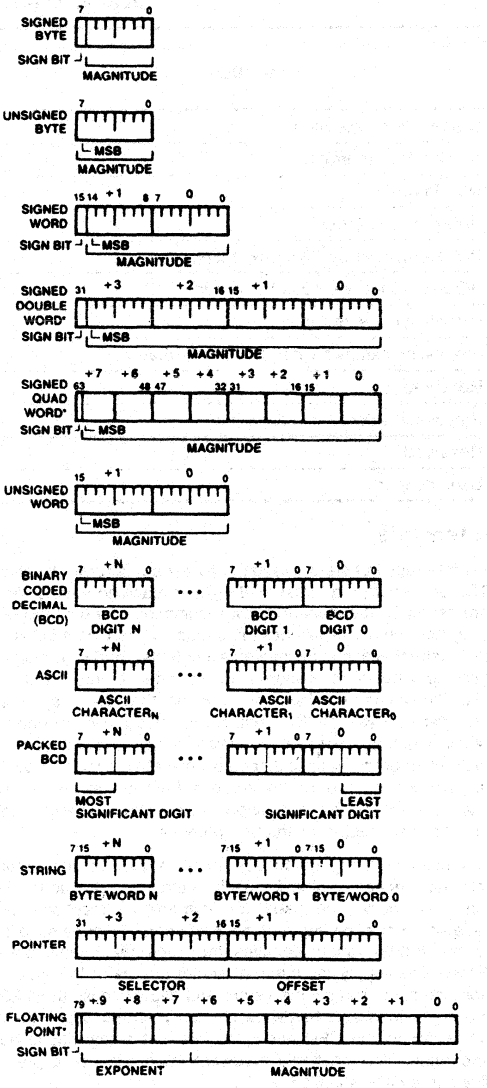
The 80286 directly supports the following data types:

Integer:	A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32 and 64-bit integers are supported using the 80287 Numeric Data Processor.
Ordinal:	An unsigned binary numeric value contained in an 8-bit byte or 16-bit word.
Pointer:	A 32-bit quantity, composed of a segment selector component and an offset component. Each component is a 16-bit word.
String:	A contiguous sequence of bytes or words. A string may contain from 1 byte to 64K bytes.
ASCII:	A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
BCD:	A byte (unpacked) representation of the decimal digits 0 – 9.
Packed BCD:	A byte (packed) representation of two decimal digits 0 – 9 storing one digit in each nibble of the byte.
Floating Point:	A signed 32, 64, or 80-bit real number representation. (Floating point operands are supported using the iAPX 287 Numeric Processor configuration.)

Figure 7 graphically represents the data types supported by the 80286.

I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. I/O instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. 8-bit port addresses are zero extended such that $A_{15} - A_8$ are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.



*Support by iAPX 286/287 Numeric Data Processor Configuration

Figure 7. 80286 Supported Data Types

Table 4. Interrupt Vector Assignments

Function	Interrupt Number	Related Instructions	Return Address Before Instruction Causing Exception?
Divide error exception	0	DIV, IDIV	Yes
Single step interrupt	1	All	
NMI interrupt	2	All	
Breakpoint interrupt	3	INT	
INTO detected overflow exception	4	INTO	No
BOUND range exceeded exception	5	BOUND	Yes
Invalid opcode exception	6	Any undefined opcode	Yes
Processor extension not available exception	7	ESC or WAIT	Yes
Reserved	8-15		
Processor extension error input	16	ESC or WAIT	
Reserved	17-31		
User defined	32-255		

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Flags) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable. Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. The return address from an exception will always point at the instruction causing the exception and include any leading instruction prefixes.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0 – 31, some of which are used for instruction exceptions, are reserved. For each interrupt, an 8-bit vector must be supplied to the 80286 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Maskable Interrupt (INTR)

The 80286 provides a maskable hardware interrupt request pin, INTR. Software enables this input by setting the interrupt flag bit (IF) in the flag word. All 224 user-defined interrupt sources can share this input, yet they can retain separate interrupt handlers. An 8-bit vector read by the CPU during the interrupt acknowledge sequence (discussed in System Interface section) identifies the source of the interrupt.

Further maskable interrupts are disabled while servicing an interrupt by resetting the IF but as part of the response to an interrupt or exception. The saved flag word will reflect the enable status of the processor prior to the interrupt. Until the flag word is restored to the flag register, the interrupt flag will be zero unless specifically set. The interrupt return instruction includes restoring the flag word, thereby restoring the original status of IF.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt input (NMI) is also provided. NMI has higher priority than INTR. A typical use of NMI would be to activate a power failure routine. The activation of this input causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed.

While executing the NMI servicing procedure, the 80286 will not service further NMI requests, INTR requests, or the processor extension segment overrun interrupt until an interrupt return (IRET) instruction is executed or the CPU is reset. If NMI occurs while currently servicing an NMI, its presence will be saved for servicing after executing the first IRET instruction. IF is cleared at the beginning of an NMI interrupt to inhibit INTR interrupts.

Single Step Interrupt

The 80286 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single step interrupt and is controlled by the single step flag bit (TF) in the flag word. Once this bit is set, an internal single step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single stepped.

Interrupt Priorities

When simultaneous interrupt requests occur, they are processed in a fixed order as shown in Table 5. Interrupt processing involves saving the flags, return address, and setting CS:IP to point at the first instruction of the interrupt handler. If other interrupts remain enabled, they are processed before the first instruction of the current interrupt handler is executed. The last interrupt processed is therefore the first one serviced.

Table 5. Interrupt Processing Order

Order	Interrupt
1	INT instruction or exception
2	Single step
3	NMI
4	Processor extension segment overrun
5	INTR

Initialization and Processor Reset

Processor initialization or start up is accomplished by driving the RESET input pin HIGH. RESET forces the 80286 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as RESET is active. After RESET becomes inactive and an internal processing interval elapses, the 80286 begins execution in real address mode with the instruction at physical location FFFFF0(H). RESET also sets some registers to predefined values as shown in Table 6.

Table 6. 80286 Initial Register State after RESET

Flag word	0002(H)
Machine Status Word	FFF0(H)
Instruction pointer	FFF0(H)
Code segment	F000(H)
Data segment	0000(H)
Extra segment	0000(H)
Stack segment	0000(H)

Machine Status Word Description

The machine status word (MSW) records when a task switch takes place and controls the operating mode of the 80286. It is a 16-bit register of which the lower four bits are used. One bit places the CPU into protected mode, while the other three bits, as shown in Table 7, control the processor extension interface. After RESET, this register contains FFF0(H) which places the 80286 in iAPX 86 real address mode.

Table 7. MSW Bit Functions

Bit Position	Name	Function
0	PE	Protected mode Enable places the 80286 into protected mode and cannot be cleared except by RESET.
1	MP	Monitor Processor extension allows WAIT instructions to cause a processor extension not present exception (number 7).
2	EM	Emulate processor extension causes a processor extension not present exception (number 7) on ESC instructions to allow emulating a processor extension.
3	TS	Task Switched indicates the next instruction using a processor extension will cause exception 7, allowing software to test whether the current processor extension context belongs to the current task.

The LMSW and SMSW instructions can load and store the MSW in real address mode. The recommended use of TS, EM, and MP is shown in Table 8.

Halt

The HLT instruction stops program execution and prevents the CPU from using the local bus until restarted. Either NMI, INTR with IF = 1, or RESET will force the 80286 out of halt. If interrupted, the saved CS:IP will point to the next instruction after the HLT.

iAPX 86 Real Address Mode

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in real address mode. In real address mode the 80286 is object code compatible with 8086 and 8088 software. The real address mode architecture (registers and addressing modes) is exactly as described in the 80286 Base Architecture section.

Table 8. Recommended MSW Encodings For Processor Extension Control

TS	MP	EM	Recommended Use	Instructions Causing Exception
0	0	0	iAPX 86 real address mode only. Initial encoding after RESET. 80286 operation is identical to iAPX 86, 88.	None
0	0	1	No processor extension is available. Software will emulate its function.	ESC
1	0	1	No processor extension is available. Software will emulate its function. The current processor extension context may belong to another task.	ESC
0	1	0	A processor extension exists.	None
1	1	0	A processor extension exists. The current processor extension context may belong to another task. The exception on WAIT allows software to test for an error pending from a previous processor extension operation.	ESC or WAIT

Memory Size

Physical memory is a contiguous array of up to 1,048,576 bytes (one megabyte) addressed by pins A_0 through A_{19} and $\overline{BHE}$. A_{20} through A_{23} are ignored.

Memory Addressing

In real address mode the processor generates 20-bit physical addresses directly from a 20-bit segment base address and a 16-bit offset.

The selector portion of a pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower four bits of the 20-bit segment address are always zero. Segment addresses, therefore, begin on multiples of 16 bytes. See Figure 8 for a graphic representation of address formation.

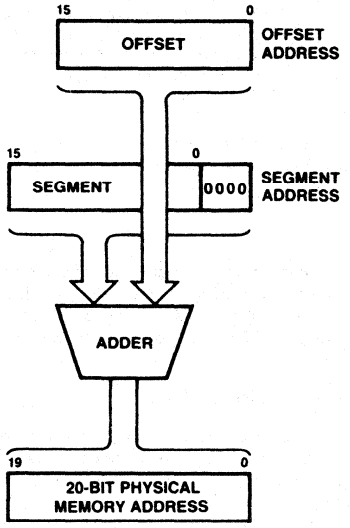


Figure 8. iAPX 86 Real Address Mode Address Calculation

All segments in real address mode are 64K bytes in size and may be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (e.g. a word with its low order byte at offset FFFF(H) and its high order byte at offset 0000(H)). If, in real address mode, the information contained in a segment does not use the full 64K bytes, the unused end of the segment may be overlayed by another segment to reduce physical memory requirements.

Reserved Memory Locations

The 80286 reserves two fixed areas of memory in real address mode (see Figure 9): system initialization area and interrupt table area. Locations from addresses FFFF0(H) through FFFFF(H) are reserved for system initialization. Initial execution begins at location FFFF0(H). Locations 00000(H) through 003FF(H) are reserved for interrupt vectors.

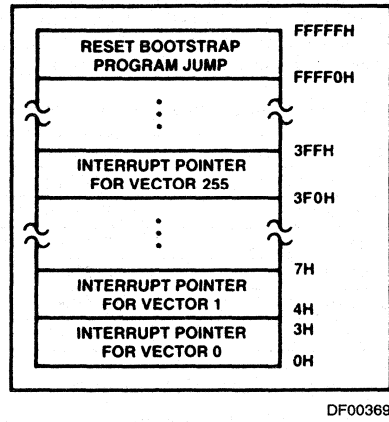


Figure 9. iAPX 86 Real Address Mode Initially Reserved Memory Locations

Table 9. Real Address Mode Addressing Interrupts

Function	Interrupt Number	Related Instructions	Return Address Before Instruction?
Interrupt tabie limit too small exception	8	INT vector is not within table limit	Yes
Processor extension segment overrun interrupt	9	ESC with memory operand extending beyond offset FFFF(H)	No
Segment overrun exception	13	Word memory reference with offset = FFFF(H) or an attempt to execute past the end of a segment	Yes

Interrupts

Table 9 shows the interrupt vectors reserved for exceptions and interrupts which indicate an addressing error. The exceptions leave the CPU in the state existing before attempting to execute the failing instruction (except for PUSH, POP, PUSHA, or POPA). Refer to the next section on protected mode initialization for a discussion on exception 8.

Protected Mode Initialization

To prepare the 80286 for protected mode, the LIDT instruction is used to load the 24-bit interrupt table base and 16-bit limit for the protected mode interrupt table. This instruction can also set a base and limit for the interrupt vector table in real address mode. After reset, the interrupt table base is initialized to 000000(H) and its size set to 03FF(H). These values are compatible with iAPX 86, 88 software. LIDT should only be executed in preparation for the protected mode.

Shutdown

Shutdown occurs when a severe error is detected that prevents further instruction processing by the CPU. Shutdown and halt are externally signalled via a halt bus operation. They can be distinguished by A_1 HIGH for halt and A_1 LOW for shutdown. In real address mode, shutdown can occur under two conditions:

- Exceptions 8 or 13 happen and the IDT limit does not include the interrupt vector.
- A CALL, INT, or POP instruction attempts to wrap around the stack segment when SP is not even.

An NMI input can bring the CPU out of shutdown if the IDT limit is at least 000F(H) and SP is greater than 0005(H); otherwise, shutdown can only be exited via the RESET input.

Protected Virtual Address Mode

The 80286 executes a fully upward-compatible superset of the 8086 instruction set in protected virtual address mode (protected mode). Protected mode also provides memory management and protection mechanisms and associated instructions.

The 80286 enters protected virtual address mode from real address mode by setting the PE (Protection Enable) bit of the machine status word with the Load Machine Status Word (LMSW) instruction. Protected mode offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

All registers, instructions, and addressing modes described in the 80286 Base Architecture section remain the same. Programs for the iAPX 86, 88, 186, and real address mode 80286 can be run in protected mdoe; however, embedded constants for segment selectors are different.

Memory Size

The protected mode 80286 provides a 1 gigabyte virtual address space per task mapped into a 16 megabyte physical address space defined by the address pin $A_{23} - A_0$ and $\overline{BHE}$. The virtual address space may be larger than the physical address space since any use of an address that does not map to a physical memory location will cause a restartable exception.

Memory Addressing

As in real address mode, protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector, however, specifies an index into a memory resident table rather than the upper 16-bits of a real memory address. The 24-bit base address of the desired segment is obtained from the tables in memory. The 16-bit offset is added to the segment base address to form the physical address as shown in Figure 10. The tables are automatically referenced by the CPU whenever a segment register is loaded with a selector. All 80286 instructions which load a segment register will reference the memory based tables without additional software. The memory based tables contain 8 byte values called descriptors.

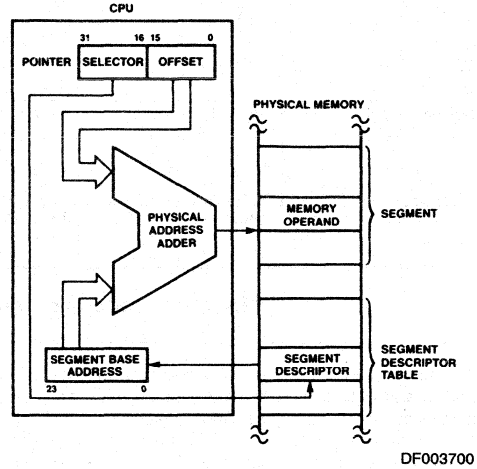


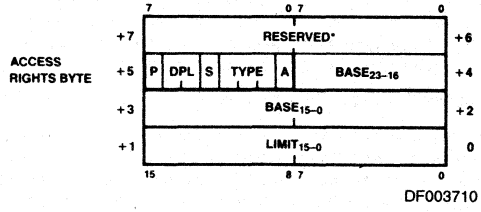
Figure 10. Protected Mode Memory Addressing

Descriptors

Descriptors define the use of memory. Special types of descriptors also define new functions for transfer of control and task switching. The 80286 has segment descriptors for code, stack and data segments, and system control descriptors for special system data segments and control transfer operations. Descriptor accesses are performed as locked bus operations to assure descriptor integrity in multi-processor systems.

Code and Data Segment Descriptors

Besides segment base addresses, code and data descriptors contain other segment attributes including segment size (1 to 64K bytes), access rights (read only, read/write, execute only, and execute/read), and presence in memory (for virtual memory systems)(See Figure 11). Any segment usage violating a segment attribute indicated by the segment descriptor will prevent the memory cycle and cause an exception or interrupt.



*Must be set to 0 for compatability with iAPX 386.

Access Rights Byte Definition

	Bit Position	Name	Function	
	7	Present (P)	P = 1 Segment is mapped into physical memory.	
			P = 0 No mapping to physical memory exists; base and limit are not used.	
	6-5	Descriptor Privilege Level (DPL)	Segment privilege attribute used in privilege tests.	
	4	Segment Descriptor (S)	S = 1 Code or Data segment descriptor S = 0 Non-segment descriptor	
Type Field Definition	3	Executable (E)	E = 0 Data segment descriptor type is:	**Data Segment**
	2	Expansion Direction (ED)	ED = 0 Grow up segment, offsets must be ⩽ limit.	
			ED = 1 Grow down segment, offsets must be > limit.	
	1	Writable (W)	W = 0 Data segment may not be written into.	
			W = 1 Data segment may be written into.	
	3	Executable (E)	E = 1 Code Segment Descriptor type is:	**Code Segment**
	2	Conforming (C)	C = 1 Code segment may only be executed when CPL ⩾ DPL.	
	1	Readable (R)	R = 0 Code segment may not be read.	
			R = 1 Code segment may be read.	
	0	Accessed (A)	A = 0 Segment has not been accessed.	
			A = 1 Segment selector has been loaded into segment register or used by selector test instructions.	

Figure 11. Code and Data Segment Descriptors

Code and data are stored in two types of segments: code segments and data segments. Both types are identified and defined by segment descriptors. Code segments are identified by the executable (E) bit set to 1 in the descriptor access rights byte. The access rights byte of both code and data segment descriptor types have three fields in common: present (P) bit, Descriptor Privilege Level (DPL), and accessed (A) bit. If P = 0, any attempted use of this segment will cause a not-present exception. DPL specifies the privilege level of the segment descriptor. DPL controls when the descriptor may be used by a task (refer to privilege discussion). The A bit shows whether the segment has been previously accessed for usage profiling, a necessity for virtual memory systems. The CPU will always set this bit when accessing the descriptor.

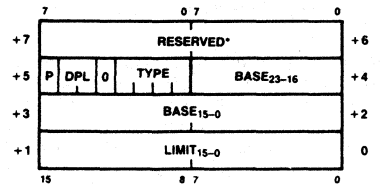
Data segments (S = 1, E = 0) may be either read-only or read-write as controlled by the W bit of the access rights byte. Read-only (W = 0) data segments may not be written into. Data segments may grow in two directions, as determined by the Expansion Direction (ED) bit: upwards (ED = 0) for data segments, and downwards (ED = 1) for a segment containing a stack. The limit field for a data segment descriptor is interpreted differently depending on the ED bit (see Figure 11).

A code segment (S = 1, E = 1) may be execute-only or execute/read as determined by the Readable (R) bit. Code segments may never be written into and execute-only code segments (R = 0) may not be read. A code segment may also have an attribute called conforming (C). A conforming code segment may be shared by programs that execute at different privilege levels. The DPL of a conforming code segment defines the range of privilege levels at which the segment may be executed (refer to privilege discussion).

System Control Descriptors

In addition to code and data segment descriptors, the protected mode 80286 defines system control descriptors. These descriptors define special system data segments and control transfer mechanisms in the protected environment. The special system data segment descriptors define segments which contain tables of descriptors (Local Descriptor Table Descriptor) and segments which contain the execution state of a task (Task State Segment Descriptor).

The control transfer descriptors are call gates, task gates, interrupt gates and trap gates. Gates provide a level of indirection between the source and destination of the control transfer. This indirection allows the CPU to automatically perform protection checks and control the entry point of the destination. Call gates are used to change privilege levels (see Privilege); task gates are used to perform a task switch; and interrupt and trap gates are used to specify interrupt service routines. The interrupt gate disables interrupts (resets IF) while the trap gate does not.



TB000088

*Must be set to 0 for compatability with iAPX 386.

System Segment Descriptor Fields

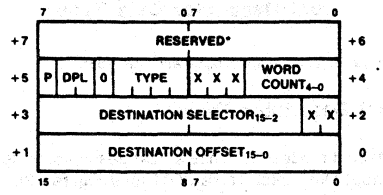
Name	Value	Description
TYPE	1	Available Task State Segment
	2	Local Descriptor Table Descriptor
	3	Busy Task State Segment
P	0	Descriptor contents are not valid
	1	Descriptor contents are valid
DPL	0 – 3	Descriptor Privilege Level
BASE	24-bit number	Base Address of special system data segment in real memory
LIMIT	16-bit number	Offset of last byte in segment

Figure 12. System Segment Format

Figure 12 gives the formats for the special system data segment descriptors. The descriptors contain a 24-bit base address of the segment and a 16-bit limit. The access byte defines the type of descriptor, its state and privilege level. The descriptor contents are valid and the segment is in physical memory if P = 1. If P = 0, the segment is not valid. The DPL field is only used in Task State Segment descriptors and indicates the privilege level at which the descriptor may be used (see Privilege). Since the Local Descriptor Table descriptor may only be used by a special privileged instruction, the DPL field is not used. Bit 4 of the access byte is 0 to indicate that it is a system control descriptor. The type field specifies the descriptor type as indicated in Figure 12.

Figure 13 shows the format of the gate descriptors. The descriptor contains a destination pointer that points to the descriptor of the target segment and the entry point offset. The destination selector in an interrupt gate, trap gate, and call gate must refer to a code segment descriptor. These gate descriptors contain the entry point to prevent a program from constructing and using an illegal entry point. Task gates may only refer to a task state segment. Since task gates invoke a task switch, the destination offset is not used in the task gate.

Exception 13 is generated when the gate is used if a destination selector does not refer to the correct descriptor type. The word count field is used in the call gate descriptor to indicate the number of parameters (0 – 31 words) to be automatically copied from the caller's stack to the stack of the called routine when a control transfer changes privilege levels. The word count field is not used by any other gate descriptor.



*Must be set to 0 for compatability with iAPX 386.

Gate Descriptor Fields

Name	Value	Description
TYPE	4 5 6 7	–Call Gate –Task Gate –Interrupt Gate –Trap Gate
P	0 1	–Descriptor Contents are not valid –Descriptor Contents are valid
DPL	0 – 3	Descriptor Privilege Level
WORD COUNT	0 – 31	Number of words to copy from callers stack to called procedures stack. Only used with call gate.
DESTINATION SELECTOR	16-bit selector	Selector to the target code segment (Call, Interrupt or Trap Gate) Selector to the target task state segment (Task Gate)
DESTINATION OFFSET	16-bit offset	Entry point within the target code segment

Figure 13. Gate Descriptor Format

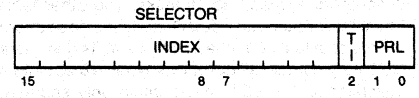
The access byte format is the same for all gate descriptors. P = 1 indicates that the gate contents are valid. P = 0 indicates the contents are not valid and causes exception 11 if referenced. DPL is the descriptor privilege level and specifies when this descriptor may be used by a task (refer to privilege discussion). Bit 4 must equal 0 to indicate a system control descriptor. The type field specifies the descriptor type as indicated in Figure 13.

Segment Descriptor Cache Registers

A segment descriptor cache register is assigned to each of the four segment registers (CS, SS, DS, ES). Segment descriptors are automatically loaded (cached) into a segment descriptor cache register (Figure 15) whenever the associated segment register is loaded with a selector. Only segment descriptors may be loaded into segment descriptor cache registers. Once loaded, all references to that segment of memory use the cached descriptor information instead of reaccessing memory. The descriptor cache registers are not visible to programs. No instructions exist to store their contents. They only change when a segment register is loaded.

Selector Fields

A protected mode selector has three fields: descriptor entry index, local or global descriptor table indicator (TI), and selector privilege (RPL) as shown in Figure 14. These fields select one of two memory based tables of descriptors, select the appropriate table entry and allow high-speed testing of the selector's privilege attribute (refer to privilege discussion).



Bits	Name	Function
1 – 0	REQUESTED PRIVILEGE LEVEL (RPL)	INDICATES SELECTOR PRIVILEGE LEVEL DESIRED
2	TABLE INDICATOR (TI)	TI = 0 USE GLOBAL DESCRIPTOR TABLE (GDT) TI = 1 USE LOCAL DESCRIPTOR TABLE (LDT)
15 – 3	INDEX	SELECT DESCRIPTOR ENTRY IN TABLE

Figure 14. Selector Fields

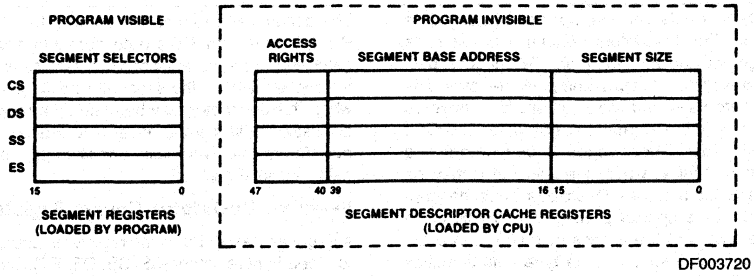


Figure 15. Descriptor Cache Registers

Local and Global Descriptor Tables

Two tables of descriptors, called descriptor tables, contain all descriptors accessible by a task at any given time. A descriptor table is a linear array of up to 8192 descriptors. The upper 13 bits of the selector value are an index into a descriptor table. Each table has a 24-bit base register to locate the descriptor table in physical memory and a 16-bit limit register that confines descriptor access to the defined limits of the table as shown in Figure 16. A restartable exception (13) will occur if an attempt is made to reference a descriptor outside the table limits.

One table, called the Global Descriptor Table (GDT), contains descriptors available to all tasks. The other table, called the Local Descriptor Table (LDT), contains descriptors that can be private to a task. Each task may have its own private LDT. The GDT may contain all descriptor types except interrupt and trap descriptors. The LDT may contain only segment, task gate, and call gate descriptors. A segment cannot be accessed by a task if its segment descriptor does not exist in either descriptor table at the time of access.

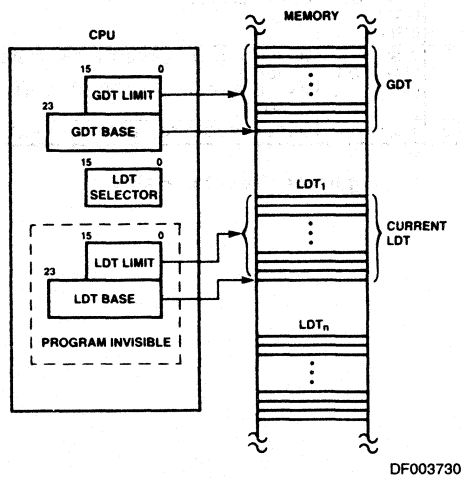


Figure 16. Local and Global Descriptor Table Definition

The LGDT and LLDT instructions load the base and limit of the global and local descriptor tables. LGDT and LLDT are protected. They may only be executed by trusted programs operating at level 0. The LGDT instruction loads a six byte field containing the 16-bit table limit and 24-bit base address of the Global Descriptor Table as shown in Figure 17. The LLDT instruction loads a selector which refers to a Local Descriptor Table descriptor containing the base address and limit for an LDT, as shown in Figure 12.

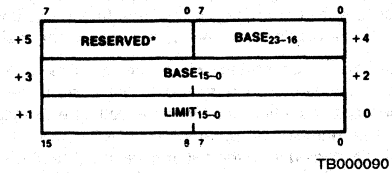


Figure 17. Global Descriptor Table and Interrupt Descriptor Data Type

*Must be set to 0 for compatibility with iAPX 386.

Interrupt Descriptor Table

The protected mode 80286 has a third descriptor table, called the Interrupt Descriptor Table (IDT) (see Figure 18), used to define up to 256 interrupts. It may contain only task gates, interrupt gates and trap gates. The IDT (Interrupt Descriptor Table) has a 24-bit base and 16-bit limit register in the CPU. The protected LIDT instruction loads these registers with a six byte value of identical form to that of the LGDT instruction (see Figure 17 and Protected Mode Initialization).

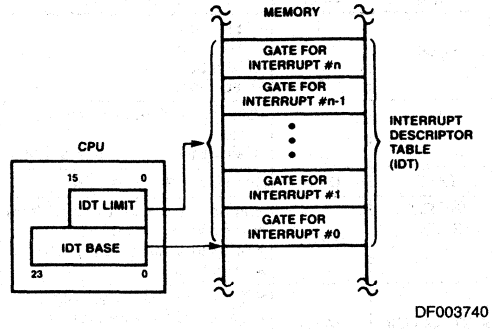


Figure 18. Interrupt Descriptor Table Definition

References to IDT entries are made via INT instructions, external interrupt vectors, or exceptions. The IDT must be at least 256 bytes in size to allocate space for all reserved interrupts.

Privilege

The 80286 has a four-level hierarchical privilege system which controls the use of privileged instructions and access to descriptors (and their associated segments) within a task. Four-level privilege, as shown in Figure 19, is an extension of the user/supervisor mode commonly found in minicomputers. The privilege levels are numbered 0 through 3. Level 0 is the most privileged level. Privilege levels provide protection within a task. (Tasks are isolated by providing private LDT's for each task.) Operating system routines, interrupt handlers, and other system software can be included and protected within the virtual address space of each task using the four levels of privilege. Tasks may also have a separate stack for each privilege level.

Tasks, descriptors, and selectors have a privilege level attribute that determines whether the descriptor may be used. Task privilege effects the use of instructions and descriptors. Descriptor and selector privilege only effect access to the descriptor.

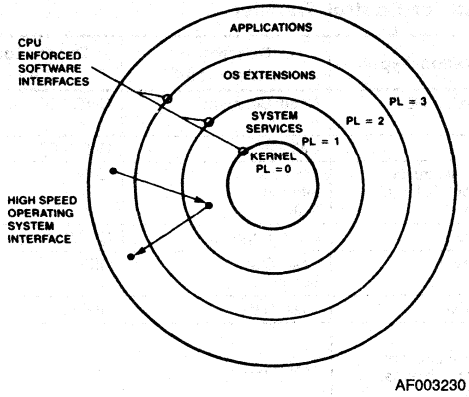


Figure 19. Hierarchical Privilege Levels

Task Privilege

The task always executes at one of the four privilege levels. A task privilege level at any specific instant is called the Current Privilege Level (CPL) and is defined by the lower two bits of the CS register. CPL cannot change during execution in a single code segment. A task's CPL may only be changed by control transfers through gate descriptors to a new code segment (See Control Transfer). Tasks begin executing at the CPL value specified by the code segment when the task is initiated via a task switch operation. A task executing at Level 0 can access all data segments defined in the GDT and the task's LDT and is considered the most trusted level. A task executed at Level 3 has the most restricted access to data and is considered the least trusted level.

Descriptor Privilege

Descriptor privilege is specified by the Descriptor Privilege Level (DPL) field of the descriptor access byte. DPL specifies the least trusted privilege level (CPL) at which a task may access the descriptor. Descriptors with DPL = 0 are the most protected. Only tasks executing at privilege level 0 (CPL = 0) may access them. Descriptors with DPL = 3 are the least protected (i.e. have the least restricted access) since tasks can access them when CPL = 0, 1, 2, or 3. This rule applies to all descriptors, except LDT descriptors.

Selector Privilege

Selector privilege is specified by the Requested Privilege Level (RPL) field in the least significant two bits of a selector. Selector RPL may establish a less trusted privilege level than the current privilege level for the use of a selector. This level is called the task's effective privilege level (EPL). RPL can only reduce the scope of a task's access to data with this selector. A task's effective privilege is the numeric maximum of RPL and CPL. A selector with RPL = 0 imposes no additional restriction on its use while a selector with RPL = 3 can only refer to segments at privilege Level 3 regardless of the task's CPL. RPL is generally used to verify that pointer parameters passed to a more trusted procedure are not allowed to use data at a more privileged level than the caller (refer to pointer testing instructions).

Descriptor Access and Privilege Validation

Determining the ability of a task to access a segment involves the type of segment to be accessed, the instruction used, the type of descriptor used and CPL, RPL, and DPL. The two basic types of segment accesses are control transfer (selectors loaded into CS) and data (selectors loaded into DS, ES or SS).

Data Segment Access

Instructions that load selectors into DS and ES must refer to a data segment descriptor or readable code segment descriptor. The CPL of the task and the RPL of the selector must be the same as or more privileged (numerically equal to or lower than) than the descriptor DPL. In general, a task can only access data segments at the same or less privileged levels than the CPL or RPL (whichever is numerically higher) to prevent a program from accessing data it cannot be trusted to use.

An exception to the rule is a readable conforming code segment. This type of code segment can be read from any privilege level.

If the privilege checks fail (e.g. DPL is numerically less than the maximum of CPL and RPL) or an incorrect type of descriptor is referenced (e.g. gate descriptor or execute only code segment), exception 13 occurs. If the segment is not present, exception 11 is generated.

Instructions that load selectors into SS must refer to data segment descriptors for writable data segments. The descriptor privilege (DPL) and RPL must equal CPL. All other descriptor types or privilege level violation will cause exception 13. A not present fault causes exception 12.

Control Transfer

Four types of control transfer can occur when a selector is loaded into CS by a control transfer operation (see Table 10). Each transfer type can only occur if the operation which loaded the selector references the correct descriptor type. Any violation of these descriptor usage rules (e.g. JMP through a call gate or RET to a Task State Segment) will cause exception 13.

The ability to reference a descriptor for control transfer is also subject to rules of privilege. A CALL or JUMP instruction may only reference a code segment descriptor with DPL equal to the task CPL or a conforming segment with DPL of equal or greater privilege than CPL. The RPL of the selector used to reference the code descriptor must have as much privilege as CPL.

RET and IRET instructions may only reference code segment descriptors with descriptor privilege equal to or less privileged than the task CPL. The selector loaded into CS is the return address from the stack. After the return, the selector RPL is

the task's new CPL. If CPL changes, the old stack pointer is popped after the return address.

When a JMP or CALL references a Task State Segment descriptor, the descriptor DPL must be the same or less privileged than the task's CPL. Reference to a valid Task State Segment descriptor causes a task switch (see Task Switch Operation). Reference to a Task State Segment descriptor at a more privileged level than the task's CPL generates exception 13.

When an instruction or interrupt references a gate descriptor, the gate DPL must have the same or less privilege than the task CPL. If DPL is at a more privileged level than CPL, exception 13 occurs. If the destination selector contained in the gate references a code segment descriptor, the code segment descriptor DPL must be the same or more privileged than the task CPL. If not, Exception 13 is issued. After the control transfer, the code segment descriptor DPL is the task's new CPL. If the destination selector in the gate references a task state segment, a task switch is automatically performed (see Task Switch Operation).

The privilege rules on control transfer require:

– JMP or CALL direct to a code segment (code segment descriptor) can only be to a conforming segment with DPL of equal or greater privilege than CPL or a non-conforming segment at the same privilege level.

– interrupts within the task or calls that may change privilege levels can only transfer control through a gate at the same or a less privileged level than CPL to a code segment at the same or more privileged level than CPL.

– return instructions that don't switch tasks can only return control to a code segment at the same or less privileged level.

– task switch can be performed by a call, a jump or an interrupt which references either a task gate or task state segment at the same or less privileged level.

Table 10. Descriptor Types Used for Control Transfer

Control Transfer Types	Operation Types	Descriptor Referenced	Descriptor Table
Intersegment within the same privilege level	JMP, CALL, RET, IRET*	Code Segment	GDT/LDT
Intersegment to the same or higher privilege level Interrupt within task may change CPL.	CALL	Call Gate	GDT/LDT
	Interrupt Instruction, Exception, External Interrupt	Trap or Interrupt Gate	IDT
Intersegment to a lower privilege level (changes task CPL)	RET, IRET*	Code Segment	GDT/LDT
Task Switch	CALL, JMP	Task State Segment	GDT
	CALL, JMP	Task Gate	GDT/LDT
	IRET** Interrupt Instruction, Exception, External Interrupt	Task Gate	IDT

*NT (Nested Task bit of flag word) = 0
**NT (Nested Task bit of flag word) = 1

Privilege Level Changes

Any control transfer that changes CPL within the task causes a change of stacks as part of the operation. Initial values of SS:SP for privilege levels 0, 1, and 2 are kept in the task state segment (refer to Task Switch Operation). During a JMP or CALL control transfer, the new stack pointer is loaded into the SS and SP registers and the previous stack pointer is pushed onto the new stack.

When returning to the original privilege level, its stack is restored as part of the RET or IRET instruction operation. For subroutine calls that pass parameters on the stack and cross privilege levels, a fixed number of words, as specified in the gate, are copied from the previous stack to the current stack. The intersegment RET instruction with a stack adjustment value will correctly restore the previous stack pointer upon return.

Protection

The 80286 includes mechanisms to protect critical instructions that affect the CPU execution state (e.g. HLT) and code or data segments from improper usage. These mechanisms are grouped under the term "protection" and have three forms:

Restricted usage of segments (e.g. no write allowed to read-only data segments). The only segments available for use are defined by descriptors in the Local Descriptor Table (LDT) and Global Descriptor Table (GDT).

Restricted access to segments via the rules of privilege and descriptor usage.

Privileged instructions or operations that may only be executed at certain privilege levels as determined by the CPL and I/O Privilege Level (IOPL). The IOPL is defined by bits 14 and 13 of the flag word.

These checks are performed for all instructions and can be split into three categories: segment load checks (Table 11), operand reference checks (Table 12), and privileged instruction checks (Table 13). Any violation of the rules shown will result in an exception. A not-present exception related to the stack segment causes exception 12.

The IRET and POPF instructions do not perform some of their defined functions if CPL is not of sufficient privilege (numerically small enough). No exceptions or other indication are given when these conditions occur.

The IF bit is not changed if CPL > IOPL.
The IOPL field of the flag word is not changed if CPL > 0.

Table 11. Segment Register Load Checks

Error Description	Exception Number
Descriptor table limit exceeded	13
Segment descriptor not-present	11 or 12
Privilege rules violated	13
Invalid descriptor/segment type segment register load: —Read only data segment load to SS —Special control descriptor load to DS, ES, SS —Execute only segment load to DS, ES, SS —Data segment load to CS —Read/Execute code segment load to SS	13

Table 12. Operand Reference Checks

Error Description	Exception Number
Write into code segment	13
Read from execute-only code segment	13
Write to read-only data segment	13
Segment limit exceeded[1]	12 or 13

Note 1: Carry out in offset calculations is ignored.

Table 13. Privileged Instruction Checks

Error Description	Exception Number
CPL ≠ 0 when executing the following instructions: LIDT, LLDT, LGDT, LTR, LMSW, CTS, HLT	13
CPL > IOPL when executing the following instructions: INS, IN, OUTS, OUT, STI, CLI, LOCK	13

Exceptions

The 80286 detects several types of exceptions and interrupts in protected mode (see Table 14). Most are restartable after the exceptional condition is removed. Interrupt handlers for most exceptions receive an error code, pushed on the stack after the return address, that identifies the selector involved (0 if none). The return address normally points to the failing instruction, including all leading prefixes. For a processor extension segment overrun exception, the return address will not point at the ESC instruction that caused the exception; however, the processor extension registers may contain the address of the failing instruction.

Table 14. Protected Mode Exceptions

Interrupt Vector	Function	Return Address At Failing Instruction?	Always Restartable?	Error Code on Stack?
8	Double exception detected	Yes	No	Yes
9	Processor extension segment overrun	No	No	No
10	Invalid task state segment	Yes	Yes	Yes
11	Segment not present	Yes	Yes	Yes
12	Stack segment overrun or segment not present	Yes	Yes[1]	Yes
13	General protection	Yes	No	Yes

Note 1: When a PUSHA or POPA instruction attempts to wrap around the stack segment, the machine state after the exception will not be restartable. This condition is identified by the value of the saved SP being either 0000(H), 0001(H), FFFE(H), or FFFF(H).

All these checks are performed for all instructions and can be split into three categories: segment load checks (Table 11), operand reference checks (Table 12), and privileged instruction checks (Table 13). Any violation of the rules shown will result in an exception. A not-present exception related to the stack segment causes exception 12.

Special Operations

Task Switch Operation

The 80286 provides a built-in task switch operation which saves the entire 80286 execution state (registers, address space, and a link to the previous task), loads a new execution state, and commences execution in the new task. Like gates, the task switch operation is invoked by executing an inter-segment JMP or CALL instruction which refers to a Task State Segment (TSS) or task gate descriptor in the GDT or LDT. An INT n instruction, exception, or external interrupt may also invoke the task switch operation by selecting a task gate descriptor in the associated IDT descriptor entry.

The TSS descriptor points at a segment (see Figure 20) containing the entire 80286 execution state while a task gate descriptor contains a TSS selector. The limit field must be > 002B(H).

Each task must have a TSS associated with it. The current TSS is identified by a special register in the 80286 called the Task Register (TR). This register contains a selector referring to the task state segment descriptor that defines the current TSS. A hidden base and limit register associated with TR are loaded whenever TR is loaded with a new selector.

The IRET instruction is used to return control to the task that called the current task or was interrupted. Bit 14 in the flag register is called the Nested Task (NT) bit. It controls the function of the IRET instruction. If NT = 0, the IRET instruction performs the regular current task return; when NT = 1, IRET performs a task switch operation back to the previous task.

When a CALL or INT instruction initiates a task switch, the old and new TSS will be marked busy and the back link field of the new TSS set to the old TSS selector. The NT bit of the new task is set by CALL or INT initiated task switches. An interrupt that does not cause a task switch will clear NT. NT may also be set or cleared by POPF or IRET instructions.

The task state segment is marked busy by changing the descriptor type field from Type 1 to Type 3. Use of a selector that references a busy task state segment causes Exception 13.

Processor Extension Context Switching

The context of a processor extension is not changed by the task switch operation. A processor extension context need only be changed when a different task attempts to use the processor extension (which still contains the context of a previous task). The 80286 detects the first use of a processor extension after a task switch by causing the processor extension not present exception (7). The interrupt handler may then decide whether a context change is necessary.

Whenever the 80286 switches tasks, it sets the Task Switched (TS) bit of the MSW. TS indicates that a processor extension context may belong to a different task than the current one. The processor extension not present exception (7) will occur when attempting to execute an ESC or WAIT instruction if TS = 1 and a processor extension is present (MP = 1 in MSW).

Pointer Testing Instructions

The 80286 provides several instructions to speed pointer testing and consistency checks for maintaining system integrity (see Table 15). These instructions use the memory management hardware to verify that a selector value refers to an appropriate segment without risking an exception. A condition flag indicates whether use of the selector or segment will cause an exception.

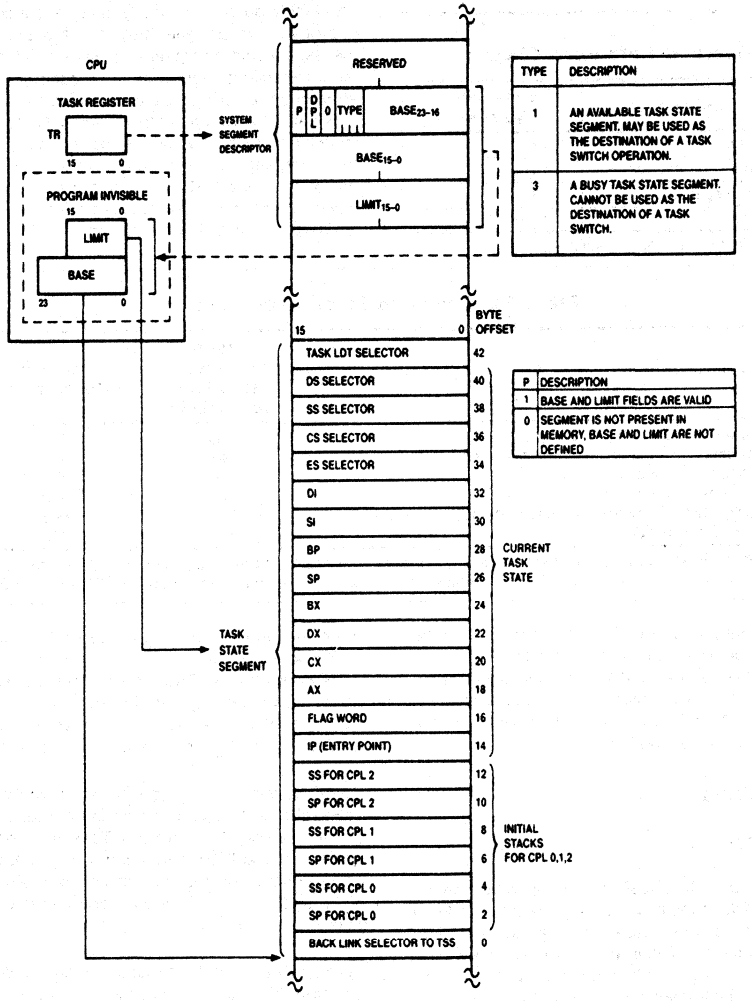


Figure 20. Task State Segment and TSS Registers

Table 15. Pointer Test Instructions

Instruction	Operands	Function
ARPL	Selector, Register	Adjust Requested Privilege Level: adjusts the RPL of the selector to the numeric maximum of current selector RPL value and the RPL value in the register. Set zero flag if selector RPL was changed.
VERR	Selector	VERify for Read: sets the zero flag if the segment referred to by the selector can be read.
VERW	Selector	VERify for Write: sets the zero flag if the segment referred to by the selector can be written.
LSL	Register, Selector	Load Segment Limit: reads the segment limit into the register if privilege rules and descriptor type allow. Set zero flag if successful.
LAR	Register, Selector	Load Access Rights: reads the descriptor access rights byte into the register if privilege rules allow. Set zero flag if successful.

Double Fault and Shutdown

If two separate exceptions are detected during a single instruction execution, the 80286 performs the double fault exception (8). If an exception occurs during processing of the double fault exception, the 80286 will enter shutdown. During shutdown no further instructions or exceptions are processed. Either NMI (CPU remains in protected mode) or RESET (CPU exits protected mode) can force the 80286 out of shutdown. Shutdown is externally signalled via a HALT bus operation with A_1 HIGH.

Protected Mode Initialization

The 80286 initially executes in real address mode after RESET. To allow initialization code to be placed at the top of physical memory, A_{23-20} will be HIGH when the 80286 performs memory references relative to the CS register, until CS is changed. A_{23-20} will be zero for references to the DS, ES, or SS segments. Changing CS in real address mode will force $A_{23} - A_{20}$ LOW whenever using CS thereafter. The initial CS:IP value of FFOO:FFFO provides 64K bytes of code space for initialization code without changing CS.

Before placing the 80286 into protected mode, several registers must be initialized. The GDT and IDT base registers must refer to a valid GDT and IDT. After executing the LMSW instruction to set PE, the 80286 must immediately execute an intrasegment JMP instruction to clear the instruction queue of instructions decoded in real address mode.

To force the 80286 CPU registers to match the initial protected mode state assumed by software, execute a JMP instruction with a selector referring to the initial TSS used in the system. This will load the task register, local descriptor table register, segment registers and initial general register state. The TR should point at a valid TSS since a task switch operation involves saving the current task state.

System Interface

The 80286 system interface appears in two forms: a local bus and a system bus. The local bus consists of address, data, status, and control signals at the pins of the CPU. A system bus is any buffered version of the local bus. A system bus may also differ from the local bus in terms of coding of status and control lines and/or timing and loading of signals. The 80286 family includes several devices to generate standard system buses such as the IEEE 796 Standard Multibus™.

Bus Interface Signals and Timing

The 80286 microsystem local bus interfaces the 80286 to local memory and I/O components. The interface has 24 address lines, 16 data lines, and 8 status and control signals.

The 80286 CPU, 82284 clock generator, 82C288 bus controller, 82289 bus arbiter, 8286/7 transceivers, and 8282/3 latches provide a buffered and decoded system bus interface. The 82284 generates the system clock and synchronizes $\overline{\text{READY}}$ and RESET. The 82C288 converts bus operation status encoded by the 80286 into command and bus control signals. These components can provide the timing and electrical power drive levels required for most system bus interfaces including the multibus.

Physical Memory and I/O Interface

A maximum of 16 megabytes of physical memory can be addressed in protected mode. One megabyte can be addressed in real address mode. Memory is accessible as bytes or words. Words consist of any two consecutive bytes addressed with the least significant byte stored in the lowest address.

Byte transfers occur on either half of the 16-bit local data bus. Even bytes are accessed over D_{7-0} while odd bytes are transferred over D_{15-8}. Even-addressed words are transferred over D_{15-0} in one bus cycle, while odd-addressed words require two bus operations. The first transfers data on D_{15-8}, and the second transfers data on D_{7-0}. Both byte data transfers occur automatically, transparent to software.

Two bus signals, A_0 and $\overline{\text{BHE}}$, control transfers over the lower and upper halves of the data bus. Even address byte transfers are indicated by A_0 LOW and $\overline{\text{BHE}}$ HIGH. Odd address byte transfers are indicated by A_0 HIGH and $\overline{\text{BHE}}$ LOW. Both A_0 and $\overline{\text{BHE}}$ are LOW for even address word transfers.

The I/O address space contains 64K addresses in both modes. The I/O space is accessible as either bytes or words, as is memory. Byte wide peripheral devices may be attached to either the upper or lower byte of the data bus. Byte-wide I/O devices attached to the upper data byte (D_{15-8}) are accessed with odd I/O addresses. Devices on the lower data byte are accessed with even I/O addresses. An interrupt controller such as the 8259A must be connected to the lower data byte (D_{7-0}) for proper return of the interrupt vector.

Bus Operation

The 80286 uses a double frequency system clock (CLK input) to control bus timing. All signals on the local bus are measured relative to the system CLK input. The CPU divides the system clock by 2 to produce the internal processor clock, which determines bus state. Each processor clock is composed of two system clock cycles named phase 1 and phase 2. The 82284 clock generator output (PCLK) identifies the next phase of the processor clock. (See Figure 21.)

Multibus is a registered trademark of Intel Corporation.

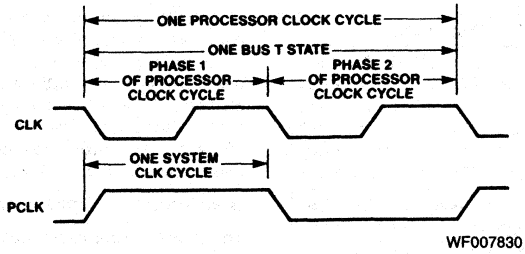


Figure 21. System and Processor Clock Relationships

Six types of bus operations are supported: memory read, memory white, I/O read, I/O write, interrupt acknowledge, and halt/shutdown. Data can be transferred at a maximum rate of one word per two processor clock cycles.

The 80286 bus has three basic states: idle (T_i), send status (T_s), and perform command (T_c). The 80286 CPU also has a fourth local bus state called hold (T_h). T_h indicates that the 80286 has surrendered control of the local bus to another bus master in response to a HOLD request.

Each bus state is one processor clock long. Figure 22 shows the four 80286 local bus states and allowed transitions.

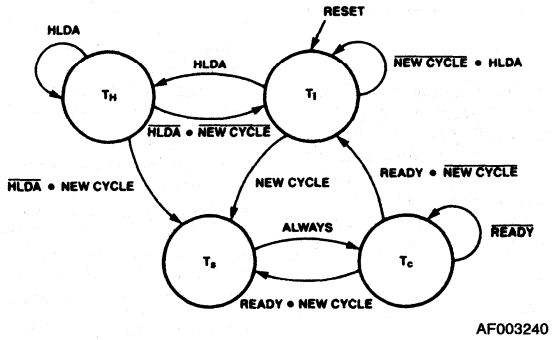


Figure 22. 80286 Bus States

Bus States

The idle (T_i) state indicates that no data transfers are in progress or requested. The first active state, T_s, is signalled by either status line $\overline{S1}$ or $\overline{S0}$ going LOW also identifying phase 1 of the processor clock. During T_s, the command encoding, the address, and data (for a write operation) are available on the 80286 output pins. The 82C288 bus controller decodes the status signals and generates Multibus compatible read/write command and local transceiver control signals.

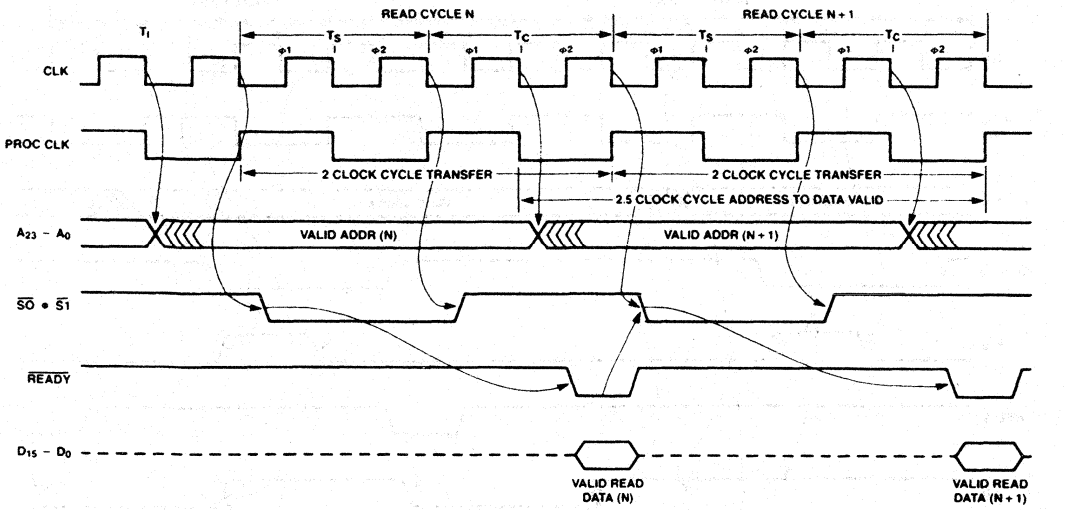
After T_s, the perform command (T_c) state is entered. Memory or I/O devices respond to the bus operation during T_c, either transferring read data to the CPU or accepting write data. T_c states may be repeated as often as necessary to assure sufficient time for the memory or I/O device to respond. The $\overline{READY}$ signal determines whether T_c is repeated.

During hold (T_h), the 80286 will float all address, data, and status output pins, enabling another bus master to use the local bus. The 80286 HOLD input signal is used to place the 80286 into the T_h state. The 80286 HLDA output signal indicates that the CPU has entered T_h.

Pipelined Addressing

The 80286 uses a local bus interface with pipelined timing to allow as much time as possible for data access. Pipelined timing allows bus operations to be performed in two processor cycles, while allowing each individual bus operation to last for three processor cycles.

The timing of the address outputs is pipelined such that the address of the next bus operation becomes available during the current bus operation. Or in other words, the first clock of the next bus operation is overlapped with the last clock of the current bus operation. Therefore, address decode and routing logic can operate in advance of the next bus operation. External address latches may hold the address stable for the entire bus operation and provide additional AC and DC buffering.



WF007840

Figure 23. Basic Bus Cycle

The 80286 does not maintain the address of the current bus operation during all T_C states. Instead, the address for the next bus operation may be emitted during phase 2 of any T_C. The address remains valid during phase 1 of the first T_C to guarantee hold time, relative to ALE, for the address latch inputs.

Bus Control Signals

The 82C288 bus controller provides control signals: address latch enable (ALE), Read/Write commands, data transmit/receive (DT/$\overline{R}$), and data enable (DEN) that control the address latches, data transceivers, write enable, and output enable for memory and I/O systems.

The Address Latch Enable (ALE) output determines when the address may be latched. ALE provides at least one system CLK period of address hold time from the end of the previous bus operation until the address for the next bus operation appears at the latch outputs. This address hold time is required to support Multibus® and common memory systems.

The data bus transceivers are controlled by 82C288 outputs Data Enable (DEN) and Data Transmit/Receive (DT/$\overline{R}$). DEN enables the data transceivers while DT/$\overline{R}$ controls transceiver direction. DEN and DT/$\overline{R}$ are timed to prevent bus contention between the bus master, data bus transceivers, and system data bus transceivers.

Command Timing Controls

Two system timing customization options, command extension and command delay, are provided on the 80286 local bus.

Command extension allows additional time for external devices to respond to a command and is analogous to inserting wait states on the 8086. External logic can control the duration of any bus operation such that the operation is only as long as necessary. The $\overline{READY}$ input signal can extend any bus operation for as long as necessary.

Command delay allows an increase of address or write data set-up time to system bus command active for any bus operation by delaying when the system bus command becomes active. Command delay is controlled by the 82C288 CMDLY input. After T_S, the bus controller samples CMDLY at each failing edge of CLK. If CMDLY is HIGH, the 82C288 will not activate the command signal. When CMDLY is LOW, the 82C288 will activate the command signal. After the command becomes active, the CMDLY input is not sampled.

When a command is delayed, the available response time from command active to return read data or accept write data is less. To customize system bus timing, an address decoder can determine which bus operations require delaying the command. The CMDLY input does not affect the timing of ALE, DEN, or DT/$\overline{R}$.

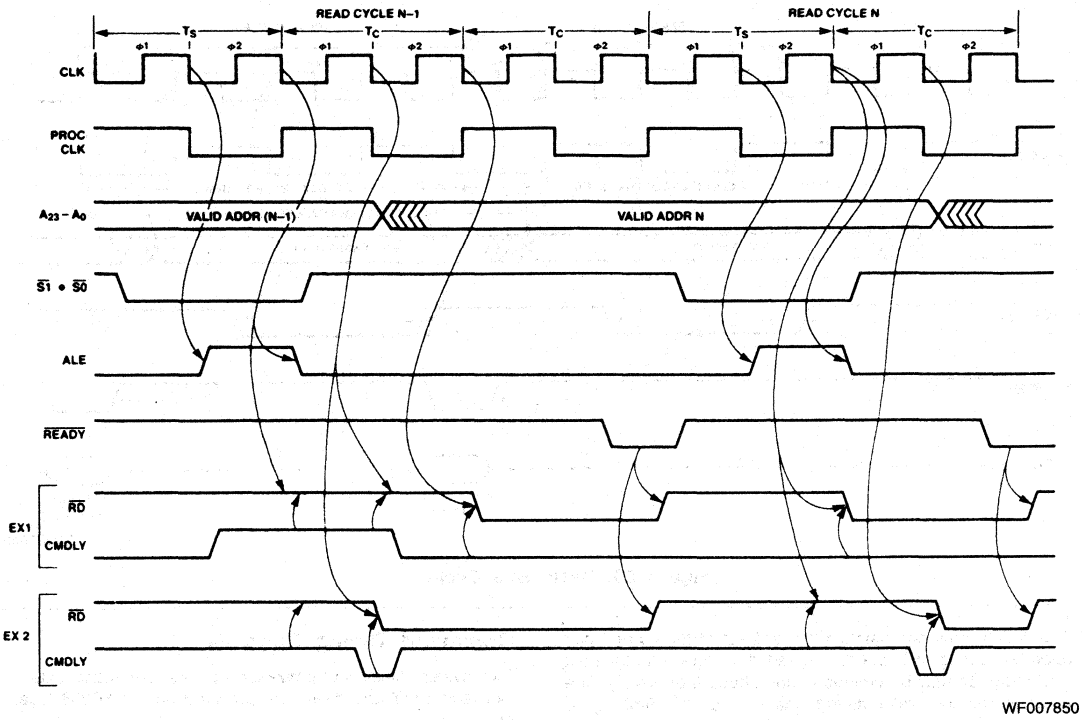


Figure 24. CMDLY Controls and Leading Edge of the Command

Figure 24 illustrates four uses of CMDLY. Example 1 shows delaying the read command two system CLKs for cycle N-1 and no delay for cycle N, and example 2 shows delaying the read command one system CLK for cycle N-1 and one system CLK delay for cycle N.

Bus Cycle Termination

At maximum transfer rates, the 80286 bus alternates between the status and command states. The bus status signals become inactive after T_S so that they may correctly signal the start of the next bus operation after the completion of the current cycle. No external indication of T_C exists on the 80286 local bus. The bus master and bus controller enter T_C directly after T_S and continue executing T_C cycles until terminated by $\overline{READY}$.

$\overline{READY}$ Operation

The current bus master and 82C288 bus controller terminate each bus operation simultaneously to achieve maximum bus bandwidth. Both are informed in advance by $\overline{READY}$ active which identifies the last T_C cycle of the current bus operation. The bus master and bus controller must see the same sense of the $\overline{READY}$ signal, thereby requiring $\overline{READY}$ be synchronous to the system clock.

Synchronous Ready

The 82284 clock generator provides $\overline{READY}$ synchronization from both synchronous and asynchronous sources (see Figure 25). The synchronous ready input ($\overline{SRDY}$) of the clock generator is sampled with the falling edge of CLK at the end of phase 1 of each T_C. The state of $\overline{SRDY}$ is then broadcast to the bus master and bus controller via the $\overline{READY}$ output line.

Asynchronous Ready

Many systems have devices or subsystems that are asynchronous to the system clock. As a result, their ready outputs cannot be guaranteed to meet the 82284 $\overline{SRDY}$ set-up and hold time requirements. The 82284 asynchronous ready input ($\overline{ARDY}$) is designed to accept such signals. The $\overline{ARDY}$ input is sampled at the beginning of each T_C cycle by 82284 synchronization logic. This provides a system CLK cycle time to resolve its value before broadcasting it to the bus master and bus controller.

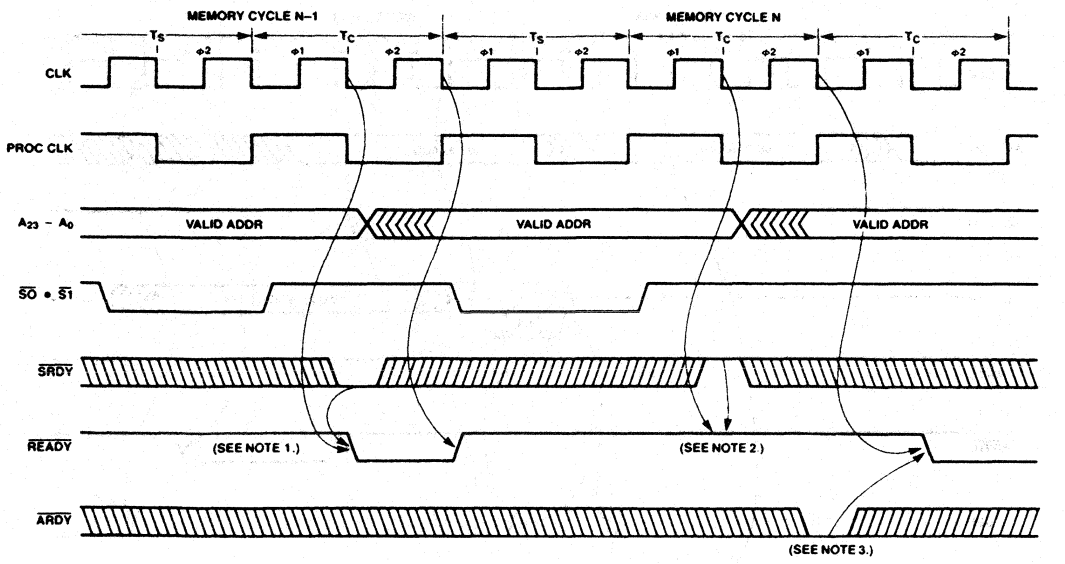


Figure 25. Synchronous and Asynchronous Ready

Notes: 1. $\overline{SRDYEN}$ is active low.
2. If $\overline{SRDYEN}$ is high, the state of $\overline{SRDY}$ will not effect $\overline{READY}$.
3. $\overline{ARDYEN}$ is active low.

Each ready input of the 82284 has an enable pin ($\overline{SRDYEN}$ and $\overline{ARDYEN}$) to select whether the current bus operation will be terminated by the synchronous or asynchronous ready. Either of the ready inputs may terminate a bus operation. These enable inputs are active low and have the same timing as their respective ready inputs. Address decode logic usually selects whether the current bus operation should be terminated by $\overline{ARDY}$ or $\overline{SRDY}$.

Data Bus Control

Figures 26, 27, and 28 show how the DT/$\overline{R}$, DEN, data bus, and address signals operate for different combinations of read, write, and idle bus operations. DT/$\overline{R}$ goes active (LOW) for a read operation. DT/$\overline{R}$ remains HIGH before, during, and between write operations.

The data bus is driven with write data during the second phase of T_S. The delay in write data timing allows the read data drivers, from a previous read cycle, sufficient time to enter 3-state OFF before the 80286 CPU begins driving the local data bus for write operations. Write data will always remain valid for one system clock past the last T_C to provide sufficient hold time for Multibus or other similar memory or I/O systems. During write-read or write-idle sequences the data bus enters 3- state OFF during the second phase of the processor cycle after the last T_C. In a write-write sequence the data bus does not enter 3-state OFF between T_C and T_S.

Bus Usage

The 80286 local bus may be used for several functions: instruction data transfers, data transfers by other bus masters, instruction fetching, processor extension data transfers, interrupt acknowledge, and halt/shutdown. This section describes local bus activities which have special signals or requirements.

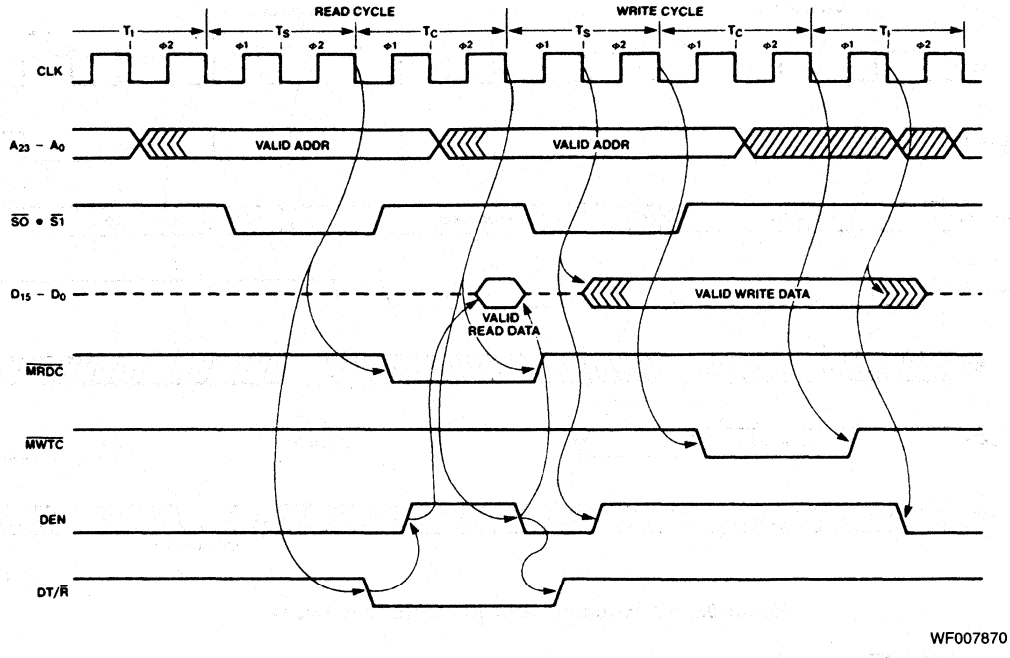


Figure 26. Back to Back Read-Write Cycles

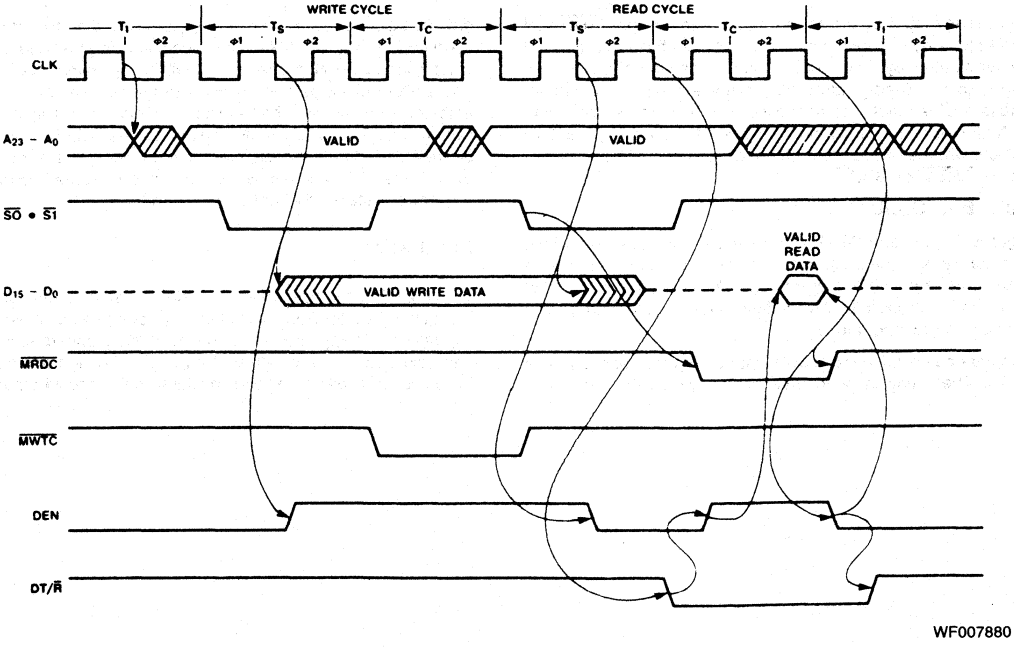
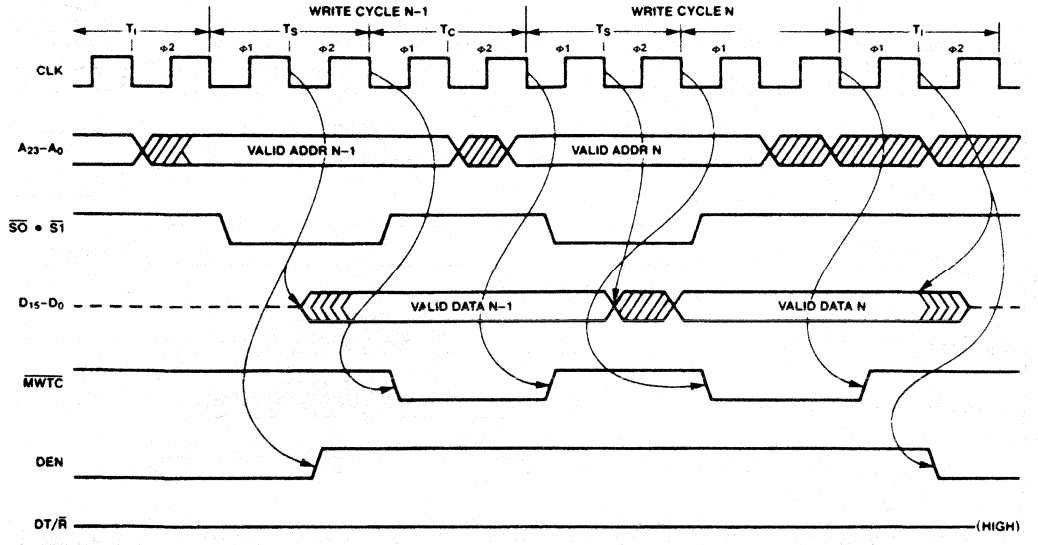


Figure 27. Back to Back Write-Read Cycles



WF007890

Figure 28. Back-to-Back Write-Write Cycles

HOLD and HLDA

HOLD and HLDA allow another bus master to gain control of the local bus by placing the 80286 bus into the T_h state. The sequence of events required to pass control between the 80286 and another local bus master are shown in Figure 29.

In this example, the 80286 is initially in the T_h, state as signaled by HLDA being active. Upon leaving T_h, as signaled by HLDA going inactive, a write operation is started. During the write operation another local bus master requests the local bus from the 80286 as shown by the HOLD signal. After completing the write operation, the 80286 performs one T_i bus cycle, to guarantee write data hold time, then enters T_h as signaled by HLDA going active.

The CMDLY signal and $\overline{ARDY}$ ready are used to start and stop the write bus command, respectively. Note that $\overline{SRDY}$ must be inactive or disabled by $\overline{SRDYEN}$ to guarantee $\overline{ARDY}$ will terminate the cycle.

Instruction Fetching

The 80286 Bus Unit (BU) will fetch instructions ahead of the current instruction being executed. This activity is called prefetching. It occurs when the local bus would otherwise be idle and obeys the following rules:

A prefetch bus operation starts when at least two bytes of the 6-byte prefetch queue are empty.

The prefetcher normally performs word prefetches independent of the byte alignment of the code segment base in physical memory.

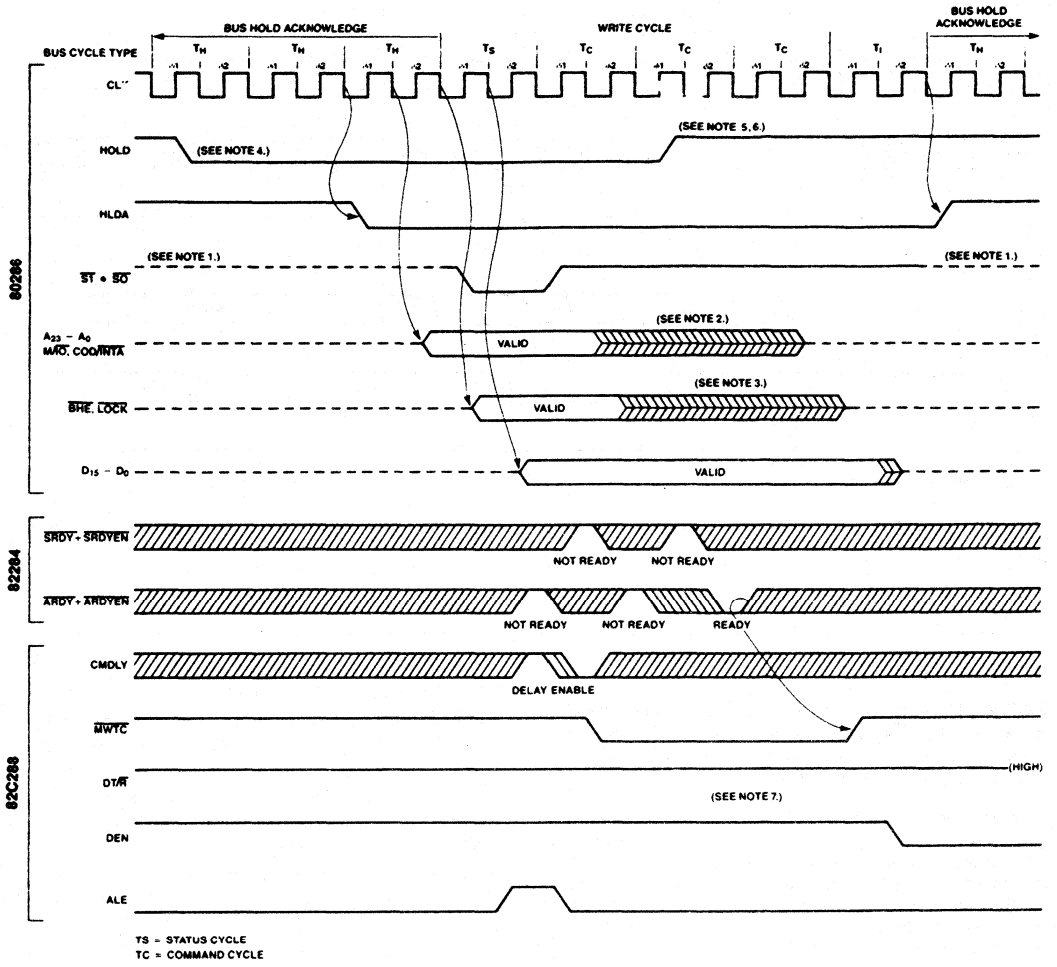
The prefetcher will perform only a byte code fetch operation for control transfers to an instruction beginning on a numerically odd physical address.

Prefetching stops whenever a control transfer or HLT instruction is decoded by the IU and placed into the instruction queue.

In real address mode, the prefetcher may fetch up to 5 bytes beyond the last control transfer or HLT instruction in a code segment.

In protected mode, the prefetcher will never cause a segment overrun exception. The prefetcher stops at the last physical memory word of the code segment. Exception 13 will occur if the program attempts to execute beyond the last full instruction in the code segment.

If the last byte of a code segment appears on an even physical memory address, the prefetcher will read the next physical byte of memory (perform a word code fetch). The value of this byte is ignored and any attempt to execute it causes exception 13.



WF007901

Figure 29. Multibus Write Terminated by Asynchronous Ready with Bus Hold

Notes:
1. Status lines are not driven by 80286, yet remain high due to pull-up resistors in 82C288 and 82289 during HOLD state.
2. Address, M/$\overline{IO}$ and COD/$\overline{INTA}$ may start floating during any TC depending on when internal 80286 bus arbiter decides to release bus to external HOLD. The float starts in $\phi 2$ of TC.
3. $\overline{BHE}$ and $\overline{LOCK}$ may start floating after the end of any TC depending on when internal 80286 bus arbiter decides to release bus to external HOLD.
4. The minimum HOLD ↓ to HLDA ↓ time is shown. Maximum is one T_H longer.
5. The earliest HOLD ↑ time is shown which will always allow a subsequent memory cycle if pending.
6. The minimum HOLD ↑ to HLDA ↑ time is shown. Maximum is a function of the instruction, type of bus cycle and other machine status (i.e., Interrupts, Waits, Lock, etc.)
7. Asynchronous ready allows termination of the cycle. Synchronous ready does not signal ready in this example. Synchronous ready state is ignored after ready is signaled via the asynchronous input.

Processor Extension Transfers

The processor extension interface uses I/O port addresses 00F8(H), 00FA(H), and 00FC(H) which are part of the I/O port address range and is a reserved area. An ESC instruction with EM = 0 and TS = 0 will perform I/O bus operations to one or more of these I/O port addresses independent of the value of IOPL and CPL.

ESC instructions with memory references enable the CPU to accept PEREQ inputs for processor extension operand transfers. The CPU will determine the operand starting address and read/write status of the instruction. For each operand transfer, two or three bus operations, one word transfer with I/O port address 00FA(H), and one or two bus operations with memory are performed. Three bus operations are required for each word operand aligned on an odd byte address.

Interrupt Acknowledge Sequence

Figure 30 illustrates an interrupt acknowledge sequence performed by the 80286 in response to an INTR input. An interrupt acknowledge sequence consists of two INTA bus operations. The first allows a master 8259A Programmable Interrupt Controller (PIC) to determine which if any of its slaves should return the interrupt vector. An eight bit vector is read by the 80286 during the second INTA bus operation to select an interrupt handler routine from the interrupt table.

The Master Cascade Enable (MCE) signal of the 82C288 is used to enable the cascade address drivers, during INTA bus operations (see Figure 30), onto the local address bus for distribution to slave interrupt controllers via the system address bus. The 80286 emits the $\overline{LOCK}$ signal (active LOW) during T_s of the first INTA bus operation. A local bus "hold" request will not be honored until the end of the second INTA bus operation.

Three idle processor clocks are provided by the 80286 between INTA bus operations to allow for the minimum INTA to INTA time and CAS (cascade address) out delay of the 8259A. The second INTA bus operation must always have at least one extra T_c state added via logic controlling $\overline{READY}$. $A_{23} - A_0$ are in 3-state OFF until after the first T_c state of the second INTA bus operation. This prevents bus contention between the cascade address drivers and CPU address drivers. The extra T_c state allows time for the 80286 to resume driving the address lines for subsequent bus operations.

Local Bus Usage Priorities

The 80286 local bus is shared among several internal units and external HOLD requests. In case of simultaneous requests, their relative priorities are:

(Highest) Any transfers which assert $\overline{LOCK}$ either explicitly (via the LOCK instruction prefix) or implicitly (i.e., segment descriptor access, interrupt acknowledge sequence, or an XCHG with memory).

The second of the two byte bus operations required for an odd aligned word operand.

Local bus request via HOLD input.

Processor extension data operand transfer via PEREQ input.

Data transfer performed by EU as part of an instruction.

(Lowest) An instruction prefetch request from BU. The EU will inhibit prefetching two processor clocks in advance of any data transfers to minimize waiting by EU for a prefetch to finish.

Halt or Shutdown Cycles

The 80286 externally indicates halt or shutdown conditions as a bus operation. These conditions occur due to a HLT instruction or multiple protection exceptions while attempting to execute one instruction. A halt or shutdown bus operation is signalled when $\overline{S1}$, $\overline{S0}$ and COD/$\overline{INTA}$ are LOW and M/$\overline{IO}$ is HIGH. A_1 HIGH indicates halt, and A_1 LOW indicates shutdown. The 82C288 bus controller does not issue ALE, nor is $\overline{READY}$ required to terminate a halt or shutdown bus operation.

During halt or shutdown, the 80286 may service PEREQ or HOLD requests. A processor extension segment overrun exception during shutdown will inhibit further service of PEREQ. Either NMI or RESET will force the 80286 out of either halt or shutdown. An INTR, if interrupts are enabled, or a processor extension segment overrun exception will also force the 80286 out of halt.

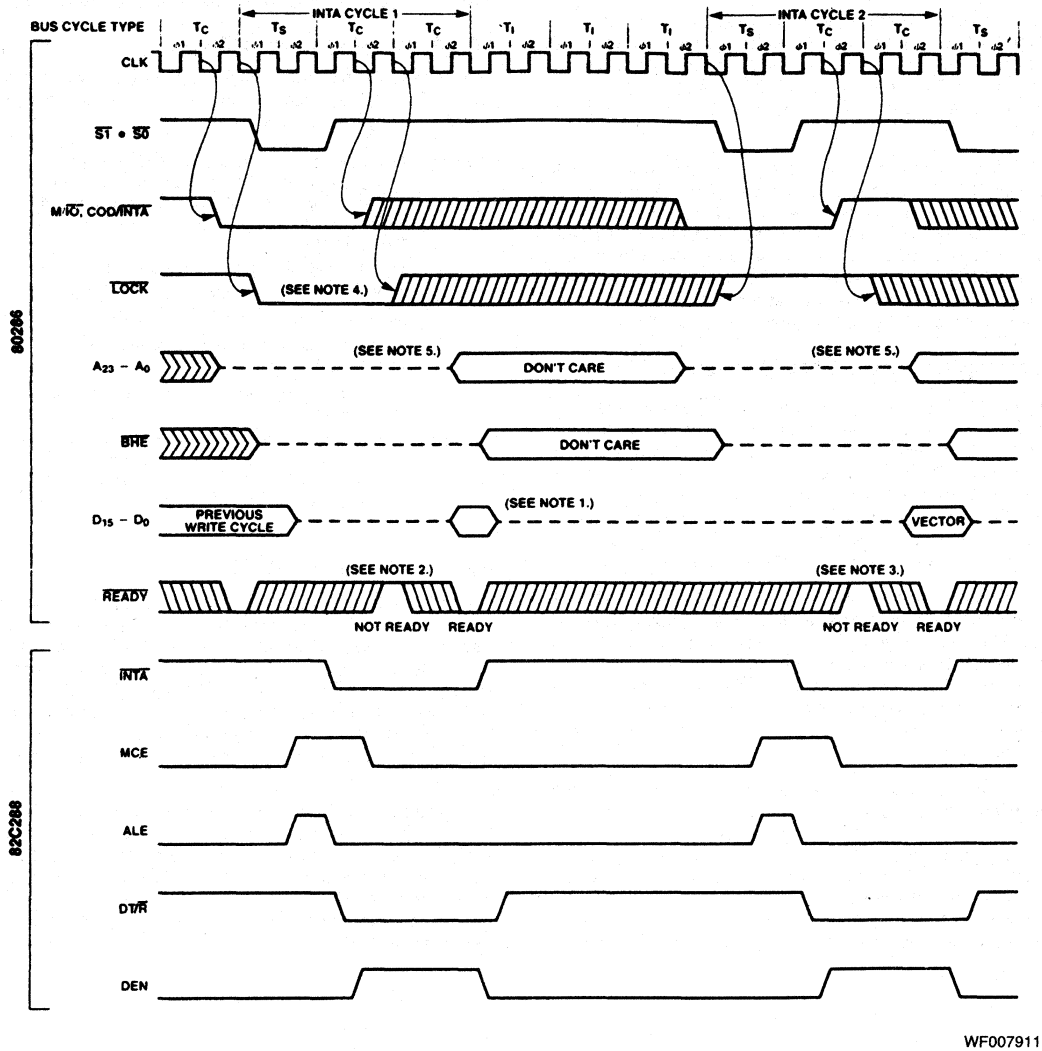
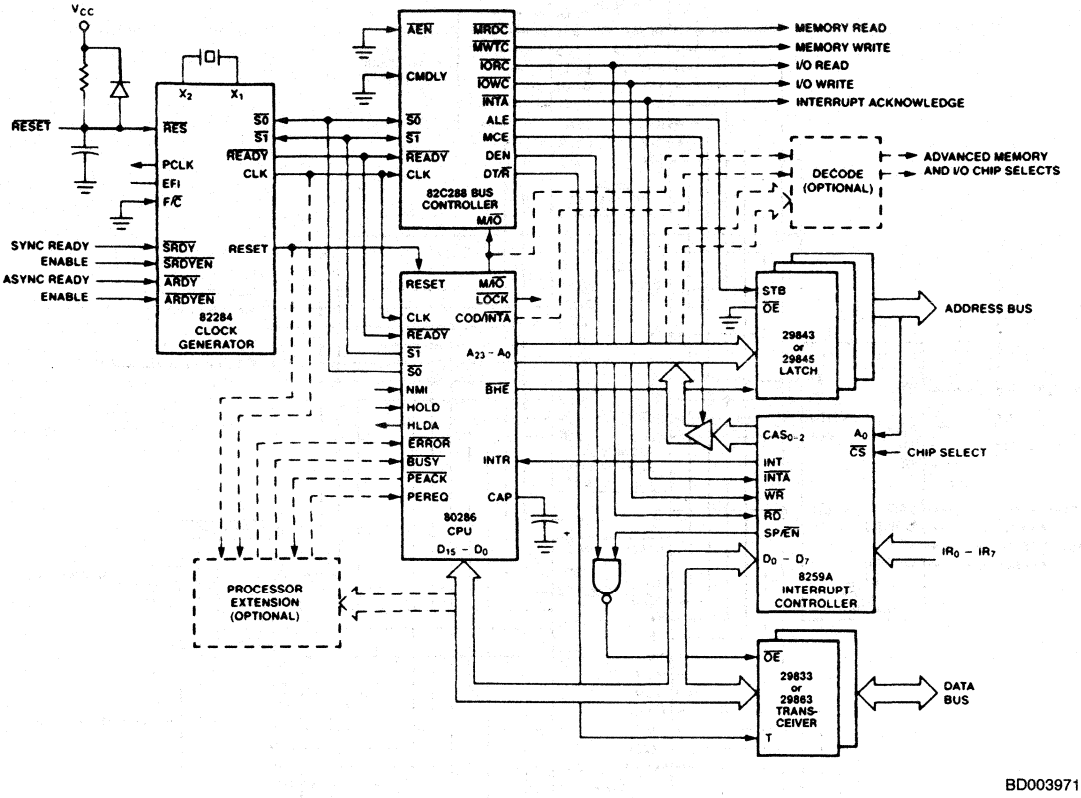


Figure 30. Interrupt Acknowledge Sequence

Notes:
1. Data is ignored.
2. First INTA cycle should have at least one wait state inserted to meet 8259A minimum INTA pulse width.
3. Second INTA cycle must have at least one wait state inserte d since the CPU will not drive $A_{23} - A_0$, $\overline{BHE}$, and $\overline{LOCK}$ until after the first TC state.
   The CPU imposed one/clock delay prevents bus contention between cascade address buffer being disabled by MCE ↓ and address outputs.
   Without the wait state, the 80286 address will not be valid for a memory cycle started immediately after the second INTA cycle. The 8259A also requires one wait state for minimum INTA pulse width.
4. $\overline{LOCK}$ is active for the first INTA cycle to prevent the 82289 from releasing the bus between INTA cycles in a multi-master system.
5. $A_{23} - A_0$ exits 3-state OFF during $\phi 2$ of the second T_C in the INTA cycle.



BD003971

Figure 31. Basic 80286 System Configuration

System Configurations

The versatile bus structure of the 80286 microsystem, with a full complement of support chips, allows flexible configuration of a wide range of systems. The basic configuration, shown in Figure 31, is similar to an iAPX 86 maximum mode system. It includes the CPU plus an 8259A interrupt controller, 82284 clock generator, and the 82C288 Bus Controller. The iAPX 86 latches (29843 and 29845) and transceivers (29833 and 29863) may be used in an 80286 microsystem.

As indicated by the dashed lines in Figure 31, the ability to add processor extensions is an integral feature of 80286 microsystems. The processor extension interface allows external hardware to perform special functions and transfer data concurrent with CPU execution of other instructions. Full system integrity is maintained because the 80286 supervises all data transfers and instruction execution for the processor extension.

The 80286 with the 80287 numeric processor extension (NPX) uses this interface. The iAPX 286/287 has all the instructions and data types of an iAPX 86/87 or iAPX 88/87. The 80287 NPX can perform numeric calculations and data transfers concurrently with CPU program execution. Numerics code and data have the same integrity as all other information protected by the 80286 protection mechanism.

The 80286 can overlap chip select decoding and address propagation during the data transfer for the previous bus operation. This information is latched into the 29843/45's by ALE during the middle of a T_S cycle. The latched chip select and address information remains stable during the bus operation while the next cycle's address is being decoded and propagated into the system. Decode logic can be implemented with a high speed bipolar PROM.

The optional decode logic shown in Figure 31 takes advantage of the overlap between address and data of the 80286 bus cycle to generate advanced memory and I/O-select signals. This minimizes system performance degradation caused by address propagation and decode delays. In addition to selecting memory and I/O, the advanced selects may be used with configurations supporting local and system buses to enable the appropriate bus interface for each bus cycle. The COD/$\overline{\text{INTA}}$ and M/$\overline{\text{IO}}$ signals are applied to the decode logic to distinguish between interrupt, I/O, code and data bus cycles.

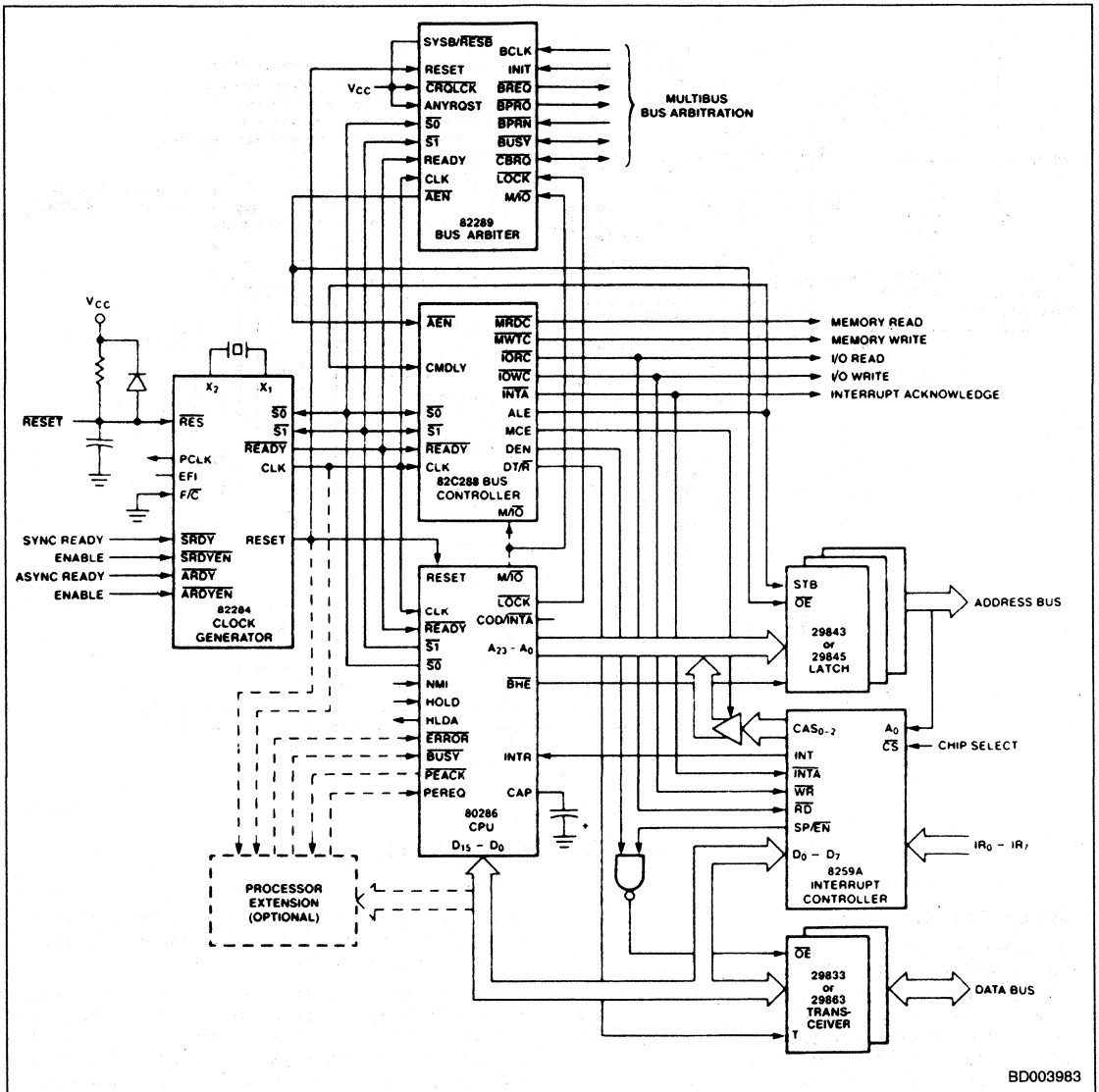


Figure 32. Multibus System Bus Interface

By adding the 82289 bus arbiter chip, the 80286 provides a Multibus system bus interface as shown in Figure 32. The ALE output of the 82C288 for the Multibus bus is connected to its CMDLY input to delay the start of commands one system CLK as required to meet Multibus address and write data set-up times. This arrangement will add at least one extra T_C state to each bus operation which uses the Multibus.

A second 82C288 bus controller and additional latches and transceivers could be added to the local bus of Figure 32. This configuration allows the 80286 to support an on-board bus for local memory and peripherals and the Multibus for system bus interfacing.

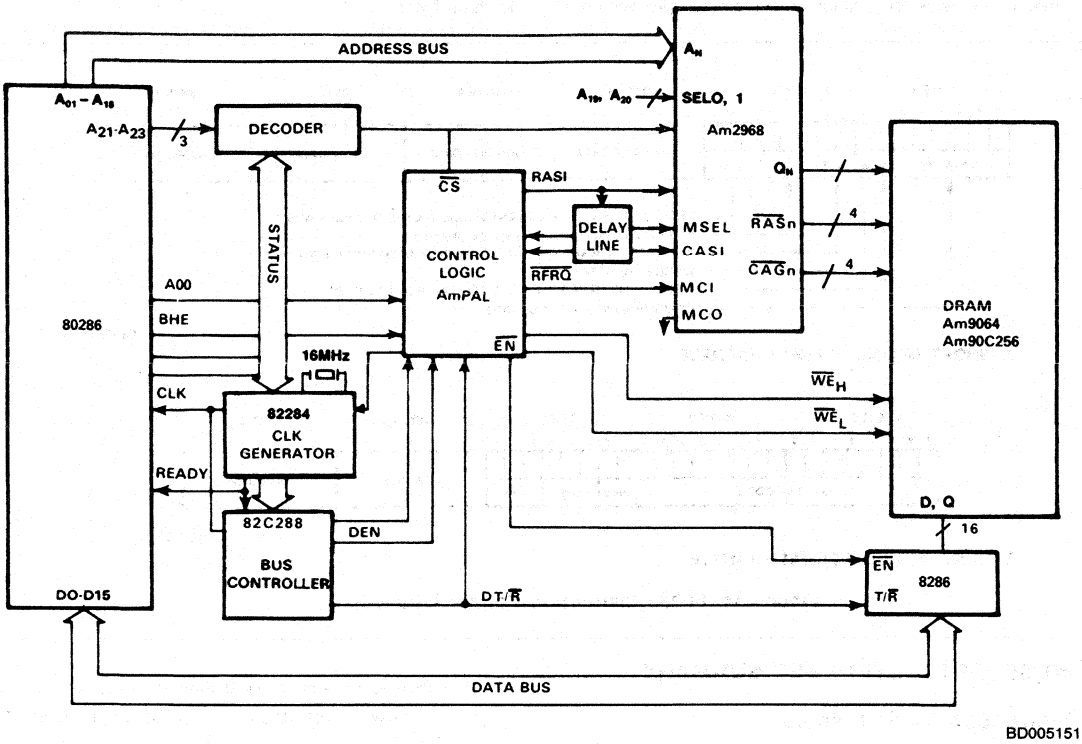


Figure 33.

Figure 33 shows the interface of the 80286 with the Am2968 Dynamic Memory Controller. The interface is a timing controller which consists of some control logic and a delay line. The timing controller runs asynchronously to the CPU. It arbitrates between memory requests and refresh requests by generating the proper signals to the dynamic memory controller and memory. The design described is a simple, cost-effective solution to interfacing the 80286 with the Am2968. A further description about DRAM selection based on processor speed may be found in the Am2968 Application Note.

Table 16. 80286 Systems Recommended Pull-up Resistor Values

80286 Pin and Name	Pull-up Value	Purpose
4 — $\overline{S1}$	20KΩ±10%	Pull $\overline{S0}$, $\overline{S1}$, and $\overline{PEACK}$ inactive during 80286 hold periods.
5 — $\overline{S0}$		
6 — $\overline{PEACK}$		
53 — $\overline{ERROR}$	20KΩ±10%	Pull $\overline{ERROR}$ and $\overline{BUSY}$ inactive when 80287 not present (or temporarily removed from socket).
54 — $\overline{BUSY}$		
63 — $\overline{READY}$	910Ω±5%	Pull $\overline{READY}$ inactive within required minimum time (C_L = 150pF, $I_R \leq$ 7mA).

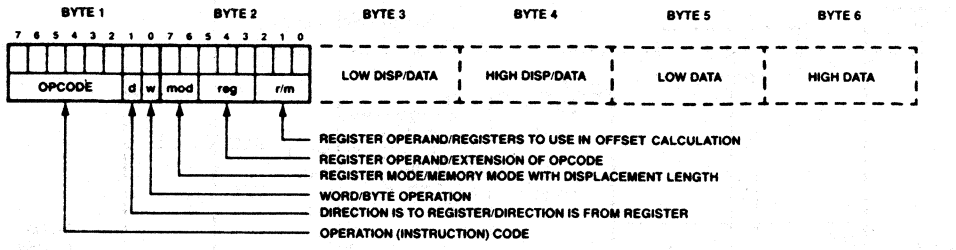
Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high level instructions, and processor control. These categories are summarized in Figures 4a – 4g.

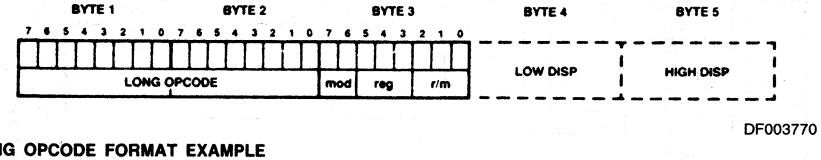
An 80286 instruction can reference zero, one, or two operands where an operand resides in a register, in the instruction itself, or in memory. Zero-operand instructions (e.g., NOP and HLT) are usually one byte long. One-operand instructions (e.g., INC and DEC) are usually two bytes long, but some are encoded in only one byte. One-operand instructions may reference a register or memory location. Two-operand instructions permit the following six types of instruction operations:

- Register to Register
- Memory to Register
- Immediate to Register
- Memory to Memory
- Register to Memory
- Immediate to Memory

Two-operand instructions (e.g. MOV and ADD) are usually three to six bytes long. Memory to memory operations are provided by a special class of string instructions requiring one to three bytes.



A. SHORT OPCODE FORMAT EXAMPLE



B. LONG OPCODE FORMAT EXAMPLE

Figure 34. 80286 Instruction Format Examples

80286 INSTRUCTION SET SUMMARY

Instruction Timing Notes

The instruction clock counts listed below establish the maximum execution rate of the 80286. With no delays in bus cycles, the actual clock count of an 80286 program will average 5% more than the calculated clock count, due to instruction sequences which execute faster than they can be fetched from memory.

To calculate elapsed times for instruction sequences, multiply the sum of all instruction clock counts, as listed in the table below, by the processor clock period. An 8 MHz processor clock has a clock period of 125 nanoseconds and requires an 80286 system clock (CLK input) of 16 MHz.

Instruction Clock Count Assumptions

1. The instruction has been prefetched, decoded, and is ready for execution. Control transfer instruction clock counts include all time required to fetch, decode, and prepare the next instruction for execution.
2. Bus cycles do not require wait states.
3. There are no processor extension data transfer or local bus HOLD requests.
4. No exceptions occur during instruction execution.

Instruction Set Summary Notes

Addressing displacements selected by the MOD field are not shown. If necessary they appear after the instruction fields shown.

Above/below refers to unsigned value
Greater refers to positive signed value
Less refers to less positive (more negative) signed values

if d = 1 then to register; if d = 0 then from register

if w = 1 then word instruction; if w = 0 then byte instruction

if s = 0 then 16-bit immediate data to form the operand

if s = 1 then an immediate data byte is sign-extended to form the 16-bit operand

x = don't care

z = used for string primitives for comparison with ZF FLAG

If two clock counts are given, the smaller refers to a register operand and the larger refers to a memory operand.

* = add one clock if offset calculation requires summing 3 elements

n = number of times repeated

m = number of bytes of code in next instruction

Level (L)—Lexical nesting level of the procedure

The following comments describe possible exceptions, side effects, and allowed usage for instructions in both operating modes of the 80286.

Real Address Mode Only

1. This is a protected mode instruction. Attempted execution in real address mode will result in an undefined opcode exception (6).
2. A segment overrun exception (13) will occur if a word operand reference at offset FFFF(H) is attempted.
3. This instruction may be executed in real address mode to initialize the CPU for protected mode.
4. The IOPL and NT fields will remain 0.
5. Processor extension segment overrun interrupt (9) will occur if the operand exceeds the segment limit.

Either Mode

6. An exception may occur, depending on the value of the operand.

7. $\overline{\text{LOCK}}$ is automatically asserted regardless of the presence or absence of the LOCK instruction prefix.

Protected Virtual Address Mode Only

8. The destination of an INT, JMP, CALL, RET or IRET instruction must be in the defined limit of a code segment or a general protection exception (13) occurs.

9. A general protection exception (13) will occur if the memory operand cannot be used due to either a segment limit or access rights violation. If a stack segment limit is violated, a stack segment overrun exception (12) occurs.

10. For segment load operations, the CPL, RPL, and DPL must agree with privilege rules to avoid an exception. The segment must be present to avoid a not-present exception (11). If the SS register is the destination, and a segment not-present violation occurs, a stack exception (12) occurs.

11. All segment descriptor accesses in the GDT or LDT made by this instruction will automatically assert $\overline{\text{LOCK}}$ to maintain descriptor integrity in multiprocessor systems.

12. JMP, CALL, INT, RET, IRET instructions referring to another code segment will cause a general protection exception (13) if any privilege rule is violated.

13. A general protection exception (13) occurs if CPL $\neq$ 0.

14. A general protection exception (13) occurs if CPL > IOPL.

15. The IF field of the flag word is not updated if CPL > IOPL. The IOPL field is updated only if CPL = 0.

16. Any violation of privilege rules as applied to the selector operand do not cause a protection exception; rather, the instruction does not return a result and the zero flag is cleared.

17. If the starting address of the memory operand violates a segment limit, or an invalid access is attempted, a general protection exception (13) will occur before the ESC instruction is executed. A stack segment overrun exception (12) will occur if the stack limit is violated by the operand's starting address. If a segment limit is violated during an attempted data transfer then a processor extension segment overrun exception (9) occurs.

18. The destination of an INT, JMP, CALL, RET, or IRET instruction must be in the defined limit of a code segment or a general protection exception (13) will occur.

80286 INSTRUCTION SET SUMMARY

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
DATA TRANSFER					
MOV = Move:					
Register to Register/Memory	1 0 0 0 1 0 0 w | mod reg r/m	2,3*	2,3*	2	9
Register/memory to register	1 0 0 0 1 0 1 w | mod reg r/m	2,5*	2,5*	2	9
Immediate to register/memory	1 1 0 0 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	2,3*	2,3*	2	9
Immediate to register	1 0 1 1 w reg | data | data if w = 1	2	2		
Memory to accumulator	1 0 1 0 0 0 0 w | addr-low | addr-high	5	5	2	9
Accumulator to memory	1 0 1 0 0 0 1 w | addr-low | addr-high	3	3	2	9
Register/memory to segment register	1 0 0 0 1 1 1 0 | mod 0 reg r/m	2,5*	17,19*	2	9,10,11
Segment register to register/memory	1 0 0 0 1 1 0 0 | mod 0 reg r/m	2,3*	2,3**	2	9
PUSH = Push:					
Memory	1 1 1 1 1 1 1 1 | mod 1 1 0 r/m	5*	5*	2	9
Register	0 1 0 1 0 reg	3	3	2	9
Segment register	0 0 0 reg 1 1 0	3	3	2	9
Immediate	0 1 1 0 1 0 s 0 | data | data if s = 0	3	3	2	9
PUSHA = Push All	0 1 1 0 0 0 0 0	17	17	2	9
POP = Pop:					
Memory	1 0 0 0 1 1 1 1 | mod 0 0 0 r/m	5*	5*	2	9
Register	0 1 0 1 1 reg	5	5	2	9
Segment register	0 0 0 reg 1 1 1 | (reg ≠ 01)	5	20	2	9,10,11
POPA = Pop All	0 1 1 0 0 0 0 1	19	19	2	9
XCHG = Exchange:					
Register/memory with register	1 0 0 0 0 1 1 w | mod reg r/m	3,5*	3,5*	2,7	7,9
Register with accumulator	1 0 0 1 0 reg	3	3		
IN = Input from:					
Fixed port	1 1 1 0 0 1 0 w | port	5	5		14
Variable port	1 1 1 0 1 1 0 w	5	5		14
OUT = Output to:					
Fixed port	1 1 1 0 0 1 1 w | port	3	3		14
Variable port	1 1 1 0 1 1 1 w	3	3		14
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1	5	5		9
LEA = Load EA to register	1 0 0 0 1 1 0 1 | mod reg r/m	3*	3*		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1 | mod reg r/m | (mod ≠ 11)	7*	21*	2	9,10,11
LES = Load pointer to ES	1 1 0 0 0 1 0 0 | mod reg r/m | (mod ≠ 11)	7*	21*	2	9,10,11
LAHF = Load AH with flags	1 0 0 1 1 1 1 1	2	2		
SAHF = Store AH into flags	1 0 0 1 1 1 1 0	2	2		
PUSHF = Push flags	1 0 0 1 1 1 0 0	3	3	2	9
POPF = Pop flags	1 0 0 1 1 1 0 1	5	5	2,4	9,15

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
ARITHMETIC					
ADD = Add:					
Reg/memory with register to either	0 0 0 0 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register / memory	1 0 0 0 0 0 s w | mod 0 0 0 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 0 0 1 0 w | data | data if w = 1	3	3		
ADC = Add with carry:					
Reg/memory with register to either	0 0 0 1 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 s w | mod 0 1 0 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 1 0 1 0 w | data | data if w = 1	3	3		
INC = Increment:					
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 0 r/m	2,7*	2,7*	2	9
Register	0 1 0 0 0 reg	2	2		
SUB = Subtract:					
Reg/memory and register to either	0 0 1 0 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate from register/memory	1 0 0 0 0 0 s w | mod 1 0 1 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate from accumulator	0 0 1 0 1 1 0 w | data | data if w = 1	3	3		
SBB = Subtract with borrow:					
Reg/memory and register to either	0 0 0 1 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate from register/memory	1 0 0 0 0 0 s w | mod 0 1 1 r/m | data | data if s:w = 01	3,7*	3,7*	2	9
Immediate from accumulator	0 0 0 1 1 1 0 w | data | data if w = 1	3	3		
DEC = Decrement:					
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 1 r/m	2,7*	2,7*	2	9
Register	0 1 0 0 1 reg	2	2		
CMP = Compare:					
Register/memory with register	0 0 1 1 1 0 1 w | mod reg r/m	2,6*	2,6*	2	9
Register with register/memory	0 0 1 1 1 0 0 w | mod reg r/m	2,7*	2,7*	2	9
Immediate with register/memory	1 0 0 0 0 0 s w | mod 1 1 1 r/m | data | data if s:w = 01	3,6*	3,6*	2	9
Immediate with accumulator	0 0 1 1 1 1 0 w | data | data if w = 1	3	3		
NEG = Change sign	1 1 1 1 0 1 1 w | mod 0 1 1 r/m	2	7*	2	7
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1	3	3		
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1	3	3		
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1	3	3		
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1	3	3		
MUL = Multiply (unsigned)	1 1 1 1 0 1 1 w | mod 1 0 0 r/m				
Register-Byte		13	13		
Register-Word		21	21		
Memory-Byte		16*	16*	2	9
Memory-Word		24*	24*	2	9
IMUL = Integer multiply (signed)	1 1 1 1 0 1 1 w | mod 1 0 1 r/m				
Register-Byte		13	13		
Register-Word		21	21		
Memory-Byte		16*	16*	2	9
Memory-Word		24*	24*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
ARITHMETIC (Continued)					
IMUL = Integer immediate multiply (signed)	0 1 1 0 1 0 s 1 | mod reg r/m | data | data if s = 0	21,24*	21,24*	2	5
DIV = Divide (unsigned):	1 1 1 1 0 1 1 w | mod 1 1 0 r/m				
Register-Byte		14	14		
Register-Word		22	22		
Memory-Byte		17*	17*	2,6	6,9
Memory-Word		25*	25*	2,6	6,9
IDIV = Integer divide (signed):	1 1 1 1 0 1 1 w | mod 1 1 1 r/m				
Register-Byte		17	17		
Register-Word		25	25		
Memory-Byte		20*	20*	2	9
Memory-Word		28*	28*	2	9
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	16	16		
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	14	14		
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	2		
CWD = Convert word to double word	1 0 0 1 1 0 0 1	2	2		
LOGIC					
Shift/Rotate Instructions:					
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2,7*	2,7*	2	9
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m	5 + n,8 + n*	5 + n,8 + n*	2	9
Register/Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m | count	5 + n,8 + n*	5 + n,8 + n*	5	8
	TTT: 0 0 0 ROL; 0 0 1 ROR; 0 1 0 RCL; 0 1 1 RCR; 1 0 0 SHL/SAL; 1 0 1 SHR; 1 1 1 SAR				
AND = And:					
Reg/memory and register to either	0 0 1 0 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 1 0 0 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 0 0 1 0 w | data | data if w = 1	3	3		
TEST = And function to flags, no result:					
Register/memory and register	1 0 0 0 0 1 0 w | mod reg r/m	2,6*	2,6*	2	9
Immediate data and register/memory	1 1 1 1 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	3,6*	3,6*	2	9
Immediate data and accumulator	1 0 1 0 1 0 0 w | data | data if w = 1	3	3		
OR = Or:					
Reg/memory and register to either	0 0 0 0 1 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 0 0 1 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 0 0 1 1 0 w | data | data if w = 1	3	3		
XOR = Exclusive or:					
Reg/memory and register to either	0 0 1 1 0 0 d w | mod reg r/m	2,7*	2,7*	2	9
Immediate to register/memory	1 0 0 0 0 0 0 w | mod 1 1 0 r/m | data | data if w = 1	3,7*	3,7*	2	9
Immediate to accumulator	0 0 1 1 0 1 0 w | data | data if w = 1	3	3		
NOT = Invert register/memory	1 1 1 1 0 1 1 w | mod 0 1 0 r/m	2,7*	2,7*	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
STRING MANIPULATION:					
MOVS = Move byte/word	1 0 1 0 0 1 0 w	5	5	2	9
CMPS = Compare byte/word	1 0 1 0 0 1 1 w	8	8	2	9
SCAS = Scan byte/word	1 0 1 0 1 1 1 w	7	7	2	9
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w	5	5	2	9
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w	3	3	2	9
INS = Input byte/wd from DX port	0 1 1 0 1 1 0 w	5	5	2	9,14
OUTS = Output byte/wd to DX port	0 1 1 0 1 1 1 w	5	5	2	9,14
Repeated by count in CX					
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	5 + 4n	5 + 4n	2	9
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	5 + 9n	5 + 9n	2	9
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	5 + 8n	5 + 8n	2	9
LODS = Load string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 1 0 w	5 + 4n	5 + 4n	2	9
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	4 + 3n	4 + 3n	2	9
INS = Input string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 0 w	5 + 4n	5 + 4n	2	9,14
OUTS = Output string	1 1 1 1 0 0 1 0 | 0 1 1 0 1 1 1 w	5 + 4n	5 + 4n	2	9,14
CONTROL TRANSFER					
CALL = Call:					
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	7 + m	7 + m	2	8
Register memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	7 + m,11 + m*	7 + m,11 + m*	2	8,9
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset | segment selector	13 + m	26 + m	2	8,11,12
Protected Mode Only (Direct intersegment):					
Via call gate to same privilege level			41 + m		8,11,12
Via call gate to different privilege level, no parameters			82 + m		8,11,12
Via call gate to different privilege level, x parameters			86 + 4x + m		8,11,12
Via TSS			177 + m		8,11,12
Via task gate			182 + m		8,11,12
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m (mod ≠ 11)	16 + m	29 + m*	2	8,9,11,12
Protected Mode Only (Indirect intersegment):					
Via call gate to same privilege level			44 + m*		8,9,11,12
Via call gate to different privilege level, no parameters			83 + m*		8,9,11,12
Via call gate to different privilege level, x parameters			90 + 4x + m*		8,9,11,12
Via TSS			180 + m*		8,9,11,12
Via task gate			185 + m*		8,9,11,12
JMP = Unconditional jump:					
Short/long	1 1 1 0 1 0 1 1 | disp-low	7 + m	7 + m		8
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	7 + m	7 + m		8
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	7 + m,11 + m*	7 + m,11 + m*	2	8,9
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset | segment selector	11 + m	23 + m		8,11,12
Protected Mode Only (Direct intersegment):					
Via call gate to same privilege level			38 + m		8,11,12
Via TSS			175 + m		8,11,12
Via task gate			180 + m		8,11,12
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m (mod ≠ 11)	15 + m*	26 + m*	2	8,9,11,12

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
CONTROL TRANSFER (Continued):					
Protected Mode Only (Indirect Intersegment):					
Via call gate to same privilege level			41 + m*		8,9,11,12
Via TSS			178 + m*		8,9,11,12
Via task gate			183 + m*		8,9,11,12
RET = Return from CALL:					
Within segment	1 1 0 0 0 0 1 1	11 + m	11 + m	2	8,9
Within seg adding immed to SP	1 1 0 0 0 0 1 0 | data-low | data-high	11 + m	11 + m	2	8,9
Intersegment	1 1 0 0 1 0 1 1	15 + m	25 + m	2	8,9,11,12
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0 | data-low | data-high	15 + m		2	8,9,11,12
Protected Mode Only (RET):					
To different privilege level			55 + m		
JE/JZ = Jump on equal zero	0 1 1 1 0 1 0 0 | disp	7 + m or 3	7 + m or 3		8
JL/JNGE = Jump on less not greater or equal	0 1 1 1 1 1 0 0 | disp	7 + m or 3	7 + m or 3		8
JLE/JNG = Jump on less or equal not greater	0 1 1 1 1 1 1 0 | disp	7 + m or 3	7 + m or 3		8
JB/JNAE = Jump on below not above or equal	0 1 1 1 0 0 1 0 | disp	7 + m or 3	7 + m or 3		8
JBE/JNA = Jump on below or equal not above	0 1 1 1 0 1 1 0 | disp	7 + m or 3	7 + m or 3		8
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0 | disp	7 + m or 3	7 + m or 3		8
JO = Jump on overflow	0 1 1 1 0 0 0 0 | disp	7 + m or 3	7 + m or 3		8
JS = Jump on sign	0 1 1 1 1 0 0 0 | disp	7 + m or 3	7 + m or 3		8
JNE/JNZ = Jump on not equal not zero	0 1 1 1 0 1 0 1 | disp	7 + m or 3	7 + m or 3		8
JNL/JGE = Jump on not less greater or equal	0 1 1 1 1 1 0 1 | disp	7 + m or 3	7 + m or 3		8
JNLE/JG = Jump on not less or equal greater	0 1 1 1 1 1 1 1 | disp	7 + m or 3	7 + m or 3		8
JNB/JAE = Jump on not below above or equal	0 1 1 1 0 0 1 1 | disp	7 + m or 3	7 + m or 3		8
JNBE/JA = Jump on not below or equal above	0 1 1 1 0 1 1 1 | disp	7 + m or 3	7 + m or 3		8
JNP/JPO = Jump on not par / par odd	0 1 1 1 1 0 1 1 | disp	7 + m or 3	7 + m or 3		8
JNO = Jump on not overflow	0 1 1 1 0 0 0 1 | disp	7 + m or 3	7 + m or 3		8
JNS = Jump on not sign	0 1 1 1 1 0 0 1 | disp	7 + m or 3	7 + m or 3		8
LOOP = Loop CX times	1 1 1 0 0 0 1 0 | disp	8 + m or 4	8 + m or 4		8
LOOPZ/LOOPE = Loop while zero equal	1 1 1 0 0 0 0 1 | disp	8 + m or 4	8 + m or 4		8
LOOPNZ/LOOPNE = Loop while not zero equal	1 1 1 0 0 0 0 0 | disp	8 + m or 4	8 + m or 4		8
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1 | disp	8 + m or 4	8 + m or 4		8
ENTER = Enter Procedure	1 1 0 0 1 0 0 0 | data-low | data-high | L				
L = 0		11	11	2	9
L = 1		15	15	2	9
L > 1		16 + 4(L − 1)	16 + 4(L − 1)	2	9
LEAVE = Leave Procedure	1 1 0 0 1 0 0 1	5	5	2	9

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
CONTROL TRANSFER (Continued):					
INT = Interrupt:					
Type specified	11001101 | type	23 + m		2	
Type 3	11001100	23 + m		2	
INTO = Interrupt on overflow	11001110	24 - m or 3 (3 if no) (Interrupt)	24 - or 3 (3 if no) (Interrupt)	2	
Protected Mode Only:					
Via interrupt or trap gate to same privilege level			40 + m		8,11,12
Via interrupt or trap gate to fit different privilege level			78 + m		8,11,12
Via Task Gate			167 + m		8,11,12
IRET = Interrupt return	11001111	17 + m	31 + m	2,4	8,9,11,12,15
Protected Mode Only:					
To different privilege level			55 + m		8,9,11,12,15
To different task (NT = 1)			169 + m		8,9,11,12
BOUND = Detect value out of range	01100010 | mod reg r/m	13*	13* (Use INT clock count if exception 5)	2,6	6,8,9,11,12
PROCESSOR CONTROL					
CLC = Clear carry	11111000	2	2		
CMC = Complement carry	11110101	2	2		
STC = Set carry	11111001	2	2		
CLD = Clear direction	11111100	2	2		
STD = Set direction	11111101	2	2		
CLI = Clear interrupt	11111010	3	3		14
STI = Set interrupt	11111011	2	2		14
HLT = Halt	11110100	2	2		13
WAIT = Wait	10011011	3	3		
LOCK = Bus lock prefix	11110000	0	0		14
CTS = Clear task switched flag	00001111 | 00000110	2	2	3	13
ESC = Processor Extension Escape	11011TTT | mod LLL r/m (TTT LLL are opcode to processor extension)	9-20*	9-20*	5	17

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

80286 INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	CLOCK COUNT: Real Address Mode	CLOCK COUNT: Protected Virtual Address Mode	COMMENTS: Real Address Mode	COMMENTS: Protected Virtual Address Mode
PROTECTION CONTROL					
LGDT = Load global descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 1 0 r/m	11*	11*	2,3	9,13
SGDT = Store global descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 0 0 r/m	11*	11*	2,3	9
LIDT = Load interrupt descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 1 1 r/m	12*	12*	2,3	9,13
SIDT = Store interrupt descriptor table register	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 0 0 1 r/m	12*	12*	2,3	9
LLDT = Load local descriptor table register from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 1 0 r/m		17,19*	1	9,11,13
SLDT = Store local descriptor table register to register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 0 0 r/m		2,3*	1	9
LTR = Load task register from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 1 1 r/m		17,19*	1	9,11,13
STR = Store task register to register memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 0 0 1 r/m		2,3*	1	9,11,13
LMSW = Load machine status word from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 1 1 0 r/m	3,6*	3,6*	2,3	9,13
SMSW = Store machine status word	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 1 | mod 1 0 0 r/m	2,3*	2,3*	2,3	9
LAR = Load access rights from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 1 0 | mod reg r/m		14,16*	1	9,16
LSL = Load segment limit from register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 1 1 | mod reg r/m		14,16*	1	9,16
ARPL = Adjust requested privilege level from register/memory	0 1 1 0 0 0 1 1 | mod reg r/m		10*,11*	2	9
VERR = Verify read access: register/memory	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 1 0 0 r/m		14,16*	1	9,16
VERR = Verify write access:	0 0 0 0 1 1 1 1 | 0 0 0 0 0 0 0 0 | mod 1 0 1 r/m		14,16*	1	9,16

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

Footnotes

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)		8-Bit (w = 0)	
000	AX	000	AL
001	CX	001	CL
010	DX	010	DL
011	BX	011	BL
100	SP	100	AH
101	BP	101	CH
110	SI	110	DH
111	DI	111	BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground..... −1.0 to +7.0V
Power Dissipation ..... 3.3 Watts

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
80286-6	T_A = 0°C to 55°C	5.0V ±5%
80286-8		

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V±5%)

Parameters	Description	Test Conditions	6MHz		8MHz		Units
			−6 Min	−6 Max	−8 Min	−8 Max	
V_{IL}	Input LOW Voltage		−.5	.8	−.5	.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		−.5	.6	−.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage	I_{OL} = 2.0mA		.45		.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = −400µA	2.4		2.4		V
I_{LI}	Input Leakage Current	$0V \leq V_{IN} \leq V_{CC}$		±10		±10	mA
I_{LO}	Output Leakage Current	$.45V \leq V_{OUT} \leq V_{CC}$		±10		±10	µA
I_{CC}	Supply Current (turn on, 0°C)	Note 1		600		600	µA
C_{CLK}	CLK Input Capacitance	F_C = 1MHz		20		20	pF
C_{IN}	Other Input Capacitance	F_C = 1MHz		10		10	pF
C_O	Input/Output Capacitance	F_C = 1MHz		20		20	pF
I_{LO}	Output Leakage Current	$0V \leq V_{out} < .045V$		±1		±1	mA
I_{IL}	Input Sustaining Current on $\overline{BUSY}$ and $\overline{ERROR}$ pins	V_{in} = 0V	30	500	30	500	µA

Notes: 1. Low temperature is worst case.

SWITCHING CHARACTERISTICS (T_A = 0°C to 55°C)

AC timings are referenced to 0.8V and 2.0V points of signals as illustrated in datasheet waveforms, unless otherwise noted.

Parameters	Description	Test Conditions	6MHz		8MHz		Units
			-6 Min	-6 Max	-8 Min	-8 Max	
1	System Clock (CLK) Period		83	250	62	250	ns
2	System Clock (CLK) LOW Time	at 0.6V	20	225	15	225	ns
3	System Clock (CLK) HIGH Time	at 3.2V	25	230	25	235	ns
17	System Clock (CLK) Rise Time	1.0V to 3.5V		10		10	ns
18	System Clock (CLK) Fall Time	3.5V to 1.0V		10		10	ns
4	Asynch. Inputs Set-up Time	Note 1	30		20		ns
5	Asynch. Inputs Hold Time	Note 1	30		20		ns
6	RESET Set-up Time		33		28		ns
7	RESET Hold Time		5		5		ns
8	Read Data Set-up Time		20		10		ns
9	Read Data Hold Time		8		8		ns
10	$\overline{READY}$ Set-up Time		50		38		ns
11	$\overline{READY}$ Hold Time		35		25		ns
12	Status/$\overline{PEACK}$ Valid Delay	Note 2 Note 3	1	55	1	40	ns
13	Address Valid Delay	Note 2 Note 3	1	80	1	60	ns
14	Write Data Valid Delay	Note 2 Note 3	0	65	0	50	ns
15	Address/Status/Data Float Delay	Note 2 Note 4	0	80	0	50	ns
16	HLDA Valid Delay	Note 2 Note 3	0	80	0	50	ns

Notes: 1. Asynchronous inputs are INTR, NMI, HOLD PEREQ, $\overline{ERROR}$, and $\overline{BUSY}$. This specification is given only for testing purposes to assure recognition at a specific CLK edge.
2. Delay from 0.8V on the CLK to 0.8V or 2.0V or float on the output as appropriate for valid or floating condition.
3. Output load: C1 = 100pF.
4. Float condition occurs when output current is less than I_{LO} in magnitude.

82284 Timing Requirements

Parameters	Description	Test Conditions	82284-6		82284-8		Units
			Min	Max	Min	Max	
11	$\overline{SRDY}$/$\overline{SRDYEN}$ Set-up Time		25		15		ns
12	$\overline{SRDY}$/$\overline{SRDYEN}$ Hold Time		0		0		ns
13	$\overline{ARDY}$/$\overline{ARDYEN}$ Set-up Time	See Note 1	5		0		ns
14	$\overline{ARDY}$/$\overline{ARDYEN}$ Hold Time	See Note 1	30		16		ns
19	PCLK Delay	C_L = 75pF I_{OL} = 5mA I_{OH} = -1mA	0	45	0	45	ns

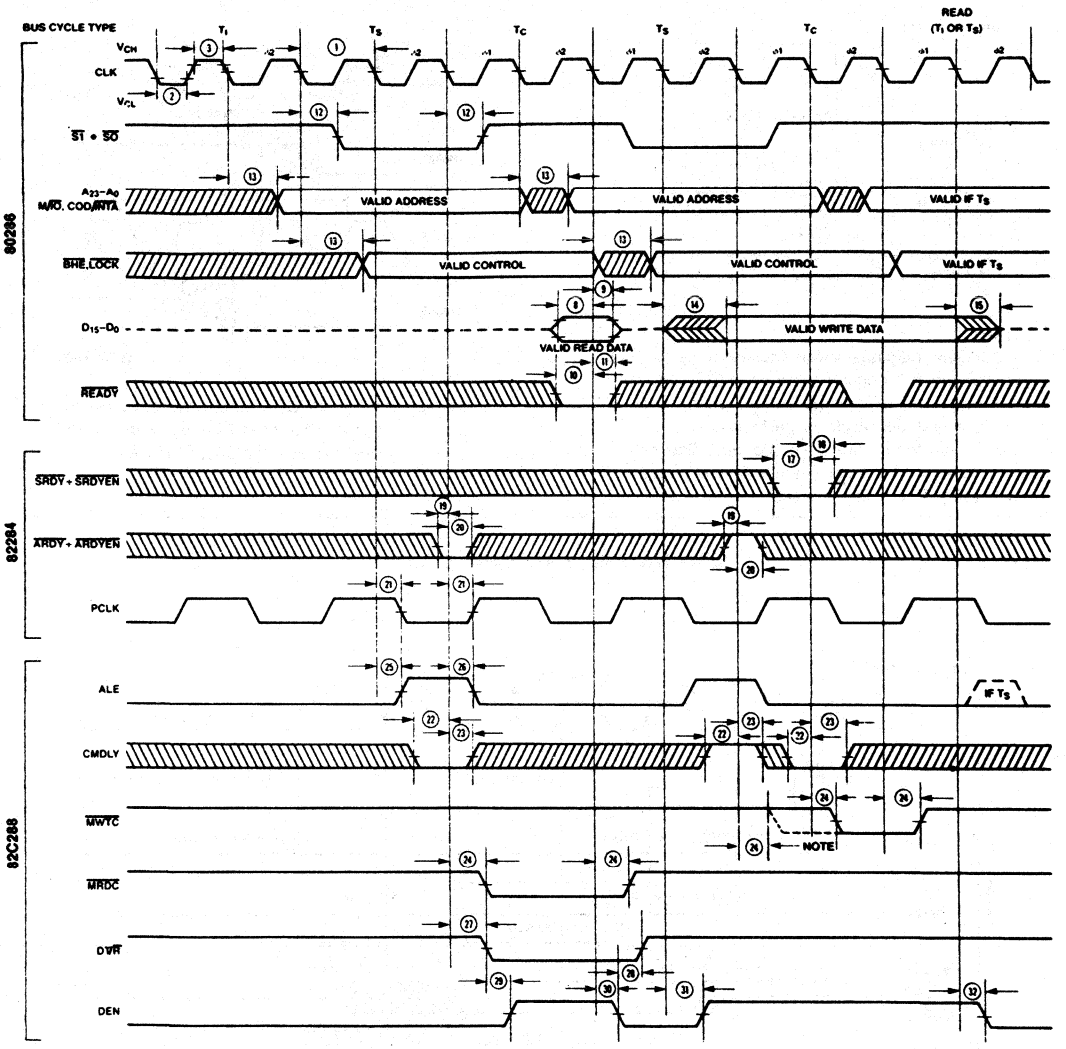
Note 1. These times are given for testing purposes to assure a predetermined action.

82C288 Timing Requirements

Parameters	Description		Test Conditions	82C288-6		82C288-8		Units
				Min	Max	Min	Max	
12	CMDLY Set-up Time			25		20		ns
13	CMDLT Hold Time			0		0		ns
30	Command Delay from CLK	Command Inactive	C_L = 300pF Max I_{OL} = 32mA Max I_{OH} = 5mA Max	3	30	3	20	ns
29		Command Active		3	40	3	20	
16	ALE Active Delay		C_L = 150pF I_{OL} = 16mA Max I_{OH} = -1mA Max	3	25	3	15	ns
17	ALE Inactive Delay				35		20	ns
19	DT/$\overline{R}$ Read Active Delay				40	0	20	ns
22	DT/$\overline{R}$ Read Inactive Delay			5	45	10	40	ns
20	DEN Read Active Delay			10	50	10	40	ns
21	DEN Read Inactive Delay			3	40	3	15	ns
23	DEN Write Active Delay				35		30	ns
24	DEN Write Inactive Delay			3	35	3	30	ns

SWITCHING WAVEFORMS

MAJOR CYCLE TIMING

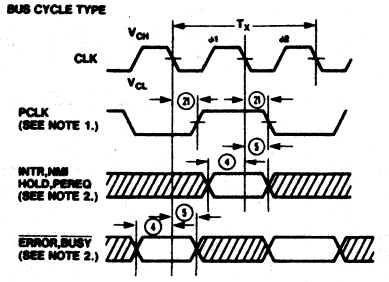


WF007981

Note: 1. $\overline{MWTC}$ is valid at this point only if CMDLY is low.

SWITCHING WAVEFORMS (Continued)

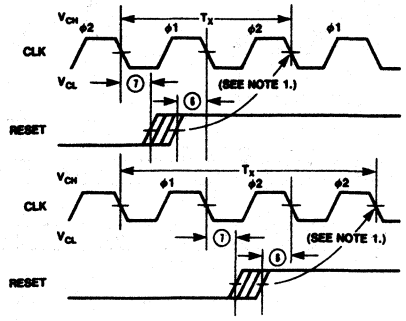
80286 ASYNCHRONOUS INPUT SIGNAL TIMING



WF009930

Notes:
1. PCLK indicates which processor cycle phase will occur on the next CLK. PCLK may not indicate the correct phase until the first bus cycle is performed.
2. These inputs are asynchronous. The set-up and hold times shown assure recognition for testing purposes.

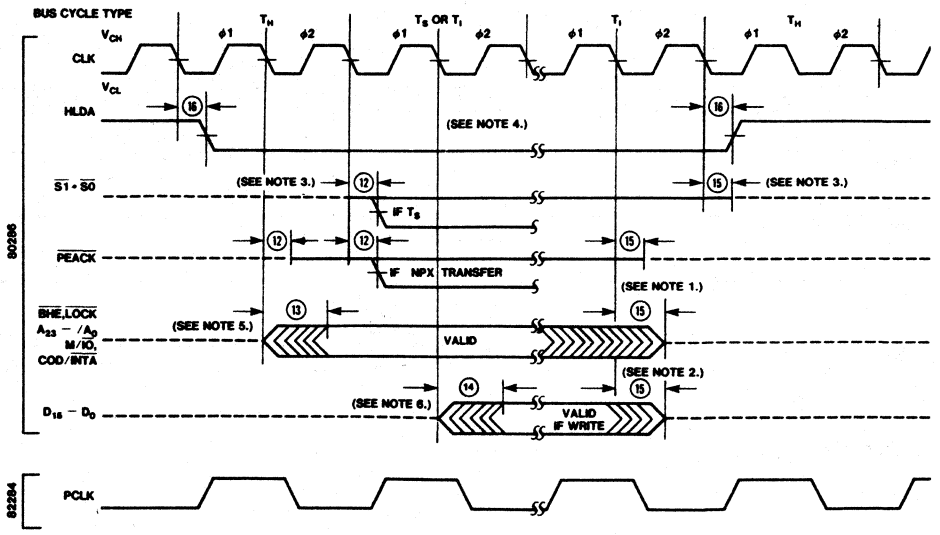
80286 RESET INPUT TIMING AND SUBSEQUENT PROCESSOR CYCLE PHASE



WF007930

Note 1: When RESET meets the set-up time shown, the next CLK will start or repeat $\phi 1$ of a processor cycle.

EXITING AND ENTERING HOLD

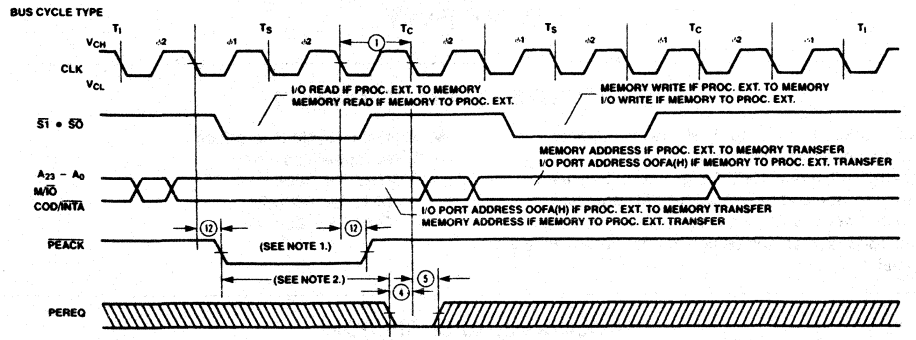


WF009941

Notes: 1. These signals may not be driven by the 80286 during the time shown. The worst case in terms of latest float time is shown.
2. The data bus will be driven as shown if the last cycle before T_I in the diagram was a write T_C.
3. The 80286 floats its status pins during T_H. External 20KΩ resistors keep these signals high (see Table 16).
4. For HOLD request set-up to HLDA, refer to Figure 29.
5. $\overline{BHE}$ and $\overline{LOCK}$ are driven at this time but will not become valid until T_S.
6. The data bus will remain in 3-state OFF if a read cycle is performed.

SWITCHING WAVEFORMS (Continued)

80286 PEREQ/$\overline{\text{PEACK}}$ TIMING REQUIRED PEREQ TIMING FOR ONE TRANSFER ONLY



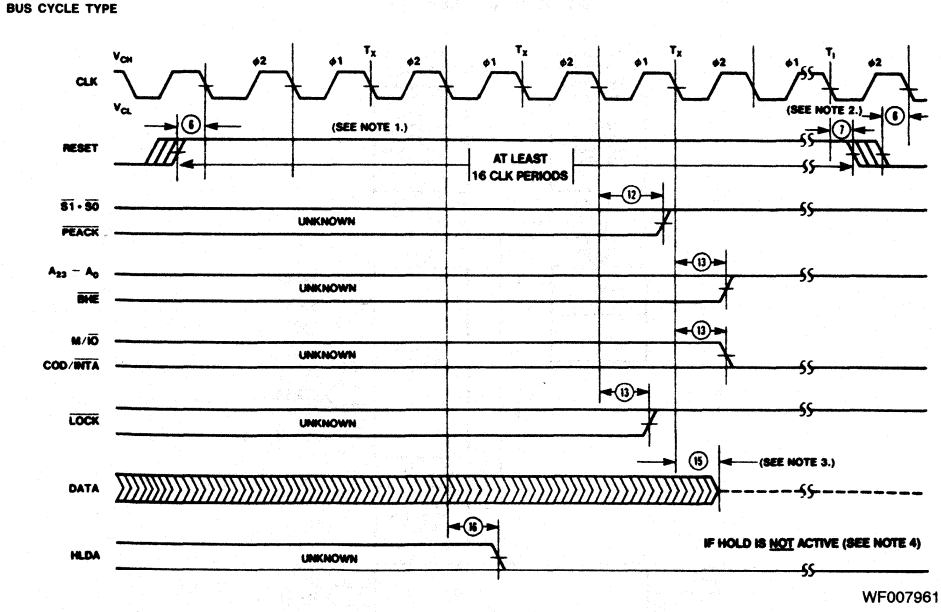
WF007952

ASSUMING WORD-ALIGNED MEMORY OPERAND; IF ODD ALIGNED, 80286 TRANSFERS TO/FROM MEMORY BYTE-AT-A-TIME WITH TWO MEMORY CYCLES.

Notes: 1. $\overline{\text{PEACK}}$ always goes active during the first bus operation of a processor extension data operand transfer sequence. The first bus operation will be either a memory read at operand address or I/O read at port address OOFA(H).

2. To prevent a second processor extension data operand transfer, the worst case maximum time (Shown above) is: $3 \times ① - ⑪_{max} - ④_{min}$. The actual, configuration dependent, maximum time is: $3 \times ① - ⑪_{max} - ④_{min} + A \times 2 \times ①$. A is the number of extra T_C states added to either the first or second bus operation of the processor extension data operand transfer sequence.

INITIAL 80286 PIN STATE DURING RESET



WF007961

Notes: 1. Set-up time for RESET ↑ may be violated with the consideration that $\phi 1$ of the processor clock may begin one system CLK period later.

2. Set-up and hold times for RESET ↓ must be met for proper operation, but RESET ↓ may occur during $\phi 1$ or $\phi 2$.

3. The data bus is only guaranteed to be in 3-state OFF at the time shown.

4. HOLD is acknowledged during RESET, causing HLDA to go active and the appropriate pins to float. If HOLD remains active while RESET goes inactive, the 80286 remains in HOLD state and will not perform any bus accesses until HOLD is deactivated.

8086

16-Bit Microprocessor
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Directly addresses up to 1 Mbyte of memory
- 24 operand addressing modes
- Efficient implementation of high level languages
- Instruction set compatible with 8080 software
- Bit, byte, word, and block operations
- 8 and 16-bit signed and unsigned arithmetic in binary or decimal
- Multibus* system interface
- Three speed options
  - 5MHz for 8086
  - 8MHz for 8086-2
  - 10MHz for 8086-1

GENERAL DESCRIPTION

The 8086 is a general purpose 16-bit microprocessor CPU. Its architecture is built around thirteen 16-bit registers and nine 1-bit flags. The CPU operates on 16-bit address spaces and can directly address up to 1 megabyte using offset addresses within four distinct memory segments, designated as code, data, stack and extra code. The 8086 implements a powerful instruction set with 24 operand addressing modes. This instruction set is compatible with that of the 8080 and 8085. In addition, the 8086 is particularly effective in executing high level languages.

The 8086 can operate in minimum and maximum modes. Maximum mode offloads certain bus control functions to a peripheral device and allows the CPU to operate efficiently in a multi-processor system. The CPU and its high performance peripherals are Multibus* compatible. The 8086 is implemented in N-channel, depletion load, silicon gate technology and is contained in a 40-pin CERDIP package, Molded DIP package, or Plastic Leaded Chip Carrier.

BLOCK DIAGRAM

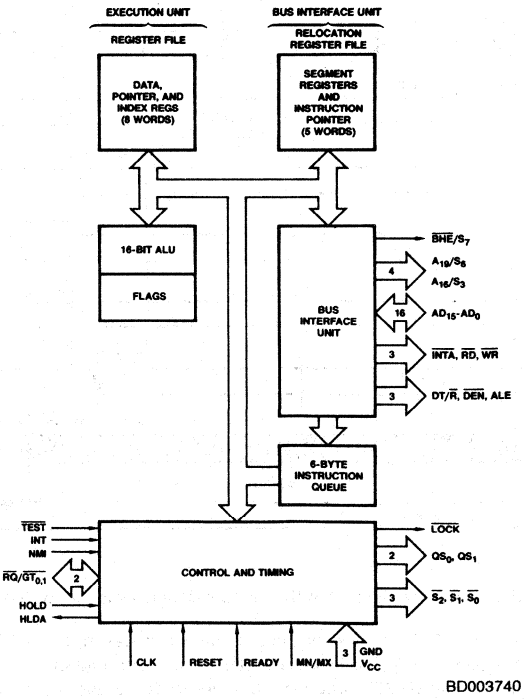


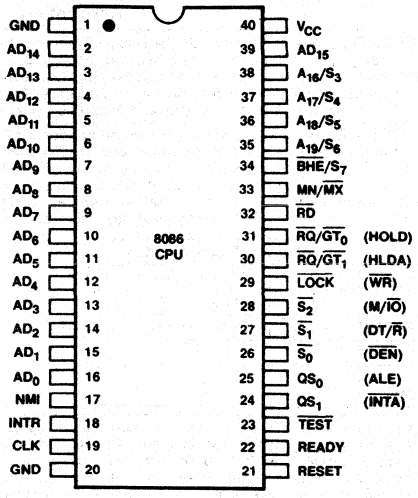
Figure 1.

Multibus is a registered trademark of Intel Corp.

01966B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40-1, P-40-1



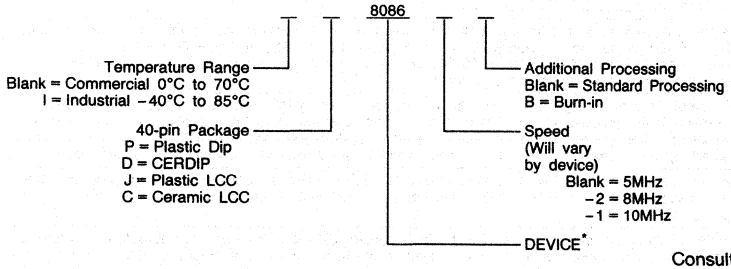
CD005510

Note: Pin 1 is marked for orientation
Also available in PLCC. See Section 7 for pinout details.

Figure 2.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations	
8086 8086-2 8086B 8086-2B	P, D, ID, J
8086-1 8086-1B	P, D, J
8086-2 8086	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

The following pin function descriptions are for 8086 systems in either minimum or maximum mode. The "Local Bus" in these descriptions is the direct multiplexed bus interface connection to the 8086 (without regard to additional bus buffers).

Pin No.	Name	I/O	Description
39, 2-16	AD_{15}-AD_0	I/O	Address Data Bus. These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, T_W, T_4) bus. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7-D_0. It is LOW during T_1 when a byte is to be transferred on the lower portion of the bus in memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use A_0 to condition chip select functions. (See $\overline{BHE}$.) These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
35-38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	O	Address/Status. During T_1 these are the four most significant address lines for memory operations. During I/O operations these lines are LOW. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W, and T_4. The status of the interrupt enable FLAG bit (S_5) is updated at the beginning of each CLK cycle. A_{17}/S_4 and A_{16}/S_3 are encoded as shown. This information indicates which relocation register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge."
34	$\overline{BHE}/S_7$	O	Bus High Enable/Status. During T_1 the bus high enable signal ($\overline{BHE}$) should be used to enable data onto the most significant half of the data bus, pins D_{15}-D_8. Eight-bit oriented devices tied to the upper half of the bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is LOW during T_1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S_7 status information is available during T_2, T_3, and T_4. The signal is active LOW and floats to 3-state OFF in "hold." It is LOW during T_1 for the first interrupt acknowledge cycle.
32	$\overline{RD}$	O	Read. Read strobe indicates that the processor is performing a memory of I/O read cycle, depending on the state of the S_2 pin. This signal is used to read devices which reside on the 8086 local bus. $\overline{RD}$ is active LOW during T_2, T_3 and T_W of any read cycle and is guaranteed to remain HIGH in T_2 until the 8086 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge."
22	READY	I	READY. Is the acknowledgment from the addressed memory or I/O device that it will complete the data transfer. The READY signal from memory/IO is synchronized by the 8284A Clock Generator to form READY. This signal is active HIGH. The 8086 READY input is not synchronized. Correct operation is not guaranteed if the set-up and hold times are not met.
18	INTR	I	Interrupt Request. Is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
23	$\overline{TEST}$	I	$\overline{TEST}$. Input is examined by the "Wait" instruction. If the $\overline{TEST}$ input is LOW, execution continues; otherwise, the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
17	NMI	I	Non-Maskable Interrupt. An edge-triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This input is internally synchronized.
21	RESET	I	Reset. Causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the Instruction Set description, when RESET returns LOW. RESET is internally synchronized.
19	CLK	I	Clock. Provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		V_{CC}. The + 5V power supply pin.
1, 20	GND		Ground. The ground pin.
33	MN/$\overline{MX}$	I	Minimum/Maximum. Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.

A_{17}/S_4	A_{16}-S_3	Characteristics
0 (LOW)	0	Alternate Data
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data
S_6 is 0 (LOW)		

$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/ to odd address
1	0	Lower byte from/ to even address
1	1	None

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
28-26	$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$	O	Status. Active during T_4, T_1, and T_2 and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. Any change by S_2, S_1, or S_0 during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals float to 3-state OFF in "hold acknowledge." These status lines are encoded as shown.

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

31, 30	$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$	I/O	Request/Grant. Pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT}_0$ having higher priority than $\overline{RQ}/\overline{GT}_1$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor so it may be left unconnected. The request/grant sequence is as follows: 1. A pulse of 1 CLK wide from another local bus master indicates a local bus request ("hold") to the 8086 (pulse 1). 2. During a T_4 or T_1 clock cycle, a pulse 1 CLK wide from the 8086 to the requesting master (pulse 2), indicates that the 8086 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge." 3. A pulse 1 CLK wide from the requesting master indicates to the 8086 (pulse 3) that the "hold" request is about to end and that the 8086 can reclaim the local bus at the next CLK. Each master-master exchange of the local bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met: 1. Request occurs on or before T_2. 2. Current cycle is not the low byte of a word (on an odd address). 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is idle when the request is made, two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
29	$\overline{LOCK}$	O	$\overline{LOCK}$. Output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW, and floats to 3-state OFF in "hold acknowledge."
24, 25	QS_1, QS_0	O	Queue Status. The queue status is valid during the CLK cycle after which the queue operation is performed. QS_1 and QS_0 provide status to allow external tracking of the internal 8086 instruction queue.
28	M/$\overline{IO}$	O	Status line. Logically equivalent to S_2 in the maximum mode. It is used to distinguish a memory access from an I/O access. M/IO becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (M = HIGH, IO = LOW). M/$\overline{IO}$ floats to 3-state OFF in local bus ""hold acknowledge.""
29	$\overline{WR}$	O	Write. Indicates that the processor is performing a write memory or write I/O cycle, depending on the state of M/IO signal. WR is active for T_2, T_3 and T_W of any write cycle. It is active LOW, and floats to 3-state OFF in local bus "hold acknowledge."
24	$\overline{INTA}$	O	$\overline{INTA}$. Is used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3 and T_W of each interrupt acknowlegde cycle.
25	ALE	O	Address Latch Enable. Provided by the processor to latch the address into 8282/8283 address latch. It is a HIGH pulse active during T_1 of any bus cycle. Note that ALE is never floated.
27	DT/$\overline{R}$	O	Data Transmit/Receive. Needed in minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/$\overline{R}$ is equivalent to $\overline{S}_1$ in the maximum mode, and its timing is the same as for M/$\overline{IO}$. (T = HIGH, R = LOW.) This signal floats to 3-state OFF in local bus "hold acknowledge."
26	$\overline{DEN}$	O	Data Enable. Provided as an output enable for the 8286/8287 in a minimum system which uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for INTA cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4, while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to 3-state OFF in local bus "hold acknowledge."

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
31, 30	HOLD, HLDA	I/O	HOLD. Indicates that another master is requesting a local bus "hold." To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor will LOWer HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. The same rules as for $\overline{RQ}/\overline{GT}$ apply, regarding when the local bus will be released. HOLD is not asynchroneous input. External synchronization should be provided if the system cannot otherwise guarantee the set-up time.

DETAILED DESCRIPTION

The 8086 CPU is internally organized into two processing units. These two units are the Bus Interface Unit (BIU) and the Execution Unit (EU). A block diagram of this organization is shown in Figure 1.

The BIU performs instruction fetch and queuing, operand fetch and store, address relocation, and basic bus control. The EU receives operands and instructions from the BIU and processes them on a 16-bit ALU. The EU accesses memory and peripheral devices through requests to the BIU. The BIU generates physical addresses in memory using the 4 segment registers and offset values.

The BIU and EU usually operate asynchronously. This permits the 8086 to overlap execution fetch and execution. Up to 6 instruction bytes can be queued. The instruction queue acts as a FIFO buffer for instructions, from which the EU extracts instruction bytes as required.

Memory Organization

The 8086 addresses up to 1 megabyte of memory. The address space is organized as a linear array, from 00000 to FFFFF in hexadecimal. Memory is subdivided into segments of 64K bytes each. There are 4 segments: code, stack, data, and extra (usually employed as an extra data segment). Each segment thus contains information of a similar type. Selection of a destination segment is automatically performed using the rules in the table below. This segmentation makes memory more easily relocatable and supports a more structured programming style.

Physical addresses in memory are generated by selecting the appropriate segment, obtaining the segment "base" address from the segment register, shifting the base address 4 digits to the left, and then adding this base to the "offset" address. For programming code, the offset address is obtained from the instruction pointer. For operands, the offset address is calculated in several ways, depending upon information contained in the addressing mode. Memory organization and address generation are shown in Figure 3a.

Certain memory locations are reserved for specific CPU operations. These are shown in Figure 3b. Addresses FFFFOH through FFFFFH are reserved for operations which include a jump to the initial program loading routine. After RESET, the CPU will always begin execution at location FFFFOH, where the jump must be located.

Addresses 00000H through 003FFH are reserved for interrupt operations. The service routine of each of the 256 possible interrupt types is signaled by a 4-byte pointer. The pointer elements must be stored in reserved memory addresses before the interrupts are invoked.

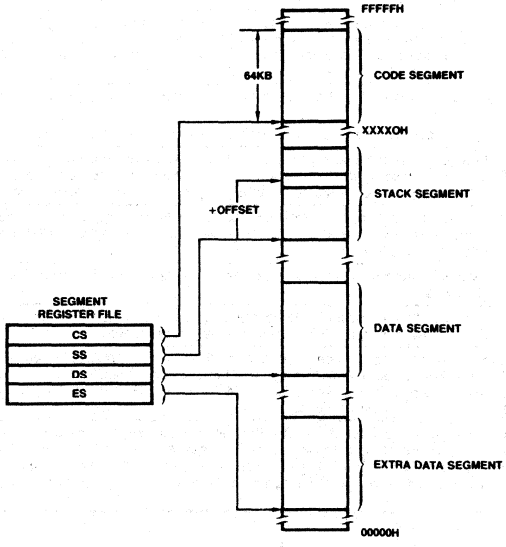


Figure 3a. Memory Organization

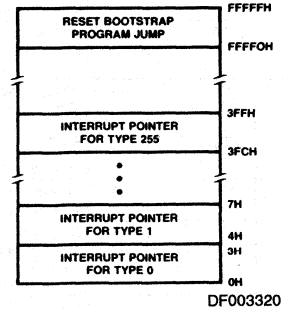


Figure 3b. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic for all prefetching of instructions.
Stack	STACK (SS)	All stack pushes and pops, and all memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references which are relative to the stack, the destination of a string operation, or explicitly overriden.
External (Global) Data	EXTRA (ES)	Destination of string operations, when they are explicitly selected using a segment override.

Minimum and Maximum Modes

The 8086 has two system configurations, minimum and maximum mode. The CPU has a strap pin, MN/$\overline{MX}$, which defines the system configuration. The status of this strap pin defines the function of pin numbers 24 through 31.

When MN/$\overline{MX}$ is strapped to GND, the 8086 operates in maximum mode. The operations of pins 24 through 31 are redefined. In maximum mode, several bus timing and control functions are "off-loaded" to the 8288 bus controller, thus freeing up the CPU. The CPU communicates status information to the 8288 through pins S_0, S_1, and S_2. In maximum mode, the 8086 can operate in a multiprocessor system, using the LOCK signal within a Multibus format.

When MN/$\overline{MX}$ is strapped to V_{CC}, the 8086 operates in minimum mode. The CPU sends bus control signals itself through pins 24 through 31. This is shown in Figure 2 (in parentheses). Examples of minimum and maximum mode systems are shown in Figure 4.

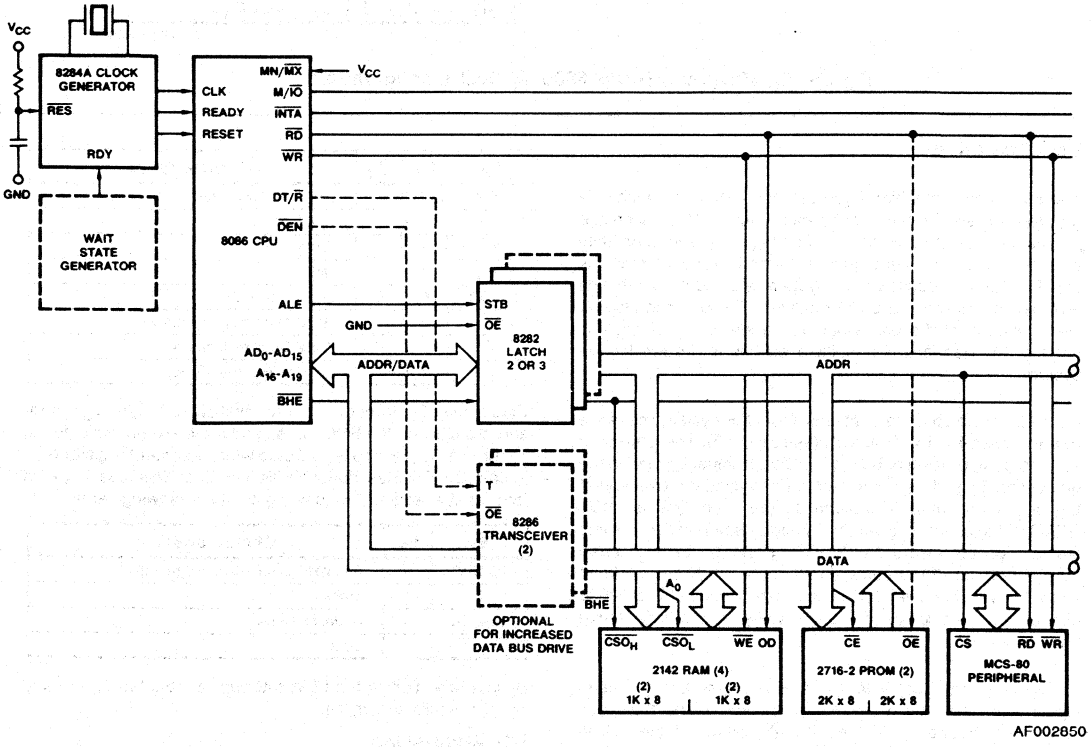


Figure 4a. Minimum Mode 8086 Typical Configuration

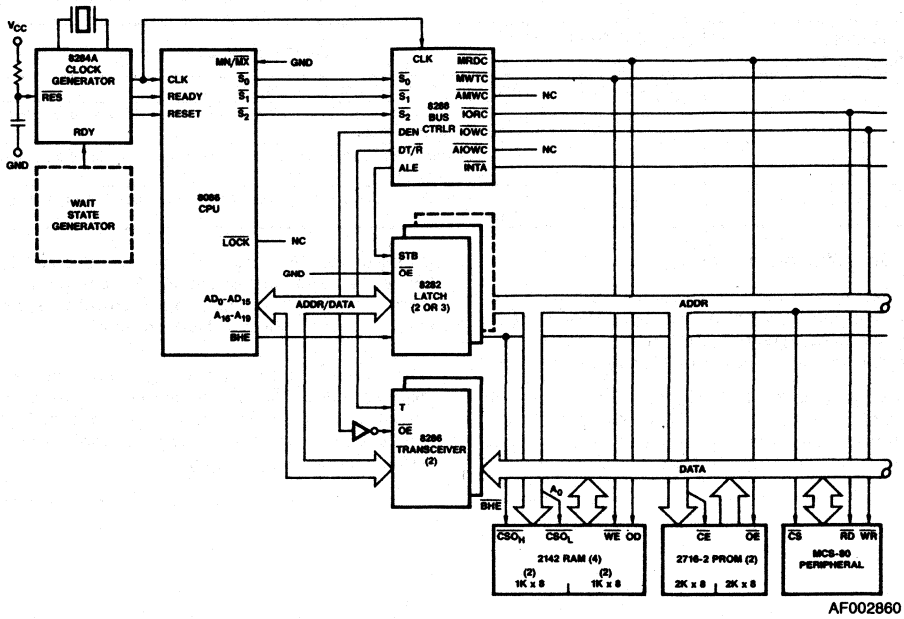


Figure 4b. Maximum Mode 8086 Typical Configuration

Bus Operation

The 8086 has a combined address and data bus, commonly referred to as "a time multiplexed bus." This technique provides the most efficient use of pins on the processor while permitting the use of a standard 40-lead package. This bus can be used throughout the system with address latching provided on memory and I/O modules. The bus can also be demultiplexed at the processor with a single set of address latches if a standard non-multiplexed bus is desired for the system.

Each bus cycle consists of at least four CLK cycles. These are referred to as T_1, T_2, T_3 and T_4 (see Figure 5). The address is sent from the processor during T_1. Data transfer occurs on the bus during T_3 and T_4. T_2 is used for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "Wait" states (T_W) are inserted between T_3 and T_4. Each inserted "Wait" state is of the same duration as a CLK cycle. "Idle" states (T_I) or inactive CLK cycles can occur between 8086 bus cycles. The processor uses these cycles for internal housekeeping.

During T_1 of any bus cycle, the ALE (Address Latch Enable) signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$ are used, in maximum mode, by the bus controller to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0(LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1(HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S_3 through S_7 are multiplexed with high-order address bits and the $\overline{BHE}$ signal, and are therefore valid during T_2 through T_4. S_3 and S_4 indicate which segment register (see Instruction Set description) was used for this bus cycle in forming the address, according to the following table:

S_4	S_3	Characteristics
0(LOW)	0	Alternate Data (extra segment)
0	1	Stack
1(HIGH)	0	Code or None
1	1	Data

S_5 is a reflection of the PSW interrupt enable bit. $S_6 = 0$ and S_7 is a spare status bit.

I/O Addressing

8086 I/O operations can address up to a maximum of 64K I/O byte registers or 32K I/O word registers. The I/O address appears in the same format as the memory address on bus lines $A_{15} - A_0$. The address lines $A_{19} - A_{16}$ are zero in I/O operations. I/O instructions which use register DX as a pointer have full address capability. Direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

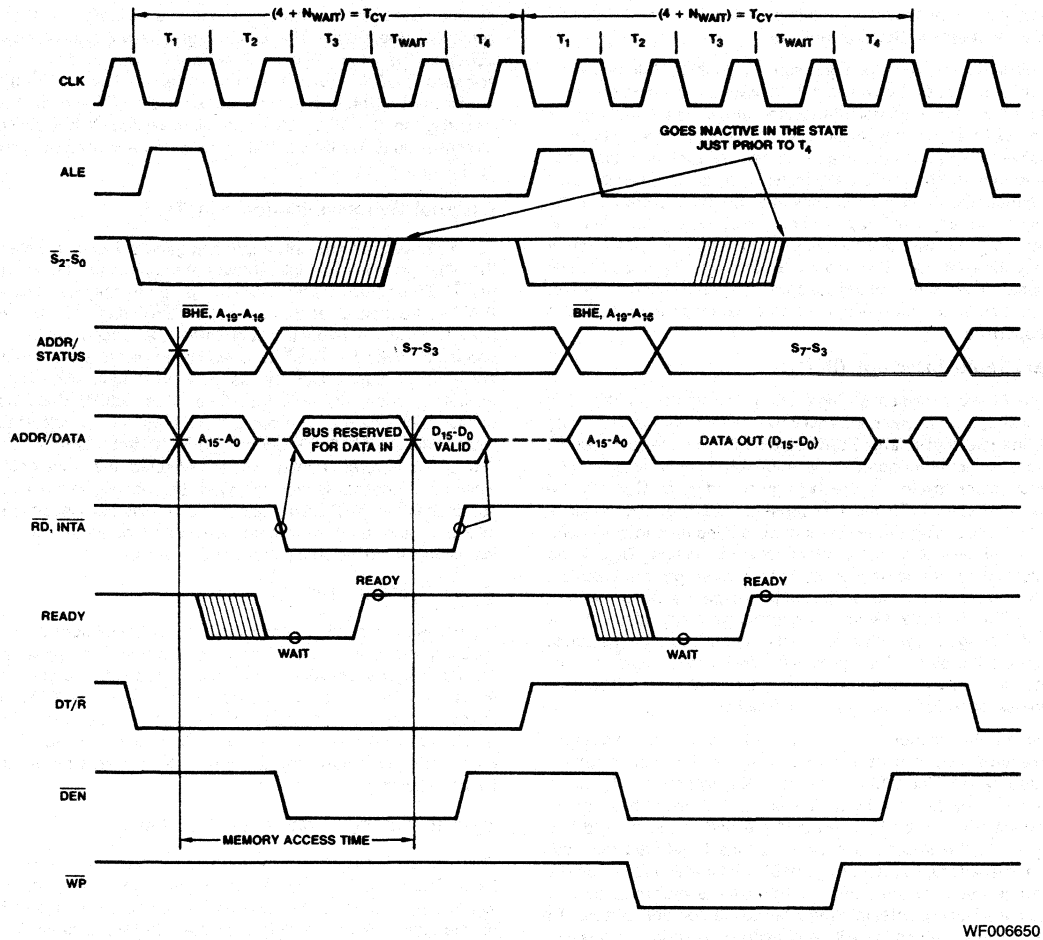


Figure 5. Basic System Timing

EXTERNAL INTERFACE

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8086 RESET is required to be HIGH for greater than 4 CLK cycles. The 8086 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 10 CLK cycles. After this interval the 8086 operates normally beginning with the instruction in absolute location FFFF0H (see Figure 3B). The details of this operation are explained in the Instruction Set description of the MCS-86 Family User's Manual. The RESET input is internally synchronized to the processor clock. At initialization the HIGH-to-LOW transition of RESET must occur no sooner than 50μs after power-up, to allow complete initialization of the 8086.

NMI may not be asserted prior to the 2nd CLK cycle following the end of RESET.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are described in the Instruction Set description. Hardware interrupts are either non-maskable or maskable.

Interrupts transfer control to a new program location. A 256-element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 3b), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number during the interrupt acknowledge sequence, which is used to "vector" through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt pin (NMI) which has higher priority than the maskable interrupt request pin (INTR). A typical use would be to activate a power

failure routine. The NMI is edge-triggered on a LOW-to-HIGH transition. The activation of this pin causes a type 2 interrupt. (See Instruction Set description.)

NMI is required to have a duration in the HIGH state of greater than two CLK cycles, but is not required to be synchronized to the clock. Any high-going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves of a block-type instruction. Worst case response to NMI would be to multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 86/10 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable FLAG status bit. The interrupt request signal is level-triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block-type instruction. During the interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single-step), although the FLAGS register, which is automatically pushed onto the stack, reflects the state of the processor prior to the Interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (Figure 6), the processor executes two successive (back-to-back) interrupt acknowledge cycles. The 8086 emits the LOCK signal from T_2 of the first bus cycle until T_2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The INTERRUPT RETURN instruction includes a FLAGS pop, which returns the status of the original interrupt enable bit when it restores the FLAGS.

HALT

When a software "HALT" instruction is executed, the processor indicates that it is entering the "HALT" state in one of two ways depending upon which mode is strapped. In minimum mode, the processor issues one ALE with no qualfying bus control signals. In Maximum Mode, the processor issues appropiate HALT status on $\overline{S}_2\overline{S}_1\overline{S}_0$, and the 8288 bus controller issues one ALE. The 8086 will not leave the "HALT" state when a local bus "hold" is entered while in "HALT." In this case, the processor reissues the HALT indicator. An interrupt request or RESET will force the 8086 out of the "HALT" state.

Read/Modify/Write (Semaphore) Operation Via Lock

The $\overline{LOCK}$ status information is provided by the processor when directly consecutive bus cycles are required during the execution of an instruction. This provides the processor with the capability of performing read/modify/write operations on memory (via the Exchange Register With Memory Instruction, for example) without the possibility of another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (forced LOW) in the clock cycle following the one in which the software "LOCK" prefix instruction is decoded by the EU. It is deactivated at the end of the last bus cycle of the instruction following the "LOCK" prefix instruction. While $\overline{LOCK}$ is active, a request on a RQ/GT pin will be recorded and then honored at the end of the LOCK.

External Synchronization Via Test

As an alternative to the interrupts and general I/O capabilities, the 8086 provides a single software-testable input known as the $\overline{TEST}$ signal. At any time, the program may execute a WAIT instruction. If at that time the $\overline{TEST}$ signal is inactive (HIGH), program execution becomes suspended while the processor waits for $\overline{TEST}$ to become active. It must remain active for at least 5 CLK cycles. The WAIT instruction is re-executed repeatedly until that time. This activity does not consume bus cycles. The processor remains in an idle state while waiting. All 8086 drivers go to 3-state OFF if bus "HOLD" is entered. If interrupts are enabled, they may occur while the processor is waiting. When this occurs, the processor fetches the WAIT instruction one extra time, processes the interrupt, and then re-fetches and re-executes the WAIT instruction upon returning from the interrupt.

Basic System Timing

Typical system configurations for the processor operating in minimum mode and in maximum mode are shown in Figures 4a and 4b, respectively. In minimum mode, the processor emits bus control signals in a manner similar to the 8085. In maximum mode, the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals. Figure 5 illustrates the signal timing relationships.

System Timing – Minimum System

The read cycle begins in T_1 with the assertion of the Address Latch Enable (ALE) signal. The trailing (low-going) edge of this signal is used to latch the address information, which is valid on the local bus at this time, into the 8282/8283 latch. The $\overline{BHE}$ and A_0 signals address the low, high, or both bytes. From T_1 to T_4, the M/$\overline{IO}$ signal indicates a memory or I/O operation. At T_2 the address is removed from the local bus and the bus goes to a high impedance state. The read control signal is also asserted at T_2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (8286/8287) is required to buffer the 8086 local bus, signals DT/$\overline{R}$ and $\overline{DEN}$ are provided by the 8086.

A write cycle also begins with the assertion of ALE and the emission of the address. The M/$\overline{IO}$ signal is again asserted to indicate a memory or I/O write operation. In the T_2 immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until the middle of T_4. During T_2, T_3, and T_W, the processor asserts the write control signal. The write ($\overline{WR}$) signal becomes active at the beginning of T_2 as opposed to the read which is delayed somewhat into T_2 to provide time for the bus to float.

The $\overline{BHE}$ and A_0 signals are used to select the proper byte(s) of the memory/IO word to be read or written according to the following table.

$\overline{BHE}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

I/O ports are addressed in the same manner as memory location. Even addressed bytes are transferred on the D_7-D_0 bus lines and odd addressed bytes on D_{15}-D_8.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge signal ($\overline{INTA}$) is asserted in place of the read ($\overline{RD}$) signal and the address bus is floated. (See Figure 6.) In the second of two successive INTA cycles, a byte of information is read from bus lines D_7-D_0 as supplied by the interrupt system logic (i.e., 8259A Priority Interrupt Controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into a interrupt vector lookup table, as described earlier.

Bus Timing – Medium Size Systems

For medium size systems, the MN/$\overline{MX}$ pin is connected to V_{SS}, and the 8288 Bus Controller is added to the system as well as an 8282/8283 latch for latching the system address and a 8286/8287 transceiver to allow for bus loading greater than the 8086 is capable of handling. Signals ALE, DEN, and DT/$\overline{R}$ are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8086 status ($\overline{S}_2$, $\overline{S}_1$, and $\overline{S}_0$) provide type-of-cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data isn't valid at the leading edge of write. The 8286/8287 transceiver receives the usual T and OE inputs from the 8288's DT/$\overline{R}$ and DEN.

The pointer into the interrupt vector table, which is passed during the second INTA cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8259A Priority Interrupt Controller is positioned on the local bus, a TTL gate is required to disable the 8286/8287 transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll."

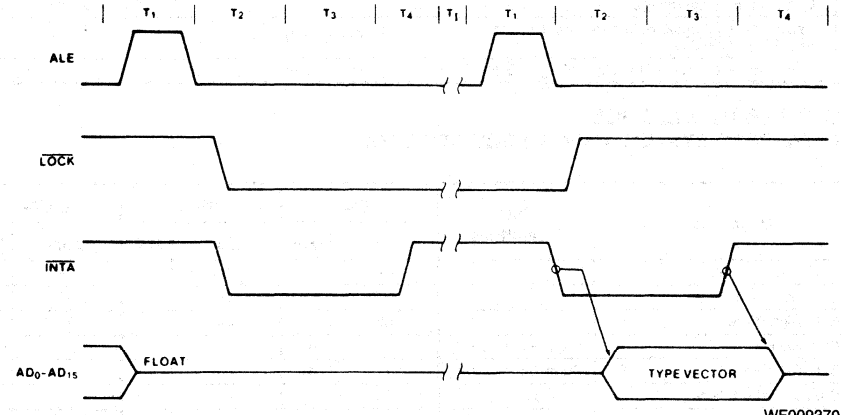


Figure 6. Interrupt Acknowledge Sequence

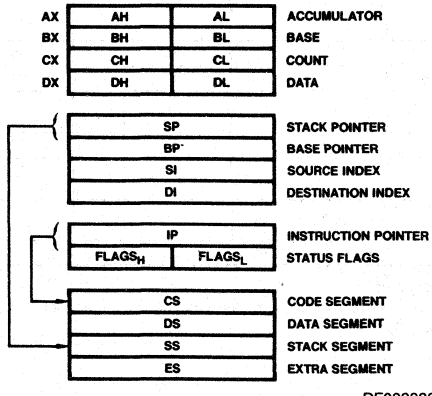


Figure 7. 8086 Register Model

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature Under Bias ..... 0 to 70°C
Voltage on any Pin
with Respect to Ground ..... −1 to +7.0V
Power Dissipitation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8086	0° to 70°C	5V ±10%
8086-2 8086-1	0° to 70°C	5V ±5%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2.5mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current			340	mA
I_{LI}	Input Leakage Current	$0V \leqslant V_{IN} \leqslant V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$0.45V \leqslant V_{OUT} \leqslant V_{CC}$		±10	μA
V_{CL}	Clock Input Low Voltage		−0.5	+0.6	V
V_{CH}	Clock Input High Voltage		3.9	V_{CC} + 1.0	V
C_{IN}	Capacitance of Input Buffer (All input except AD_0-AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		15	pF
C_{IO}	Capacitance of I/O Buffer (AD_0-AD_{15}, $\overline{RQ}/\overline{GT}$)	fc = 1 MHz		15	pF

SWITCHING CHARACTERISTICS
MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284A (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8086		118		68		53		ns
TCHRYX	READY Hold Time into 8086		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 3)		−8		−8		−10		ns
THVCH	HOLD Set-up Time		35		20		20		ns
TINVCH	INTR, NMI, $\overline{TEST}$ Set-up Time (See Note 2)		30		15		15		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

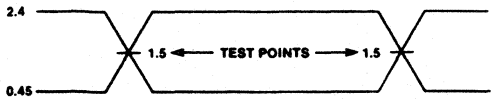
Notes: 1. Signal at 8284A shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T2 state (8ns into T3).

SWITCHING CHARACTERISTICS (Cont.)
TIMING RESPONSES

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	*C_L = 100pF for all 8086 Outputs (in addition to 8086 self-load)	10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TLHLL	ALE Width		TCLCH – 20		TCLCH – 10		TCLCH – 10		ns
TCLLH	ALE Active Delay			80		50		40	ns
TCHLL	ALE Inactive Delay			85		55		45	ns
TLLAX	Address Hold Time to ALE Inactive		TCHCL – 10		TCHCL – 10		TCHCL – 10		ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold Time After WR		TCLCH – 30		TCLCH – 30		TCLCH – 25		ns
TCVCTV	Control Active Delay 1		10	110	10	70	10	50	ns
TCHCTV	Control Active Delay 2		10	110	10	60	10	45	ns
TCVCTX	Control Inactive Delay		10	110	10	70	10	50	ns
TAZRL	Address Float to READ active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL – 45		TCLCL – 40		TCLCL – 35		ns
TCLHAV	HLDA Valid Delay		10	160	10	100	10	60	ns
TRLRH	$\overline{RD}$ Width		2TCLCL – 75		2TCLCL – 50		2TCLCL – 40		ns
TWLWH	$\overline{WR}$ Width		2TCLCL – 60		2TCLCL – 40		2TCLCL – 35		ns
TAVAL	Address Valid to ALE Low		TCLCH – 60		TCLCH – 40		TCLCH – 35		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

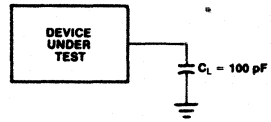
Notes: 1. Signal at 8284A shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T2 state (8ns into T3).

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF009380

SWITCHING TEST LOAD CIRCUIT



TC002190

AC TESTING INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 1.5V FOR BOTH A LOGIC "1" AND "0."

C_L INCLUDES JIG CAPACITANCE

SWITCHING CHARACTERISTICS (Cont.)
MAX MODE SYSTEM (USING 8288 BUS CONTROLLER)
TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284A (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8086		118		68		53		ns
TCHRYX	READY Hold Time into 8086		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 4)		−8		−8		−10		ns
TINVCH	Set-up Time for Recognition (INTR, NMI, $\overline{\text{TEST}}$ (See Note 2)		30		15		15		ns
TGVCH	$\overline{\text{RQ}}/\overline{\text{GT}}$ Set-up Time		30		15		12		ns
TCHGX	$\overline{\text{RQ}}$ Hold Time into 8066		40		30		20		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8ns into T3).

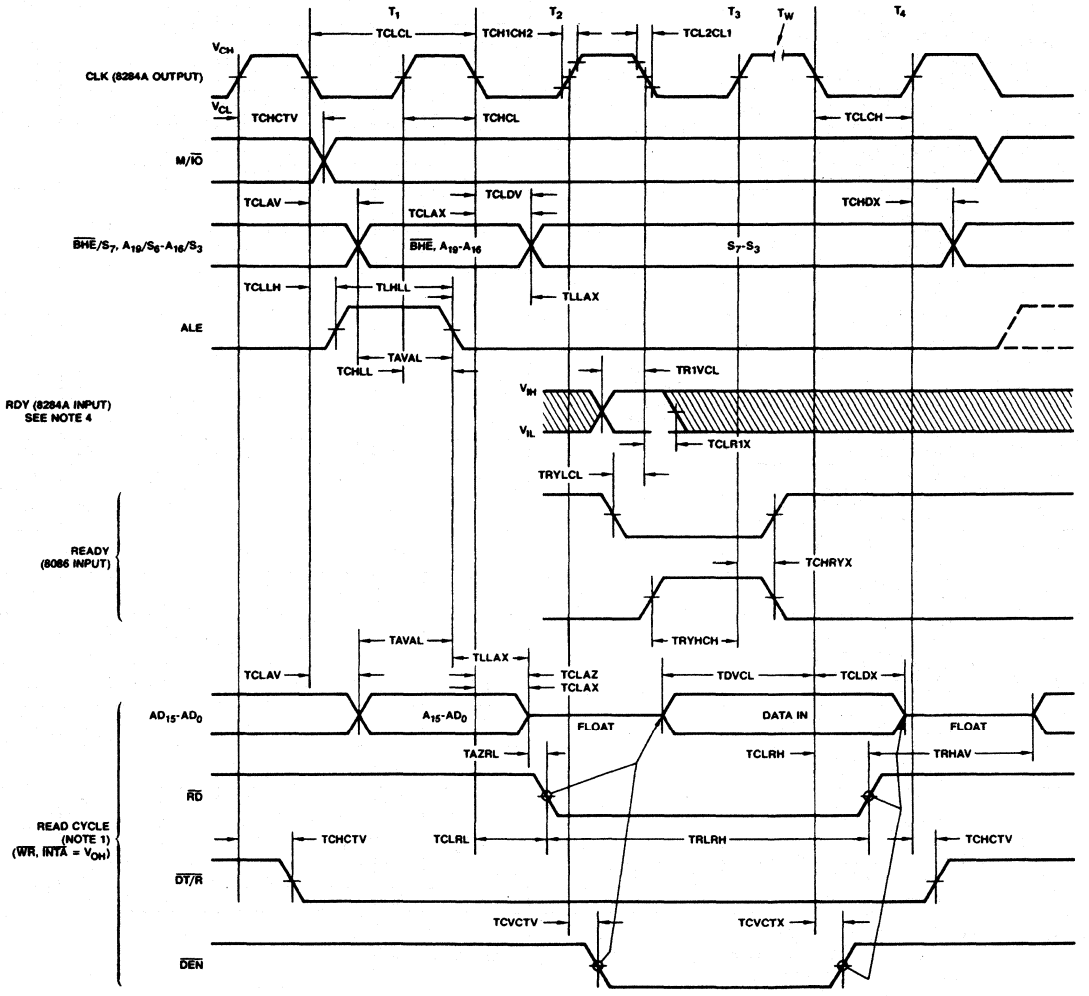
SWITCHING CHARACTERISTICS (Cont.)
TIMING RESPONSES

Parameters	Description	Test Conditions	8086		8086-2		8086-1		Units
			Min	Max	Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 100pF for all 8086 Outputs (In addition to 8086 self-load)	10	35	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	10	35	ns
TRYHSH	READY Active to Status Passive (See Note 3)			110		65		45	ns
TCHSV	Status Active Delay		10	110	10	60	10	45	ns
TCLSH	Status Inactive Delay		10	130	10	70	10	55	ns
TCLAV	Address Valid Delay		10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15		15	ns
TSVMCH	Status Valid to MCE High (See Note 1)			15		15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15		15	ns
TCLMCH	CLK Low to MCE High (See Note 1)			15		15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15		15	ns
TCLMCL	MCE Inactive Delay (See Note 1)			15		15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL – 45		TCLCL – 40		TCLCL – 35		ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30		30	ns
TCLGL	GT Active Delay		0	85	0	50	0	45	ns
TCLGH	GT Inactive Delay		0	85	0	50	0	45	ns
TRLRH	$\overline{RD}$ Width		2TCLCL – 75		2TCLCL – 50		2TCLCL – 40		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T3 and wait states.
4. Applies only to T2 state (8ns into T3).

SWITCHING WAVEFORMS

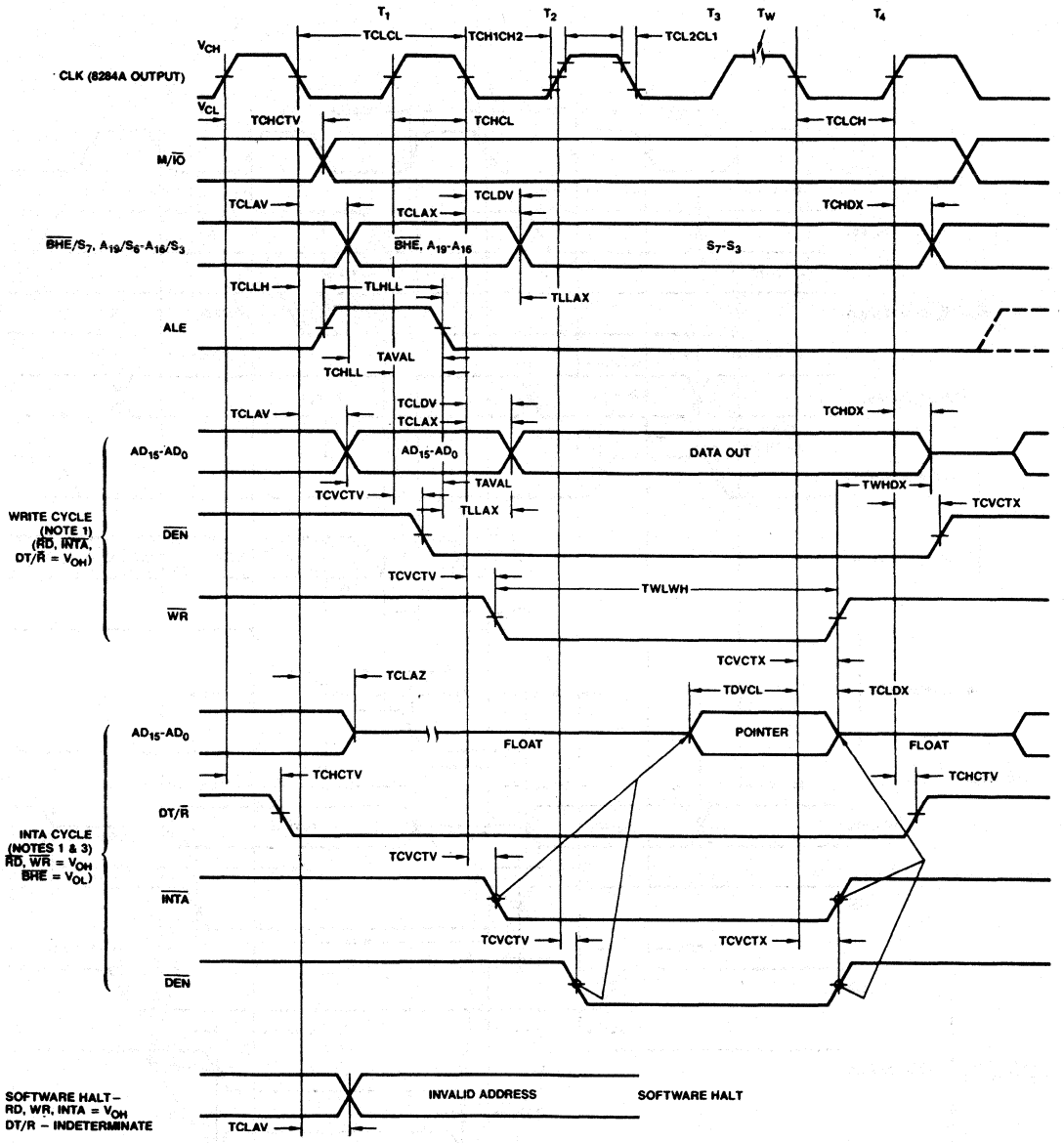
MINIMUM MODE



WF006660

SWITCHING WAVEFORMS (Cont.)

MINIMUM MODE

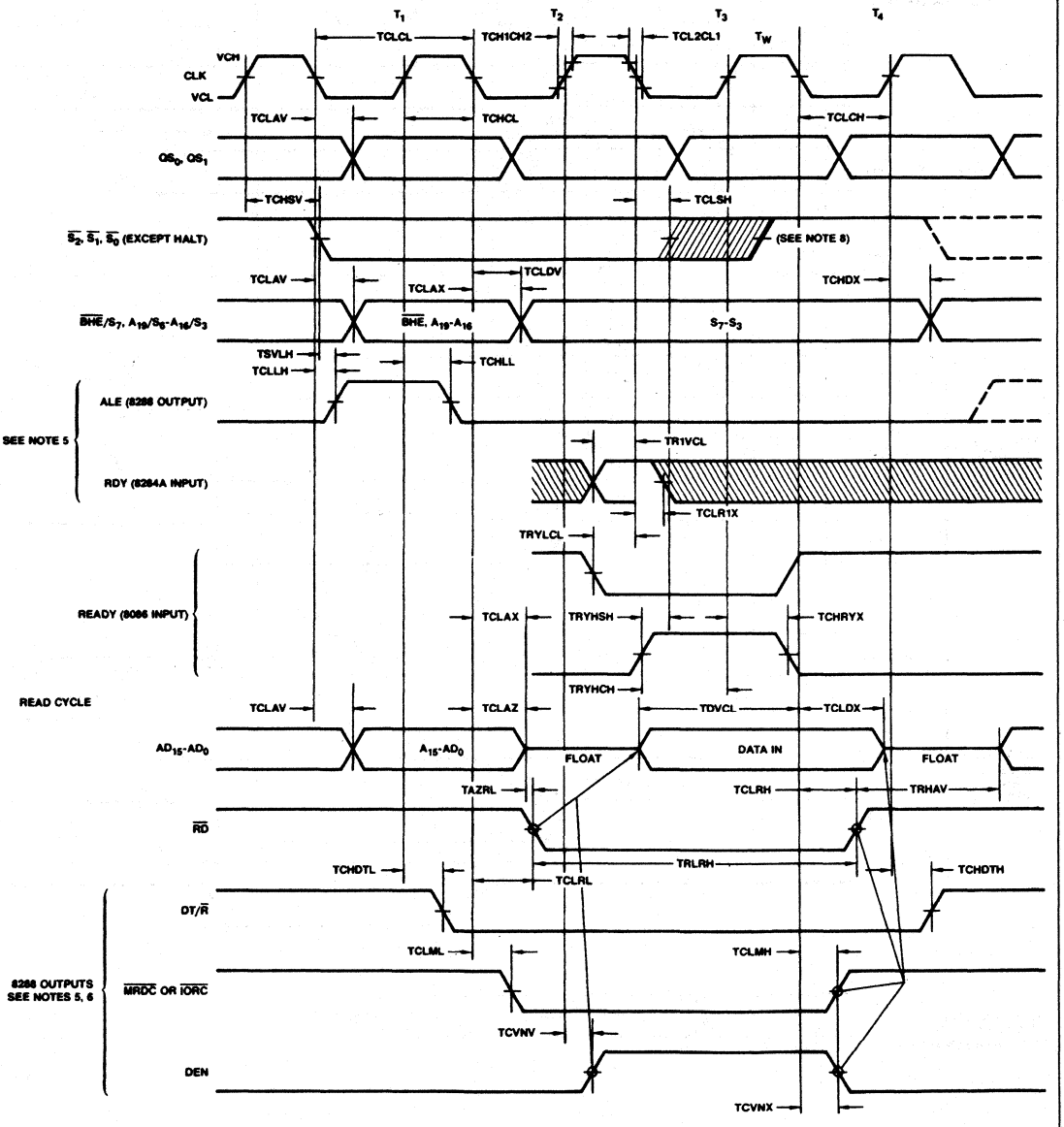


WF006670

Notes:
1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control signals are shown for second INTA cycle.
4. Signals at 8284A are shown for reference only.
5. All timing measurements are made at 1.5V unless otherwise noted.

SWITCHING WAVEFORMS (Cont.)

MAXIMUM MODE



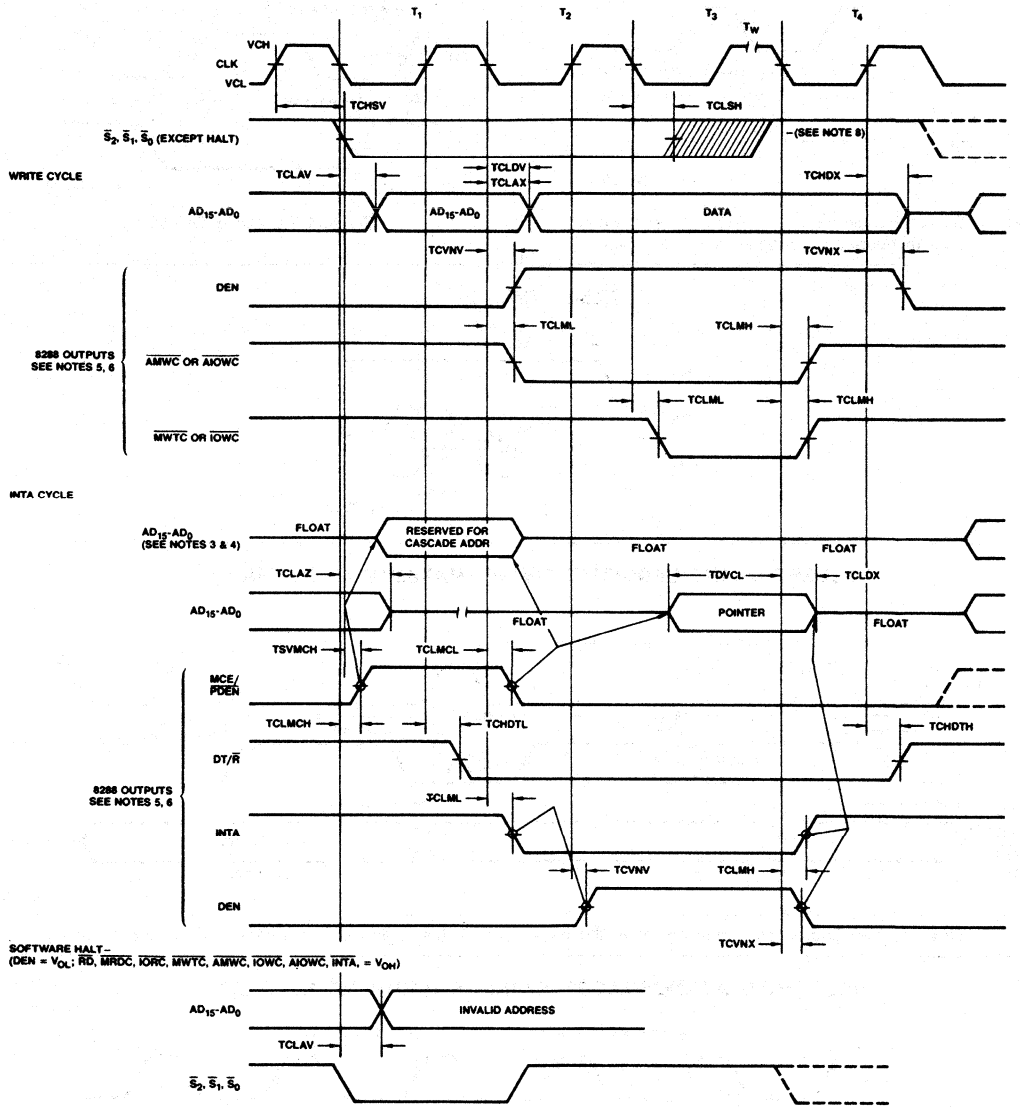
WF006680

01966B

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING WAVEFORMS (Cont.)

MAXIMUM MODE (Cont.)

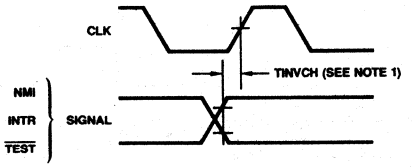


WF006730

Notes:
1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycle.
4. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284A or 8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$ and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5V unless otherwise noted.
8. Status inactive in state just prior to T_4.

SWITCHING WAVEFORMS (Cont.)

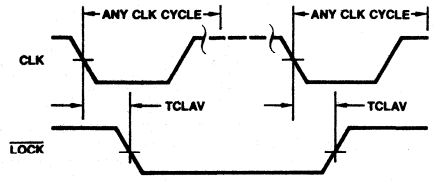
ASYNCHRONOUS SIGNAL RECOGNITION



WF006690

Note: 1. Set-up Requirements for Asynchronous signals only to guarantee recognition at next CLK.

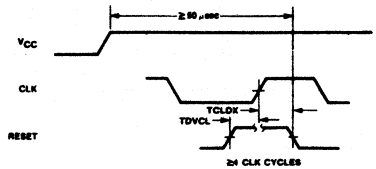
BUS LOCK SIGNAL TIMING



WF006700

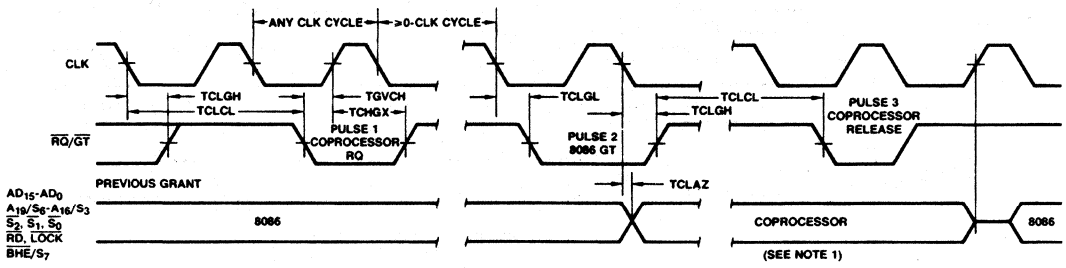
(MAXIMUM MODE ONLY)

RESET TIMING



WF009530

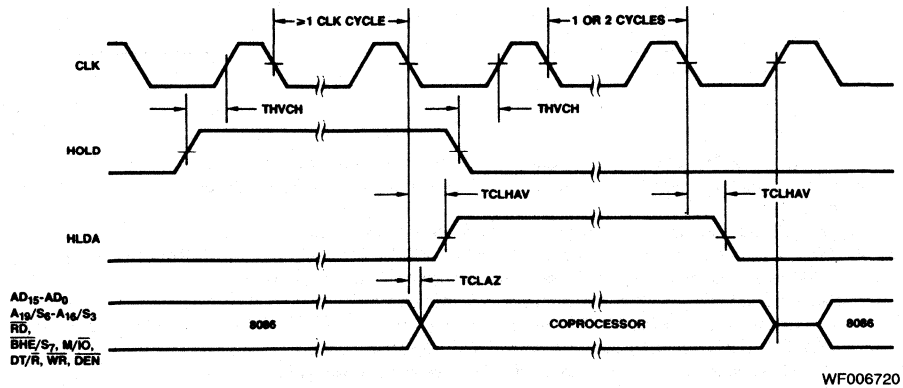
REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



WF006710

Note: 1. The Coprocessor may not drive the buses outside the region shown without risking contention.

HOLD/HOLD ACKNOWLEDGE TIMING (MINIMUM MODE ONLY)



WF006720

8086/8088
INSTRUCTION SET SUMMARY

DATA TRANSFER

MOV = Move	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to /from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to register	1 0 1 1 w reg	data	data if w = 1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Ouput to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Transtate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LANF = Load AH with flags	1 0 0 1 1 1 1 1			
SANF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			

INSTRUCTION SET SUMMARY (Cont.)

ARITHMETIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
ADD = Add				
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	
ADC = Add with carry:				
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	
INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			
SUB = Subtract:				
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	
SBB = Subtract with borrow:				
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	
DEC = Decrement:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
CMP = Compare:				
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m		
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
AAS ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD ASCH adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW Convert byte to word	1 0 0 1 1 0 0 0			
CWD Convert word to double word	1 0 0 1 1 0 0 1			

INSTRUCTION SET SUMMARY (Cont.)

LOGIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
NOT Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR Shift logical right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
SAR Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR Rotate through carry right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
TEST = And function to flags, no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
OR = Or:				
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
XOR = Exclusive or:				
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	

STRING MANIPULATION:

REP = Repeat	1 1 1 1 0 0 1 z
MOVS = Move byte/word	1 0 1 0 0 1 0 w
CMPS = Compare byte/word	1 0 1 0 0 1 1 w
SCAS = Scan byte/word	1 0 1 0 1 1 1 w
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w

INSTRUCTION SET SUMMARY (Cont.)

CONTROL TRANSFER

CALL = Call	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	
indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		

JMP = Unconditional jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		

RET = Return from CALL:			
Within segment	1 1 0 0 0 0 1 1		
Within seg adding immed to SP	1 1 0 0 0 0 1 0	data-low	data-high
Intersegment	1 1 0 0 1 0 1 1		
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp	
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp	
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp	
JB/JNAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp	
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp	
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp	
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp	
JS = Jump on sign	0 1 1 1 1 0 0 0	disp	
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp	
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp	
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp	
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp	
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp	
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	disp	
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp	
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp	
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp	
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp	
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp	
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp	

INSTRUCTION SET SUMMARY (Cont.)

CONTROL TRANSFER (Cont.)

INT = Interrupt	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Type specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0			
IRET = Interrupt return	1 1 0 0 1 1 1 1			

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0	
CMC = Complement carry	1 1 1 1 0 1 0 1	
STC = Set carry	1 1 1 1 1 0 0 1	
CLD = Clear direction	1 1 1 1 1 1 0 0	
STD = Set direction	1 1 1 1 1 1 0 1	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	
STI = Set interrupt	1 1 1 1 1 0 1 1	
HLT = Halt	1 1 1 1 0 1 0 0	
WAIT = Wait	1 0 0 1 1 0 1 1	
ESC = Processor Extension Escape	1 0 0 1 1 x x x	mod x x x r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive.
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with ZF Flag.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register files as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

8087

Numeric Data Coprocessor
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- High performance arithmetic and transcendental functions in hardware
- Supports 8-, 16-, 32-, 64-bit integer
- Performs 32-, 64-, 80-bit floating point calculations conforming to IEEE standard
- Standard 8086 instruction set and addressing modes
- Built-in exception handling functions
- Multibus* system compatible

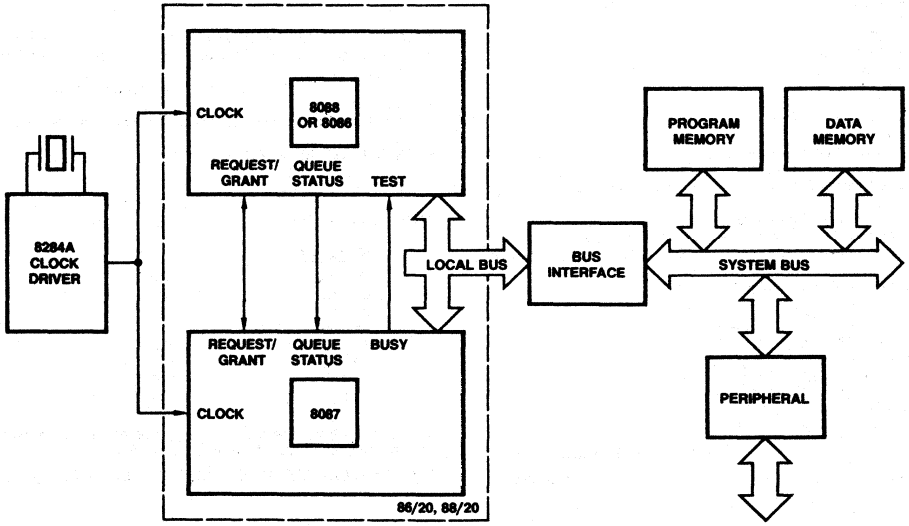
GENERAL DESCRIPTION

The 8087 is designed to do high performance numeric processing in hardware. It operates as the coprocessor to an 8086 or 8088 CPU and can improve numeric throughput by a factor of 100 over the stand-alone CPU. It is programmed with the same instruction set as the 8086/88.

The 8087 does trigonometric, logarithmic, and exponential functions, which are essential in many scientific and military applications. The 8087 can also process BCD numbers up to 18 digits with no round-off error.

The 8087 is built in N-channel depletion load technology in a 40-pin package.

BLOCK DIAGRAM



BD003720

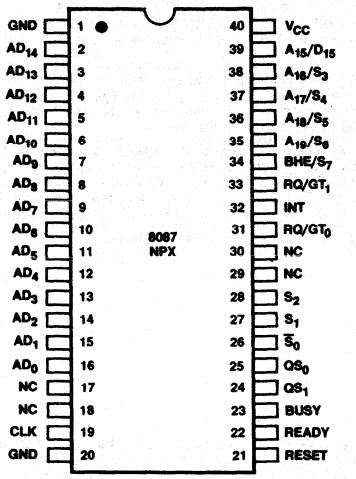
*Multibus is a registered trademark of Intel Corporation.

02037B

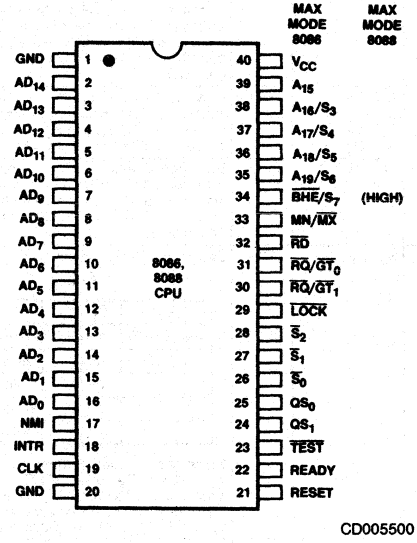
Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40



CD005490

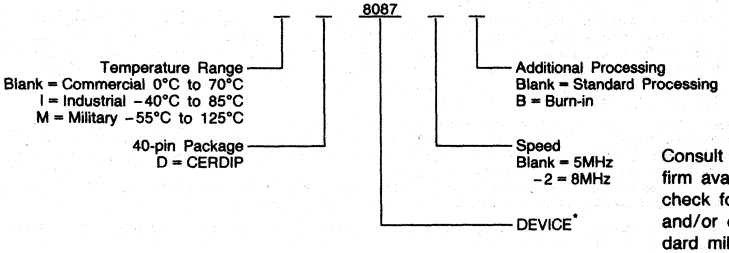


CD005500

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

*A "C" in the middle of the device type denotes CMOS version of the product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
39, 2-16	AD_{15}-AD_0	I/O	Address Data. These lines constitute the time multiplexed memory address (T_1) and data (T_2, T_3, T_W, T_4) bus. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7-D_0. It is LOW during T_1 when a byte is to be transferred on the lower portion of the bus in memory operations. Eight-bit oriented devices tied to the lower half of the bus would normally use A_0 to condition chip select functions. These lines are active HIGH. They are input/output lines for 8087 driven bus cycles and are inputs which the 8087 monitors when the 8086/8088 is in control of the bus.
35, 36, 37, 38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	I/O	Address Memory. During T_1 these are the four most significant address lines for memory operations. During memory operations, status information is available on these lines during T_2, T_3, T_W, and T_4. For 8087 controlled bus cycles, S_6, S_4, and S_3 are reserved and currently one (HIGH), while S_5 is always LOW. These lines are inputs which the 8087 monitors when the 8086/8088 is in control of the bus.
34	$\overline{BHE}/S_7$	I/O	Bus High Enable. During T_1 the bus high enable signal ($\overline{BHD}$) should be used to enable data onto the most significant half of the data bus, pins D_{15}-D_8. Eight-bit oriented devices tied to the upper half of the bus would normally use $\overline{BHE}$ to condition chip select functions. $\overline{BHE}$ is LOW during T_1 for read and write cycles when a byte is to be transferred on the high portion of the bus. The S_7 status information is available during T_2, T_3, T_W, and T_4. The signal is active LOW. S_7 is an input which the 8087 monitors during 8086/8088 controlled bus cycles.
26, 27, 28	S_0, S_1, S_2	I/O	Status. For 8087 driven bus cycles, these status lines are encoded as follows: **Table 1.** $\overline{S_2}$ / $\overline{S_1}$ / $\overline{S_0}$ 0 (LOW) X X Unused 1 (HIGH) 0 0 Unused 1 0 1 Read Memory 1 1 0 Write Memory 1 1 1 Passive Status is driven active during T_4, remains valid during T_1 and T_2, and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 Bus Controller to generate all memory access control signals. Any change in $\overline{S_2}$, $\overline{S_1}$, or $\overline{S_0}$ during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals are monitored by the 8087 when the 8086/8088 is in control of the bus.
31	$\overline{RQ}/\overline{GT}_0$	I/O	Request/Grant. This request/grant pin is used by the NDP to gain control of the local bus from the CPU for operand transfers or on behalf of another bus master. It must be connected to one of the two processor request/grant pins. The request/grant sequence on this pin is as follows: 1. A pulse one clock wide is passed to the CPU to indicate a local bus request by either the 8087 or the master connected to the 8087 $\overline{RQ}/\overline{GT}_1$ pin. 2. The NDP waits for the grant pulse and when it is received will either initiate bus transfer activity in the clock cycle following the grant or pass the grant out on the $\overline{RQ}/\overline{GT}_1$ pin in this clock if the initial request was for another bus master. 3. The 8087 will generate a release pulse to the CPU one clock cycle after the completion of the last NDP bus cycle or on receipt of the release pulse from the bus master on $\overline{RQ}/\overline{GT}_1$.
30	$\overline{RQ}/\overline{GT}_1$	I/O	Request/Grant. This request/grant pin is used by another local bus master to force the NDP to release the local bus at the end of the processor's current bus cycle. If the NDP is not in control of the bus when the request is made, the request/grant sequence is passed through the NDP on the $\overline{RQ}/\overline{GT}_0$ pin one cycle later. Subsequent grant and release pulses are also passed through the NDP with a two and one clock delay, respectively, for resynchronization. $\overline{RQ}/\overline{GT}_1$ has an internal pull-up resistor and may be left unconnected. If the NDP has control of the bus, the request/grant sequence is as follows: 1. A pulse 1 CLK wide from another local bus master indicates a local bus request to the 8087 (pulse 1). 2. During the NDP's next T_4 or T_1, a pulse 1 CLK wide from the 8087 to the requesting master (pulse 2) indicates that the 8087 has allowed the local bus to float and that it will enter the "RQ/GT acknowledge" state at the next CLK. The NDP's control unit is disconnected logically from the local bus during "RQ/GT acknowledge." 3. A pulse 1 CLK wide from the requesting master indicates to the 8087 (pulse 3) that the "RQ/GT" request is about to end and that the 8087 can reclaim the local bus at the next CLK. Each master-master exchange of the local bus is a sequence of 3 pulses. There must be one dead CLK cycle after each bus exchange. Pulses are active LOW.
24, 25	QS_1, QS_0	I	QS_1, QS_0. QS_1 and QS_0 provide the 8087 with status to allow tracking of the CPU instruction queue. QS_1 / QS_0 0 (LOW) 0 No Operation 0 1 First Byte of Op Code from Queue 1 (HIGH) 0 Empty the Queue 1 1 Subsequent Byte from Queue
32	INT	O	Interrupt. This line is used to indicate that an unmasked exception has occurred during numeric instruction execution when 8087 interrupts are enabled. This signal is typically routed to an 8259A. INT is active HIGH.
23	BUSY	O	Busy. This signal indicates that the 8087 NEU is executing a numeric instruction. It is connected to the CPU's $\overline{TEST}$ pin to provide CPU-NDP synchronization. In the case of an unmasked exception, BUSY remains active until the exception is cleared. BUSY is active HIGH.
22	READY	I	Ready. READY is the acknowledgment from the addressed memory device that it will complete the data transfer. The RDY signal from memory is synchronized by the 8284A Clock Generator to form READY. This signal is active HIGH.
21	RESET	I	Reset. RESET causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. RESET is internally synchronized.
19	CLK	I	Clock. The clock provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		Power. V_{CC} is the +5V power supply pin.
1, 20	GND		Ground. GND are the ground pins.

02037B

Refer to page 7-1 for Essential Information on Military Devices

DETAILED DESCRIPTION

The 8087 is a numeric processor extension that provides arithmetic and logical instruction support for a variety of numeric data types. It also executes numerous built-in transcendental functions (e.g., tangent and log functions). The 8087 executes instructions as a coprocessor to a maximum mode 8086 or 8088. Figure 3 presents the registers of the 8087 plus CPU combination. Table 2 shows the range of data types supported by the NDP. The 8087 is treated as an extension to the CPU, providing register, data types, control, and instruction capabilities at the hardware level. At the programmer's level, the CPU and NDP is viewed as a single unified processor.

System Configuration

As a coprocessor to an 8086 or 8088, the 8087 is wired in parallel with the CPU as shown in Figure 4. The CPU's status ($\overline{S}_0 - \overline{S}_2$) and queue status lines ($QS_0 - QS_1$) enable the 8087 to monitor and decode instructions in synchronization with the CPU and without any CPU overhead. Once started the 8087 can process in parallel with, and independent of, the host CPU. The NPX can interrupt the CPU when it detects an error or exception. The 8087's interrupt request line is typically routed to the CPU through an 8259A Programmable Interrupt Controller.

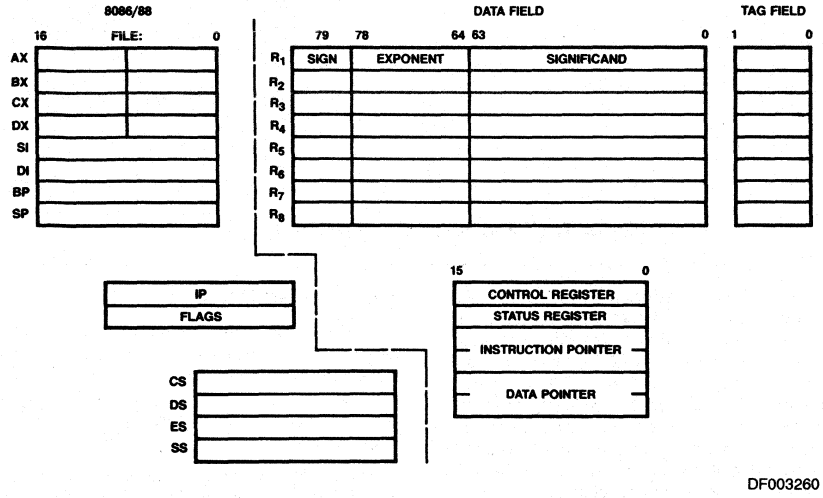


Figure 3. 8087 Register Architecture

Table 2. 8087 Data Types

Data Formats	Range	Precision	Most Significant Byte
Byte Integer	10^2	8 Bits	I_7 I_0 Two's Complement
Word Integer	10^4	16 Bits	I_{15} I_0 Two's Complement
Short Integer	10^9	32 Bits	I_{31} I_0 Two's Complement
Long Integer	10^{18}	64 Bits	I_{63} I_0 Two's Complement
Packed BCD	10^{18}	18 Digits	S — D_{17} D_{16} D_1 D_0
Short Real	$10^{\pm 38}$	24 Bits	S E_7 E_0 F_1 F_{23} F_0 Implicit
Long Real	$10^{\pm 308}$	53 Bits	S E_{10} E_0 F_1 F_{52} F_0 Implicit
Temporary Real	$10^{\pm 4932}$	64 Bits	S E_{14} E_0 F_0 F_{63}

Integer: 1
Packed BCD: $(-1)^S$ $(D_{17} \ldots D_0)$

Real: $(-1)^S$ $(2^{E\text{-BIAS}})(F_0 \cdot F_1 \ldots)$
Bias = 127 for Short Real
1023 for Long Real
16383 for Temp Real

02037B

Refer to page 7-1 for Essential Information on Military Devices

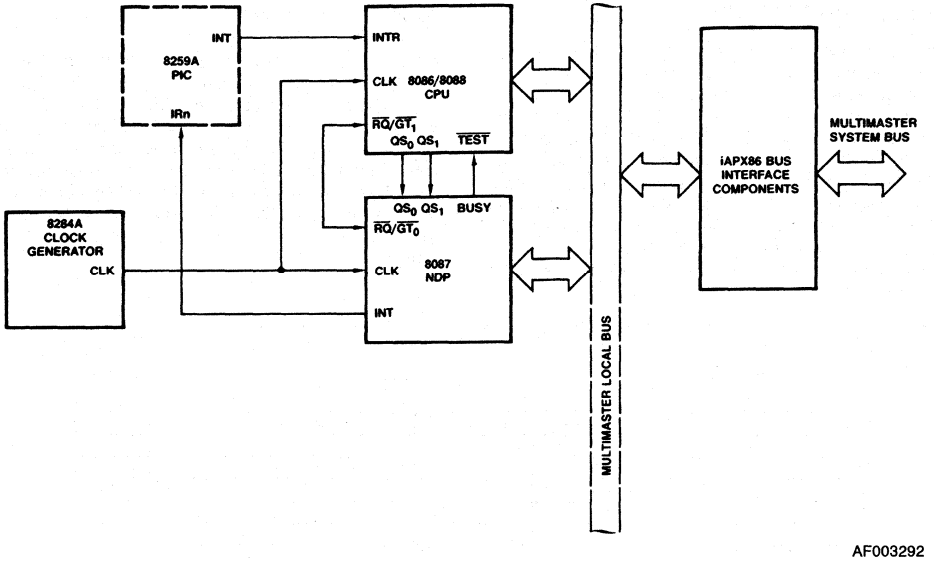


Figure 4. NDP System Configuration

The 8087 uses one of the request/grant lines (typically $\overline{RQ}/\overline{GT_1}$) to obtain control of the local bus for data transfers. The other request/grant line is available for general system use (for instance by an I/O processor in LOCAL mode). A bus master can also be connected to the 8087's RQ/GT_1 line. In this configuration the 8087 will pass the request/grant handshake signals between the CPU and the attached master when the 8087 is not in control of the bus and will relinquish the bus to the master directly when the 8087 is in control. In this way two additional masters can be configured; one will share the 8086 bus with the 8087 on a first come first served basis, and the second will be guaranteed to be higher in priority than the 8087.

As Figure 4 shows, all processors utilize the same clock generator and system bus interface components.

Bus Operation

The 8087 bus structure, operation and timing are identical to all other processors in the 8086 family. The address is time multiplexed with the data on the first 16/8 lines of the address/data bus. A_{16} through A_{19} are time multiplexed with four status lines $S_3 - S_6$; S_3, S_4 and S_6 are always one (HIGH) for 8087 driven bus cycles while S_5 is always zero (LOW). When the 8087 is monitoring CPU bus cycles (passive mode), S_6 is also monitored by the 8087 to differentiate 8086/8088 activity from that of a local I/O processor or any other local bus master. (The 8086/8088 must be the only processor on the local bus to drive S_6 low.) S_7 is multiplexed with and has the same value as $\overline{BHE}$ for all 8087 bus cycles.

The first three status lines, $\overline{S}_0 - \overline{S}_2$, are used with an 8288 bus controller to determine the type of bus cycle being run:

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	
0	X	X	Unused
1	0	0	Unused
1	0	1	Memory Data Read
1	1	0	Memory Data Write
1	1	1	Passive (no bus cycle)

Programming Interface

The NDP includes the standard 8086/88 instruction set for general data manipulation and program control. It also includes 68 numeric instructions for extended precision integer, floating point, trigonometric, logarithmic, and exponential functions. Sample execution times for several NDP functions are shown in Figure 4.

Any instruction executed by the NDP is the combined result of the CPU and NPX activity. The CPU and NPX have specialized functions and registers providing fast concurrent operation. The CPU controls overall program execution while the NPX uses the coprocessor interface to recognize and perform numeric operations.

Table 2 lists the eight data types the 8087 supports and presents the format for each type. Internally, the NPX holds all numbers in the temporary real format. Load and store instructions automatically convert operands represented in memory as 16-, 32-, or 64-bit integers, 32- or 64-bit floating point numbers, or 18-digit packed BCD numbers into temporary real format and vice versa. The NDP also provides the capability to control round off, underflow, and overflow errors in each calculation.

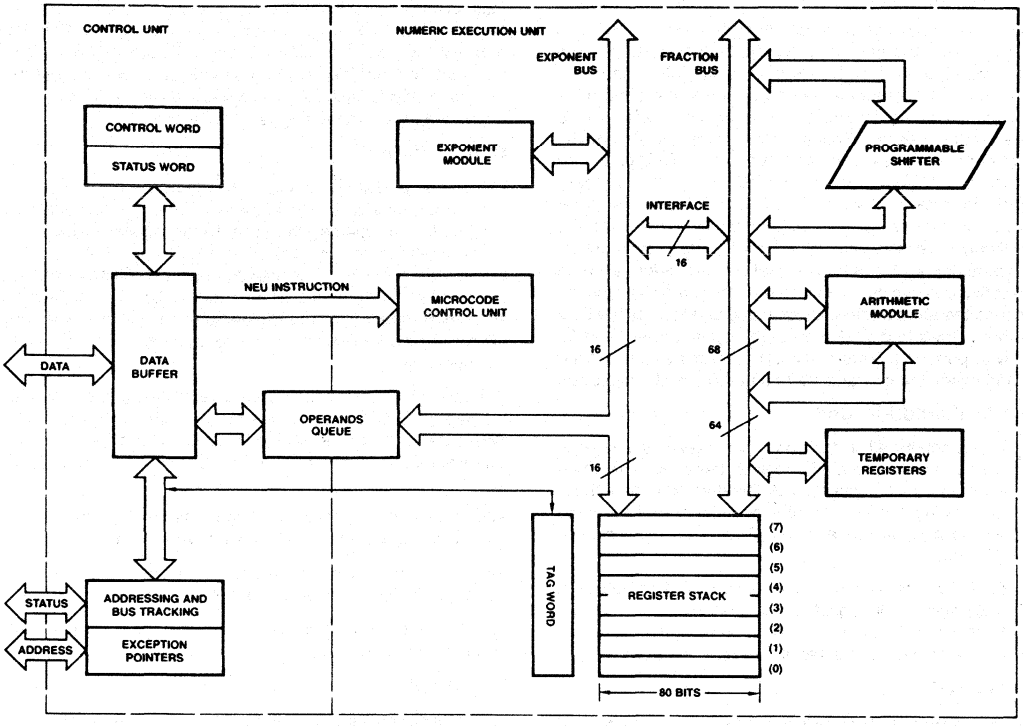
Computations in the NPX use the processor's register stack. These eight 80-bit registers provide the equivalent capacity of 20 32-bit registers. The NPX register set can be accessed as a stack, with instructions operating on the top one or two stack elements, or as a fixed register set, with instructions operating on explicitly designated registers.

All 8087 instructions appear as ESCAPE instructions to the host CPU. Assembly language programs are written in ASM-86, the 8086/88 assembly language. Table 3 gives the execution times of some typical numeric instructions.

Numeric Processor Extension Architecture

As shown in Figure 5, the 8087 is internally divided into two processing elements, the control unit (CU) and the numeric

execution unit (NEU). The NEU executes all numeric instructions, while the CU receives and decodes instructions, reads and writes memory operands, and executes NPX control instructions. The two elements are able to operate independently of one another, allowing the CU to maintain synchronization with the CPU while the NEU is busy processing a numeric instruction.



BD003730

Figure 5. 8087 Black Diagram

TABLE 3. EXECUTION TIME FOR SELECTED 8087 NUMERIC INSTRUCTIONS AND CORRESPONDING 8086 EMULATION

Floating Point Instruction	Approximate Execution Time (μs)	
	8087 (5MHz Clock)	8086 Emulation
Add/Subtract Magnitude	14/18	1,600
Multiply (single precision)	19	1,600
Multiply (extended precision)	27	2,100
Divide	39	3,200
Compare	9	1,300
Load (double percision)	10	1,700
Store (double percision)	21	1,200
Square Root	36	19,600
Tangent	90	13,000
Exponentiation	100	17,000

Control Unit

The CU keeps the 8087 operating in synchronization with its host CPU. 8087 instructions are intermixed with CPU instructions in a single instruction stream. The CPU fetches all instructions from memory; by monitoring the status signals ($\overline{S}_0 - \overline{S}_2$, S_6) emitted by the CPU, the NPX control unit determines when an 8086 instruction is being fetched. The CU monitors the Data bus in parallel with the CPU to obtain instructions that pertain to the 8087.

The CU maintains an instruction queue that is identical to the queue in the host CPU. The CU automatically determines if the CPU is an 8086 or an 8088 immediately after reset (by monitoring the $\overline{BHE}/S_7$ line) and matches its queue length accordingly. By monitoring the CPU's queue status lines (QS_0, QS_1), the CU obtains and decodes instructions from the queue in synchronization with the CPU.

A numeric instruction appears as an ESCAPE instruction to the 8086 or 8088 CPU. Both the CPU and NPX decode and execute the ESCAPE instruction together. The 8087 only recognizes the numeric instructions shown in Table 5. The start of a numeric operation is accomplished when the CPU executes the ESCAPE instruction. The instruction may or may not identify a memory operand.

The CPU does, however, distinguish between ESC instructions that reference memory and those that do not. If the instruction refers to a memory operand, the CPU calculates

the operand's address using any one of its available addressing modes, and then performs a "dummy read" of the word at that location. (Any location within the 1M byte address space is allowed.) This is a normal read cycle except that the CPU ignores the data it receives. If the ESC instruction does not contain a memory reference (e.g., an 8087 stack operation), the CPU simply proceeds to the next instruction.

An 8087 instruction can have one of three memory reference options: (1) not reference memory; (2) load an operand from memory into the 8087; or (3) store an operand from the 8087 into memory. If no memory reference is required, the 8087 simply executes its instruction. If a memory reference is required, the CU uses a "dummy read" cycle initiated by the CPU to capture and save the address that the CPU places on the bus. If the instruction is a load, the CU additionally captures the data word when it becomes available on the local data bus. If data required is longer than one word, the CU immediately obtains the bus from the CPU using the request/grant protocol and reads the rest of the information in consecutive bus cycles. In a store operation, the CU captures and saves the store address as in a load and ignores the data word that follows in the "dummy read" cycle. When the 8087 is ready to perform the store, the CU obtains the bus from the CPU and writes the operand starting at the specified address.

Numeric Execution Unit

The NEU executes all instructions that involve the register stack; these include arithmetic, logical, transcendental, constant and data transfer instructions. The data path in the NEU is 84 bits wide (68 fraction bits, 15 exponent bits and a sign bit) which allows internal operand transfers to be performed at very high speeds.

When the NEU begins executing an instruction, it activates the 8087 BUSY signal. This signal can be used in conjunction with the CPU WAIT instruction to resynchronize both processors when the NEU has completed its current instruction.

Register Set

The 8087 register set is shown in Figure 3. Each of the eight data registers in the 8087's register stack is 80 bits wide and is divided into "fields" corresponding to the NDP's temporary real data type.

At a given point in time, the TOP field in the control word identifies the current top-of-stack register. A "push" operation decrements TOP by 1 and loads a value into the new top register. A "pop" operation stores the value from the current top register and then increments TOP by 1. The 8087 register stack grows "down" toward lower-addressed registers.

Instructions may address the data registers either implicitly or explicitly. Many instructions operate on the register at the top of the stack. These instructions implicitly address the register pointed to by the TOP. Other instructions allow the programmer to explicitly specify the register which is to be used. Explicit register addressing is "top-relative."

Status Word

The status word shown in Figure 6 reflects the overall state of the 8087; it may be stored in memory and then inspected by CPU code. The status word is a 16-bit register divided into fields as shown in Figure 6. The busy bit (bit 15) indicates whether the NEU is either executing an instruction or has an interrupt request pending (B = 1), or is idle (B = 0). Several instructions which store and manipulate the status word are executed exclusively by the CU, and these do not set the busy bit themselves.

The four numeric condition code bits ($C_0 - C_3$) are similar to the flags in a CPU: various instructions update these bits to reflect the outcome of NDP operations. The effect of these instructions on the condition code bits is summarized in Table 4.

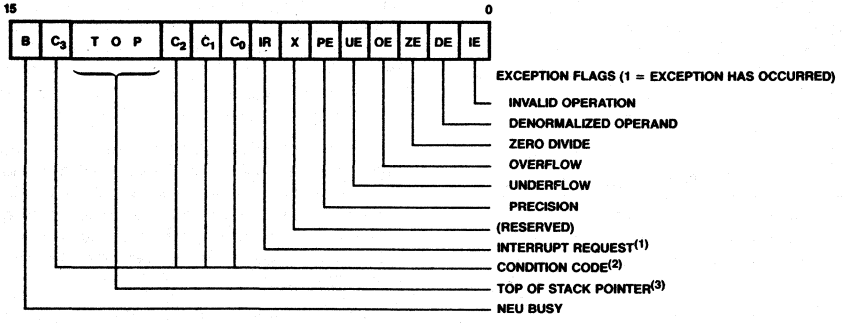
Bits 14-12 of the status word point to the 8087 register that is the current top-of-stack (TOP) as described above.

Bit 7 is the interrupt request bit. This bit is set if any unmasked exception bit is set and cleared otherwise.

Bits 5 – 0 are set to indicate that the NEU has detected an exception while executing an instruction.

Tag Word

The tag word marks the contents of each register as shown in Figure 7. The principal function of the tag word is to optimize the NDP's performance. The tag word can be used, however, to interpret the contents of 8087 registers.



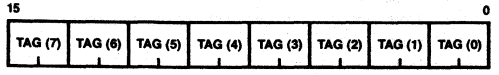
(1) R is set if any unmasked exception bit is set, cleared otherwise.
(2) See Table 3 for condition code interpretation.
(3) Top Values:
000 = Register 0 is Top of Stack.
001 = Register 1 is Top of Stack.
•
•
•
111 = Register 7 is Top of Stack.

Figure 6. 8087 Status Word

TABLE 4. CONDITION CODE INTERPRETATION

Instruction	C_3	C_2	C_1	C_0	Interpretation
Compare, Test	0	X	X	0	A > B
	0	X	X	1	A < B
	1	X	X	0	A = B
	1	X	X	1	A ? B (not comparable)
Remainder	U	0	U	U	Complete reduction
	U	1	U	U	Incomplete reduction
Examine	0	0	0	0	Valid, positive, unnormalized
	0	0	0	1	Invalid, positive, exponent = 0
	0	0	1	0	Valid, negative, unnormalized
	0	0	1	1	Invalid, negative, exponent = 0
	0	1	0	0	Valid, positive, normalized
	0	1	0	1	Infinity, positive
	0	1	1	0	Valid, negative, normalized
	0	1	1	1	Infinity, negative
	1	0	0	0	Zero, positive
	1	0	0	1	Empty
	1	0	1	0	Zero, negative
	1	0	1	1	Empty
	1	1	0	0	Invalid, positive, exponent = 0
	1	1	0	1	Empty
	1	1	1	0	Invalid, negative, exponent = 0
	1	1	1	1	Empty

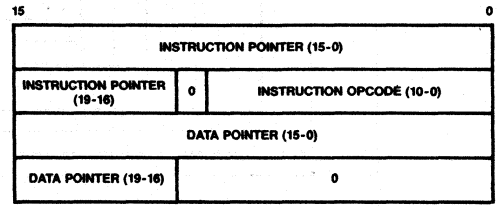
X = value is not affected by instruction.
U = value is undefined following instruction.



DF003280

TAG VALUES:
00 = VALID
01 = ZERO
10 = SPECIAL
11 = EMPTY

Figure 7. 8087 Tag Word



DF003290

Figure 8. 8087 Instruction and Data Pointers

Instruction and Data Pointers

The instruction and data pointers (see Figure 8) are provided for user-written error handlers. Whenever the 8087 executes an NEU instruction, the CU saves the instruction address, the operand address (if present), and the instruction opcode. 8087 instructions can store this data into memory.

Control Word

The 8087 provides several processing options which are selected by loading a word from memory into the control word. Figure 9 shows the format and encoding of the fields in the control word.

The low order byte of this control word configures 8087 interrupts and exception masking. Bits 5 – 0 of the control word contain individual masks for each of the six exceptions that the 8087 recognizes, and bit 7 contains a general mask bit for all 8087 interrupts. The high order byte of the control word configures the 8087 operating mode including precision, rounding, and infinity controls. The precision control bits (bits 9 – 8) can be used to set the 8087 internal operating precision at less than the default of temporary real precision. This can be useful in providing compatibility with earlier generation arithmetic processors of smaller precision than the 8087. The rounding control bits (bits 11 – 10) provide for directed rounding and true chop as well as the unbiased round to the nearest mode specified in the proposed IEEE standard. Control over closure of the number space at infinity is also provided (either affine closure, $\pm\infty$, or projective closure, ∞, is treated as unsigned, may be specified).

Exception Handling

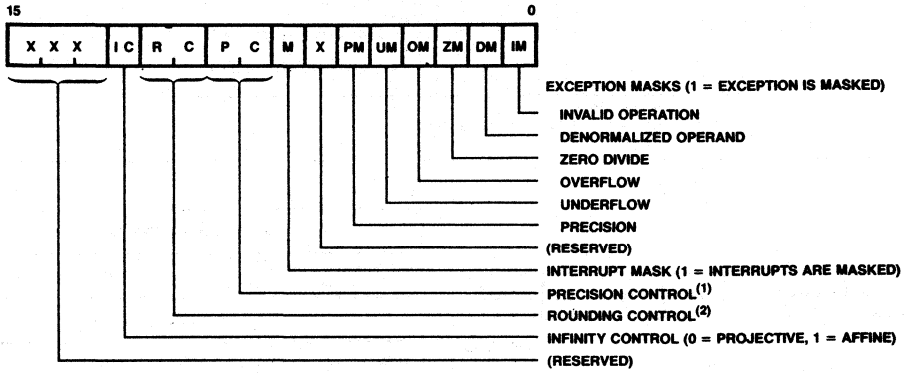
The 8087 detects six different conditions that can occur during instruction execution. Any or all exceptions will cause an interrupt if unmasked and interrupts are enabled.

If interrupts are disabled, the 8087 will simply continue execution regardless of whether the host clears the exception. If a specific exception class is masked and that exception occurs, however, the 8087 will post the exception in the status register and perform an on-chip default exception handling procedure, thereby allowing processing to continue. The exceptions that the 8087 detects are the following:

1. INVALID OPERATION: Stock overflow, stack underflow, indeterminate form (0/0, $\infty - \infty$, etc.) or the use of a Non-Number (NAN) as an operand. An exponent value is reserved and any bit pattern with this value in the exponent field is termed a Non-Number and causes this exception. If this exception is masked, the 8087's default response is to generate a specific NAN called

INDEFINITE or to propagate already existing NANs as the calculation result.

2. OVERFLOW: The result is too large in magnitude to fit the specified format. The 8087 will generate an encoding for infinity if this exception is masked.
3. ZERO DIVISOR: The divisor is zero while the dividend is a non-infinite, non-zero number. Again, the 8087 will generate an encoding for infinity if this exception is masked.
4. UNDERFLOW: The result is non-zero but too small in magnitude to fit in the specified format. If this exception is masked, the 8087 will denormalize (shift right) the fraction until the exponent is in range. This process is called gradual underflow.
5. DENORMALIZED OPERAND: At least one of the operands or the result is denormalized; it has the smallest exponent but a non-zero significand. Normal processing continues if this exception is masked off.
6. INEXACT RESULT: If the true result is not exactly representable in the specified format, the result is rounded according to the rounding mode, and this flag is set. If this exception is masked, processing will simply continue.



(1)Precision Control
00 = 24 bits
01 = Reserved
10 = 53 bits
11 = 64 bits

(2)Rounding Control
00 = Round to Nearest or Even
01 = Round Down (toward $-\infty$)
10 = Round Up (toward $+\infty$)
11 = Chop (truncate toward zero)

Figure 9. 8087 Control Word

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to Ground ..... −1.0 to +7.0V
Power Dissipation 3.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

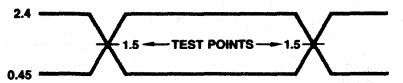
Part Number	T_A	V_{CC}
8087	T_A = 0°C to 70°C	5V ±5%
8087-2		

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over Operating Ranges)

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2.0μA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current	T_A = 25°C		475	mA
I_{LI}	Input Leakage Current	$0V \leq V_{IN} \leq V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$0.45V \leq V_{OUT} \leq V_{CC}$		±10	μA
V_{CL}	Clock Input Low Voltage		−0.5	+0.6	V
V_{CH}	Clock Input High Voltage		3.9	V_{CC} + 1.0	V
V_{IN}	Capacitance of Inputs	fc = 1MHz		10	pF
C_{IO}	Capacitance of I/O Buffer (AD_{0-15}, A_{16}-A_{19}, BHE, S_2-S_0, RQ/GT) and CLK	fc = 1MHz		15	pF
C_{OUT}	Capacitance of Outputs BUSY, INT	fc = 1 MHz		10	pF

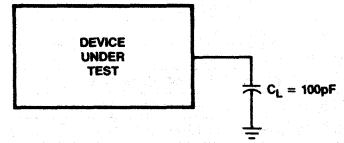
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006600

AC testing: inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." The clock is driven at 4.3V and 0.25V timing measurements are made at 1.5V for both a logic "1" and "0."

SWITCHING TEST LOAD CIRCUIT



TC001870

C_L includes jig capacitance

SWITCHING CHARACTERISTICS
TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8087		8087-2		Units
			Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	ns
TCLCH	CLK Low Time		118		68		ns
TCHCL	CLK High Time		69		44		ns
TCH_1CH_2	CLK Rise Time	From 1.0 to 3.5V		10		10	ns
TCL_2CL_1	CLK Fall Time	From 3.5 to 1.0V		10		10	ns
TDVCL	Data In Set-up Time		30		20		ns
TCLDX	Data In Hold Time		10		10		ns
TRYHCH	READY Set-up Time		118		68		ns
TCHRYX	READY Hold Time		30		20		ns
TRYLCL	READY Inactive to CLK (See Note 3)		−8		−8		ns
TGVCH	RQ/GT Set-up Time		30		15		ns
TCHGX	RQ/GT Hold Time		40		30		ns
TQVCL	QS_{0-1} Set-up Time		30		30		ns
TCLQX	QS_{0-1} Hold Time		10		10		ns
TSACH	Status Active Set-up Time		30		30		ns
TSNCL	Status Inactive Set-up Time		30		30		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12	ns

SWITCHING CHARACTERISTICS
TIMING RESPONSES

Parameters	Description	Test Conditions	8087		8087-2		Units
			Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 20-100pF for all 8087 outputs (In addition to 8087 self-load)	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	ns
TRYHSH	Ready Active to Status Passive (See Note 2)			110		65	ns
TCHSV	Status Active Delay		10	110	10	60	ns
TCLSH	Status Inactive Delay		10	130	10	70	ns
TCLAV	Address Valid Delay		10	110	10	60	ns
TCLAX	Address Hold Time		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	ns
TCHDX	Data Hold Time		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	ns
TCHBV	BUSY and INT Valid Delay		10	150	10	85	ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30	ns
TCLGL	RQ/GT Active Delay	C_L = 40pF (in addition to 8087 self-load)	0	85	0	50	ns
TCLGH	RQ/GT Inactive Delay		0	85	0	50	ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12	ns

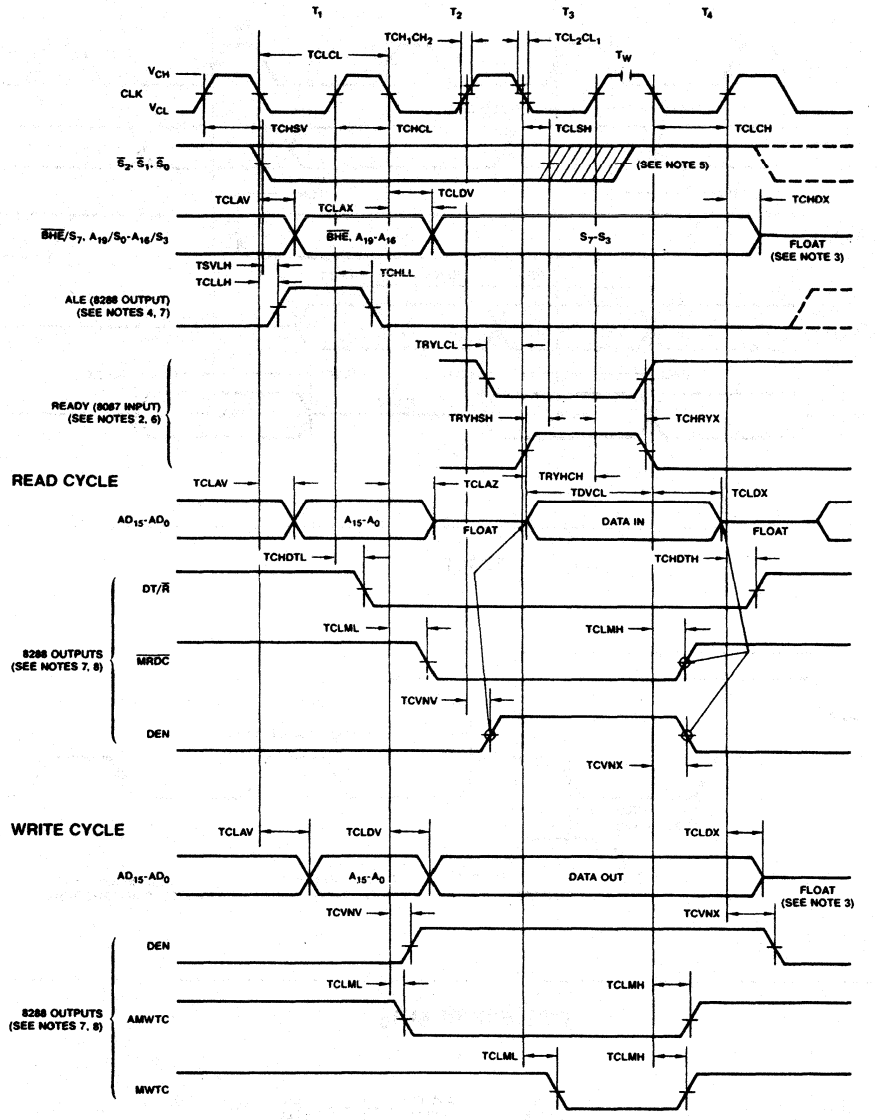
Notes: 1. Signal at 8284A or 8288 shown for reference only.
2. Applies only to T_3 and wait states.
3. Applies only to T_2 state (8ns into T_3).

02037B

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING WAVEFORMS

MASTER MODE

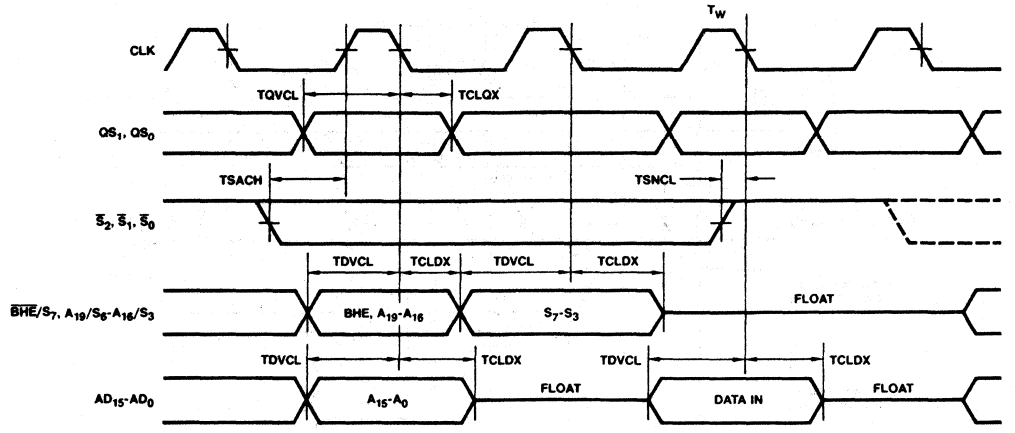


WF006610

Notes:
1. All signals switch between V_{OL} and V_{OH} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3 and T_W to determine if T_W machine states are to be inserted.
3. The local bus floats only if the 8087 is returning control to the 8086/8088.
4. ALE rises at later of (TSVLH, TCLLH).
5. Status inactive in state just prior to T_4.
6. Ready should remain active until S_{0-2} become inactive.
7. Signals at 8284A or 8288 are shown for reference only.
8. The issuance of 8288 command and control signals $\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$ and DEN lags the active high 8288 CEN.
9. All timing measurements are made at 1.5V unless otherwise noted.

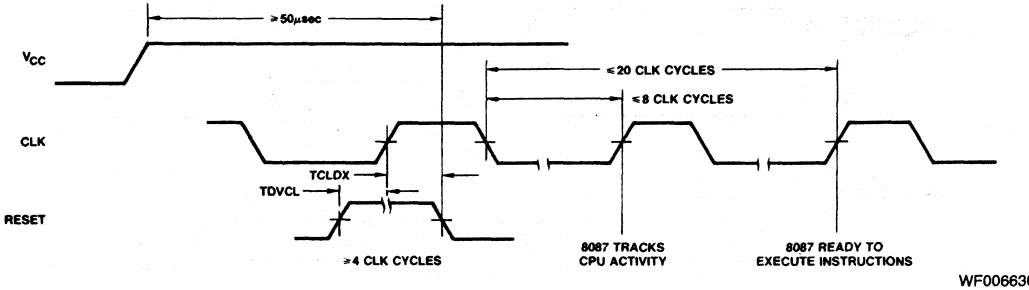
SWITCHING WAVEFORMS (Cont.)

PASSIVE MODE



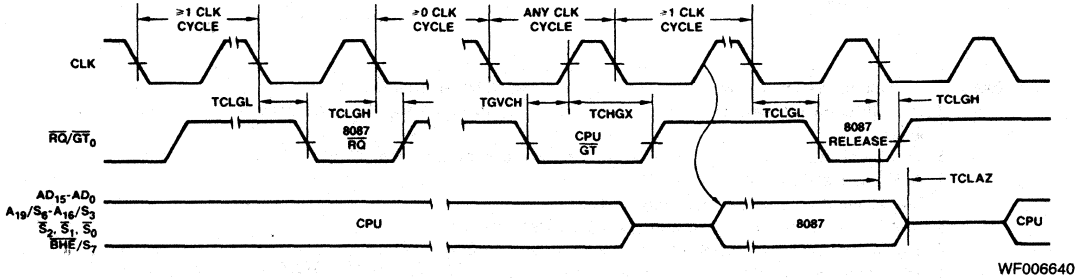
WF006620

RESET



WF006630

REQUEST/GRANT$_0$



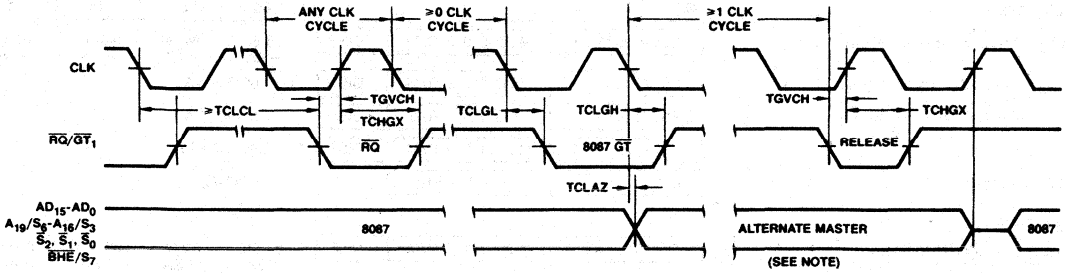
WF006640

02037B

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING WAVEFORMS (Cont.)

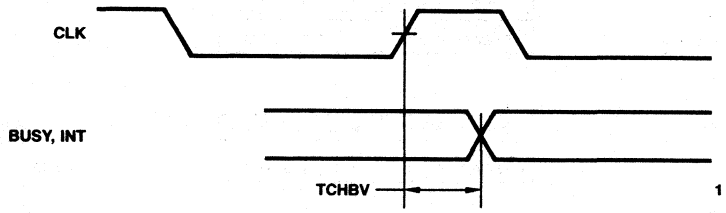
REQUEST/GRANT$_1$



WF006580

Note: Alternate master may not drive the buses outside of the region shown without risking contention.

BUSY AND INTERRUPT



WF006590

Table 5. 8087 Extensions to the 86/186 Instructions Sets

		Optional 8, 16 Bit Displacement	Clock Count Range			
Data Transfer			32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FLD = LOAD	MF =		00	01	10	11
Integer/Real Memory to ST(0)	ESCAPE MF 1 | MOD 0 0 0 R/M	DISP	38-56 +EA	52-60 +EA	40-60 +EA	46-54 +EA
Long Integer Memory to ST(0)	ESCAPE 1 1 1 | MOD 1 0 1 R/M	DISP	60-68+ EA			
Temporary Real Memory to ST(0)	ESCAPE 0 1 1 | MOD 1 0 1 R/M	DISP	53-65+ EA			
BCD Memory to ST(0)	ESCAPE 1 1 1 | MOD 1 0 0 R/M	DISP	290-310+ EA			
ST(i) to ST(0)	ESCAPE 0 0 1 | 1 1 0 0 0 ST(i)		17-22			
FST = STORE						
ST(0) to Integer/Real Memory	ESCAPE MF 1 | MOD 0 1 0 R/M	DISP	84-90 +EA	82-92 +EA	96-104 +EA	80-90 +EA
ST(0) to ST(i)	ESCAPE 1 0 1 | 1 1 0 1 0 ST(i)		15-22			
FSTP = STORE AND POP						
ST(0) to Integer/Real Memory	ESCAPE MF 1 | MOD 0 1 1 R/M	DISP	86-92 +EA	84-94 +EA	98-106 +EA	82-92 +EA
ST(0) to Long Integer Memory	ESCAPE 1 1 1 | MOD 1 1 1 R/M	DISP	95-105+ EA			
ST(0) to Temporary Real Memory	ESCAPE 0 1 1 | MOD 1 1 1 R/M	DISP	52-58+ EA			
ST(0) to BCD Memory	ESCAPE 1 1 1 | MOD 1 1 0 R/M	DISP	520-540+ EA			
ST(0) to ST(i)	ESCAPE 1 0 1 | 1 1 0 1 1 ST(i)		17-24			
FXCH = Exchange ST(i) and ST(0)	ESCAPE 0 0 1 | 1 1 0 0 1 ST(i)		10-15			
Comparison						
FCOM = Compare						
Integer/Real Memory to ST(0)	ESCAPE MF 0 | MOD 0 1 0 R/M	DISP	60-70 +EA	78-91 +EA	65-75 +EA	72-86 +EA
ST(i) to ST(0)	ESCAPE 0 0 0 | 1 1 0 1 0 ST(i)		40-50			
FCOMP = Compare and Pop						
Integer/Real Memory to ST(0)	ESCAPE MF 0 | MOD 0 1 1 R/M	DISP	63-73 +EA	80-93 +EA	67-77 +EA	74-88 +EA
ST(i) to ST(0)	ESCAPE 0 0 0 | 1 1 0 1 1 ST(i)		45-52			
FCOMPP = Compare ST(1) to ST(0) and Pop Twice	ESCAPE 1 1 0 | 1 1 0 1 1 0 0 1		45-55			
FTST = Test ST(0)	ESCAPE 0 0 1 | 1 1 1 0 0 1 0 0		38-48			
FXAM = Examine ST(0)	ESCAPE 0 0 1 | 1 1 1 0 0 1 0 1		12-23			
Constants						
FLDZ = LOAD +0.0 into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 1 1 0		11-17			
FLD1 = LOAD +1.0 into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 0 0 0		15-21			
FLDPI = LOAD π into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 0 1 1		16-22			
FLDL2T = LOAD $\log_2 10$ into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 0 0 1		16-22			
FLDL2E = LOAD $\log_2 e$ into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 0 1 0		15-21			
FLDLG2 = LOAD $\log_{10} 2$ into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 1 0 0		18-24			
FLDLN2 = LOAD $\log_e 2$ into ST(0)	ESCAPE 0 0 1 | 1 1 1 0 1 1 0 1		17-23			
Arithmetic						
FADD = Addition						
Integer/Real Memory with ST(0)	ESCAPE MF 0 | MOD 0 0 0 R/M	DISP	90-120 +EA	108-143 +EA	95-125 +EA	102-137 +EA
ST(i) and ST(0)	ESCAPE d P 0 | 1 1 0 0 0 ST(i)		70-100 (Note 1)			

Note: 1. If P = 1 then add 5 clocks.

02037B

Refer to page 7-1 for Essential Information on Military Devices

Table 5. 8087 Extensions to the 86/186 Instructions Sets (Cont.)

Arithmetic (Cont.)			Optional 8, 16 Bit Displacement	Clock Count Range			
				32 Bit Real	32 Bit Integer	64 Bit Real	16 Bit Integer
FSUB = Subtraction	MF	=		00	01	10	11
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 1 0 R R/M	DISP	90-120 +EA	108-143 +EA	95-125 +EA	102-137 +EA
ST(i) and ST(0)	ESCAPE d P 0	1 1 1 0 R R/M		70-100 (Note 1)			
FMUL = Multiplication							
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 0 0 1 R/M	DISP	110-125 +EA	130-144 +EA	112-168 +EA	124-138 +EA
ST(i) and ST(0)	ESCAPE d P 0	1 1 0 0 1 R/M		90-145 (Note 1)			
FDIV = Division							
Integer/Real Memory with ST(0)	ESCAPE MF 0	MOD 1 1 R R/M	DISP	215-225 +EA	230-243 +EA	220-230 +EA	224-238 +EA>
ST(i) and ST(0)	ESCAPE d P 0	1 1 1 1 R R/M		193-203 (Note 1)			
FSQRT = Square Root of ST(0)	ESCAPE 0 0 1	1 1 1 1 1 0 1 0		180-186			
FSCALE = Scale ST(0) by ST(1)	ESCAPE 0 0 1	1 1 1 1 1 1 0 1		32-38			
FPREM = Partial Remainder of ST(0) ÷ ST(1)	ESCAPE 0 0 1	1 1 1 1 1 0 0 0		15-190			
FRNDINT = Round ST(0) to Integer	ESCAPE 0 0 1	1 1 1 1 1 1 0 0		16-50			
FXTRACT = Extract Components of ST(0)	ESCAPE 0 0 1	1 1 1 1 0 1 0 0		27-55			
FABS = Absolute Value of ST(0)	ESCAPE 0 0 1	1 1 1 0 0 0 0 1		10-17			
FCHS = Change Sign of ST(0)	ESCAPE 0 0 1	1 1 1 0 0 0 0 0		10-17			
Transcendental							
FPTAN = Partial Tangent of ST(0)	ESCAPE 0 0 1	1 1 1 1 0 0 1 0		30-540			
FPATAN = Partial Arctangent of ST(0) ÷ ST(1)	ESCAPE 0 0 1	1 1 1 1 0 0 1 1		250-800			
F2XM1 = $2^{ST(0)} - 1$	ESCAPE 0 0 1	1 1 1 1 0 0 0 0		310-630			
FYL2X = ST(1) · Log_2 [ST(0)]	ESCAPE 0 0 1	1 1 1 1 0 0 0 1		900-1100			
FYL2XP1 = ST(1) · Log_2 [ST(0) + 1]	ESCAPE 0 0 1	1 1 1 1 1 0 0 1		700-1000			
Processor Control							
FINIT = Initialized 8087	ESCAPE 0 1 1	1 1 1 0 0 0 1 1		2-8			
FENI = Enable Interrupts	ESCAPE 0 1 1	1 1 1 0 0 0 0 0		2-8			
FDISI = Disable Interrupts	ESCAPE 0 1 1	1 1 1 0 0 0 0 1		2-8			
FLDCW = Load Control Word	ESCAPE 0 0 1	MOD 1 0 1 R/M	DISP	7-14+EA			
FSTCW = Store Control Word	ESCAPE 0 0 1	MOD 1 1 1 R/M	DISP	12-18+EA			
FSTSW = Store Status Word	ESCAPE 1 0 1	MOD 1 1 1 R/M	DISP	12-18+EA			
FCLEX = Clear Exceptions	ESCAPE 0 1 1	1 1 1 0 0 0 1 0		2-8			
FSTENV = Store Environment	ESCAPE 0 0 1	MOD 1 1 0 R/M	DISP	40-50+EA			
FLDENV = Load Environment	ESCAPE 0 0 1	MOD 1 0 0 R/M	DISP	35-45+EA			
FSAVE = Save State	ESCAPE 1 0 1	MOD 1 1 0 R/M	DISP	197-207+EA			
FRSTOR = Restore State	ESCAPE 1 0 1	MOD 1 0 0 R/M	DISP	197-207+EA			
FINCSTP = Increment Stack Pointer	ESCAPE 0 0 1	1 1 1 1 0 1 1 1		6-12			
FDECSTP = Decrement Stack Pointer	ESCAPE 0 0 1	1 1 1 1 0 1 1 0		6-12			
FFREE = Free ST(i)	ESCAPE 1 0 1	1 1 0 0 0 ST(i)		9-16			
FNOP = No Operation	ESCAPE 0 0 1	1 1 0 1 0 0 0 0		10-16			
FWAIT = CPU Wait for 8087	1 0 0 1 1 0 1 1			3 + 5n*			

*n = number of times CPU examines TEST line before 8087 lowers BUSY.

NOTES:

1. if mod = 00 then DISP = 0*, disp-low and disp-high are absent
   if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
   if mod = 10 then DISP = disp-high; disp-low
   if mod = 11 then r/m is treated as an ST(i) field
2. if r/m = 000 then EA = (BX) + (SI) + DISP
   if r/m = 001 then EA = (BX) + (DI) + DISP
   if r/m = 010 then EA = (BP) + (SI) + DISP
   if r/m = 011 then EA = (BP) + (DI) + DISP
   if r/m = 100 then EA = (SI) + DISP
   if r/m = 101 then EA = (DI) + DISP
   if r/m = 110 then EA = (BP) + DISP
   if r/m = 111 then EA = (BX) + DISP

   *except if mode = 000 and r/m = 110 then EA = disp-high; disp-low.
3. MF = Memory Format
   00 — 32-bit Real
   01 — 32-bit Integer
   10 — 64-bit Real
   11 — 16-bit Integer
4. ST(0) = Current stack top
   ST(i) = i^{th} register below stack top
5. d = Destination
   0 — Destination is ST(0)
   1 — Destination is ST(i)
6. P = Pop
   0 — No pop
   1 — Pop ST(0)
7. R = Reverse: When d = 1 reverse the sense of R
   0 — Destination (op) Source
   1 — Source (op) Destination
8. For **FSQRT:** $-0 \leqslant ST(0) \leqslant +\chi$
   For **FSCALE:** $-2^{15} \leqslant ST(1) < +2^{15}$ and ST(1) integer
   For **F2XM1:** $0 \leqslant ST(0) \leqslant 2^{-1}$
   For **FYL2X:** $0 < ST(0) < \chi$
   $-\chi < ST(1) < +\chi$
   For **FYL2XP1:** $0 \leqslant |ST(0)| < (2-\sqrt{2})/2$
   $-\chi < ST(1) < \chi$
   For **FPTAN:** $0 \leqslant ST(0) \leqslant \pi/4$
   For **FPATAN:** $0 \leqslant ST(0) < ST(1) < +\chi$

8088

8-Bit Microprocessor CPU
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

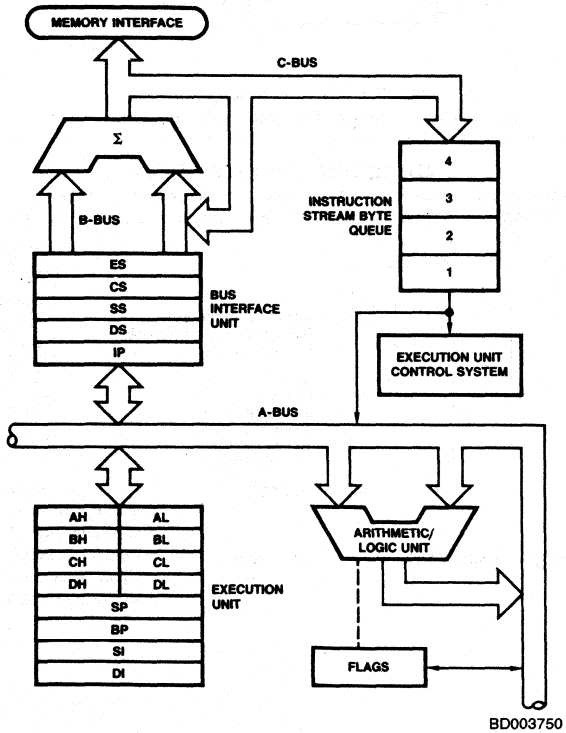
- 8-bit data bus, 16-bit internal architecture
- Directly addresses 1 Mbyte of memory
- Software compatible with 8086 CPU
- Byte, word, and block operations
- 24 operand addressing modes
- Powerful instruction set
- Efficient high level language implementation
- Three block options: 5MHz 8088
  8MHz 8088-2
  10MHz 8088-1

GENERAL DESCRIPTION

The 8088 CPU is an 8-bit processor designed around the 8086 internal structure. Most functions of the 8088 are identical to the equivalent 8086 functions. The pinout is slightly different. The 8088 handles the external bus the same way the 8086 does, but it handles only 8 bits at a time. Sixteen-bit words are fetched or written in two consecutive bus cycles. Both processors will appear identical to the software engineer, with the exception of execution time.

The 8088 is made with N-channel silicon gate technology and is packaged in a 40-pin Plastic dip, CERDIP or Plastic Leaded Chip Carrier.

BLOCK DIAGRAM

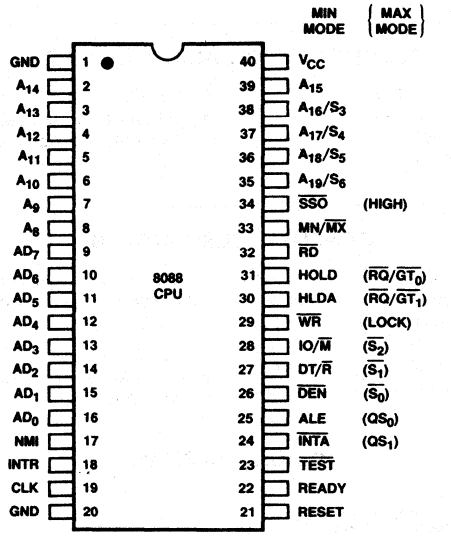


02338B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40
P-40

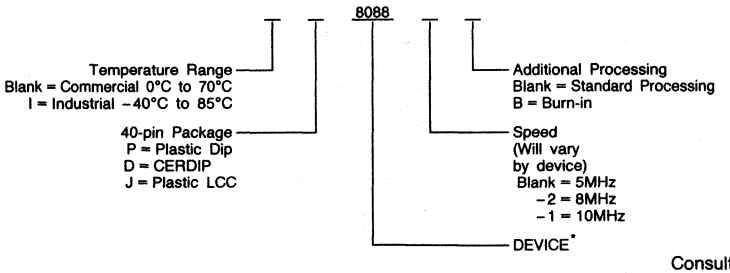


CD005520

Note: Pin 1 is marked for orientation
Also available in PLCC. See Section 7 for pinout details.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations	
8088 8088-2	P, D, ID, J
8088B 8088-2B	P, D, ID, J
8088-1 8088-1B	P, D, J
8088 8088-2	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

The following pin function descriptions are for 8088 systems in either minimum or maximum mode. The "local bus" in these descriptions is the direct multiplexed bus interface connection to the 8088 (without regard to additional bus buffers).

Pin No.	Name	I/O	Description
9-16	AD_7–AD_0	I/O	Address Data Bus. These lines constitute the time multiplexed memory/IO address (T_1) and data (T_2, T_3, T_W, and T_4) bus. These lines are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
39, 2-8	A_{15}–A_8	O	Address Bus. These lines provide address bits 8 through 15 for the entire bus cycle (T_1–T_4). These lines do not have to be latched by ALE to remain valid. A_{15}–A_8 are active HIGH and float to 3-state OFF during interrupt acknowledge and local bus "hold acknowledge."
35-38	A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3	O	Address/Status. During T_1, these are the four most significant address lines for memory operations. During I/O operations, these lines are LOW. During memory and I/O operations, status information is available on these lines during T_2, T_3, T_W, and T_4. S_6 is always LOW. The status of the interrupt enable flat bit (S_5) is updated at the beginning of each clock cycle. S_4 and S_3 are encoded as shown. This information indicates which segment register is presently being used for data accessing. These lines float to 3-state OFF during local bus "hold acknowledge." S_4 / S_3 / Characteristics: 0 (LOW) / 0 / Alternate Data 0 / 1 / Stack 1 (HIGH) / 0 / Code or None 1 / 1 / Data S_6 is 0 (LOW)
32	$\overline{RD}$	O	Read. Read strobe indicates that the processor is performing a memory or I/O read cycle, depending on the state of the IO/$\overline{M}$ pin or S_2. This signal is used to read devices which reside on the 8088 local bus. $\overline{RD}$ is active LOW during T_2, T_3 and T_W of any read cycle, and is guaranteed to remain HIGH in T_2 until the 8088 local bus has floated. This signal floats to 3-state OFF in "hold acknowledge."
22	READY	I	READY. The acknowledgement from the addressed memory or I/O device that it will complete the data transfer. The RDY signal from memory or I/O is synchronized by the 8284 clock generator to form READY. This signal is active HIGH. The 8088 READY input is not synchronized. Correct operation is not guaranteed if the set-up and hold times are not met.
18	INTR	I	Interrupt Request. A level-triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. A subroutine is vectored to via an interrupt vector lookup table located in system memory. It can be internally masked by software resetting the interrupt enable bit. INTR is internally synchronized. This signal is active HIGH.
23	$\overline{TEST}$	I	$\overline{TEST}$. Input is examined by the "wait for test" instruction. If the $\overline{TEST}$ input is LOW, execution continues; otherwise, the processor waits in an "Idle" state. This input is synchronized internally during each clock cycle on the leading edge of CLK.
17	NMI	I	Non-Maskable Interrupt. An edge-triggered input which causes a type 2 interrupt. A subroutine is vectored to via an interrupt vector lookup table located in system memory. NMI is not maskable internally by software. A transition from a LOW to HIGH initiates the interrupt at the end of the current instruction. This Input is internally synchronized.
21	RESET	I	RESET. Causes the processor to immediately terminate its present activity. The signal must be active HIGH for at least four clock cycles. It restarts execution, as described in the instruction set description, when RESET returns LOW. RESET is internally synchronized.
19	CLK	I	Clock. Provides the basic timing for the processor and bus controller. It is asymmetric with a 33% duty cycle to provide optimized internal timing.
40	V_{CC}		V_{CC}. The +5V ±10% power supply pin.
1, 20	GND		GND. The ground pins.
33	MIN/$\overline{MX}$	I	Minimum/Maximum. Indicates what mode the processor is to operate in. The two modes are discussed in the following sections.
28	IO/$\overline{M}$	O	Status Line. An inverted maximum mode $\overline{S_2}$. It is used to distinguish a memory access from an I/O access. IO/$\overline{M}$ becomes valid in the T_4 preceding a bus cycle and remains valid until the final T_4 of the cycle (I/O = HIGH, M = LOW). IO/$\overline{M}$ floats to 3-state OFF in local bus "hold acknowledge."
29	$\overline{WR}$	O	Write. Strobe indicates that the processor is performing a write memory or write I/O cycle, depending on the state of the IO/$\overline{M}$ signal. WR is active for T_2, T_3, and T_W of any write cycle. It is active LOW and floats to 3-state OFF in local bus "hold acknowledge."
24	$\overline{INTA}$	O	INTA. Used as a read strobe for interrupt acknowledge cycles. It is active LOW during T_2, T_3 and T_W of each interrupt acknowledge cycle.
25	ALE	O	Address Latch Enable. Provided by the processor to latch the address into 8282/8283 address latch. It is a HIGH pulse active during clock low of T_1 of any bus cycle. Note that ALE is never floated.
27	DT/$\overline{R}$	O	Data Transmit/Receive. Needed in a minimum system that desires to use an 8286/8287 data bus transceiver. It is used to control the direction of data flow through the transceiver. Logically DT/$\overline{R}$ is equivalent to $\overline{S_1}$ in the maximum mode, and its timing is the same as for IO/$\overline{M}$ (T = HIGH, R = LOW.) This signal floats to 3-state OFF in local bus "hold acknowledge."
26	$\overline{DEN}$	O	Data Enable. Provided as an output enable for the 8286/8287 in a minimum system that uses the transceiver. $\overline{DEN}$ is active LOW during each memory and I/O access and for $\overline{INTA}$ cycles. For a read or $\overline{INTA}$ cycle, it is active from the middle of T_2 until the middle of T_4; while for a write cycle, it is active from the beginning of T_2 until the middle of T_4. $\overline{DEN}$ floats to 3-state OFF during local bus "hold acknowledge."

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
31, 30	HOLD, HLDA	I/O	HOLD. Indicates that another master is requesting a local bus "hold." To be acknowledged, HOLD must be active HIGH. The processor receiving the "hold" request will issue HLDA (HIGH) as an acknowledgement in the middle of a T_4 or T_1 clock cycle. Simultaneous with the issuance of HLDA, the processor will float the local bus and control lines. After HOLD is detected as being LOW, the processor lowers HLDA, and when the processor needs to run another cycle, it will again drive the local bus and control lines. HOLD is not an asynchronous input. External synchronization should be provided if the system cannot otherwise guarantee the set-up time.
34	$\overline{SSO}$	O	Status Line. Logically equivalent to $\overline{S0}$ in the maximum mode. The combination of $\overline{SSO}$, IO/$\overline{M}$ and DT/$\overline{R}$ allows the system to completely decode the current bus cycle status.

IO/$\overline{M}$	DT/$\overline{R}$	$\overline{SSO}$	Characteristics
1 (HIGH)	0	0	Interrupt Acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0 (LOW)	0	0	Code Access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive

Pin No.	Name	I/O	Description
28-26	$\overline{S}_2$, $\overline{S}_1$, $\overline{S}_0$	O	Status. Active during clock high of T_4, T_1, and T_2 and is returned to the passive state (1, 1, 1) during T_3 or during T_W when READY is HIGH. This status is used by the 8288 bus controller to generate all memory and I/O access control signals. Any change by $\overline{S}_2$, $\overline{S}_1$, or $\overline{S}_0$ during T_4 is used to indicate the beginning of a bus cycle, and the return to the passive state in T_3 or T_W is used to indicate the end of a bus cycle. These signals float to 3-state OFF during "hold acknowledge." During the first clock cycle after RESET becomes active, these signals are active HIGH. After this first clock, they float to 3-state OFF.

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

Pin No.	Name	I/O	Description
31, 30	$\overline{RQ}/\overline{GT}_0$, $\overline{RQ}/\overline{GT}_1$	I/O	Request/Grant. Pins are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle. Each pin is bidirectional with $\overline{RQ}/\overline{GT}_0$ having higher priority than $\overline{RQ}/\overline{GT}_1$. $\overline{RQ}/\overline{GT}$ has an internal pull-up resistor, so may be left unconnected. The request/grant sequence is as follows: 1. A pulse of one CLK wide from another local bus master indicates a local bus request ("hold") to the 8088 (pulse 1). 2. During a T_4 or T_1 clock cycle, a pulse one clock wide from the 8088 to the requesting master (pulse 2), indicates that the 8088 has allowed the local bus to float and that it will enter the "hold acknowledge" state at the next CLK. The CPU's bus interface unit is disconnected logically from the local bus during "hold acknowledge." The same rules as for HOLD/HLDA apply as for when the bus is released. 3. A pulse one CLK wide from the requesting master indicates to the 8088 (pulse 3) that the "hold" request is about to end and that the 8088 can reclaim the local bus at the next CLK. The CPU then enters T_4. Each master-master exchange of the local bus is a sequence of three pulses. There must be one idle CLK cycle after each bus exchange. Pulses are active LOW. If the request is made while the CPU is performing a memory cycle, it will release the local bus during T_4 of the cycle when all the following conditions are met: 1. Request occurs on or before T_2. 2. Current cycle is not the low bit of a word. 3. Current cycle is not the first acknowledge of an interrupt acknowledge sequence. 4. A locked instruction is not currently executing. If the local bus is idle when the request is made, two possible events will follow: 1. Local bus will be released during the next clock. 2. A memory cycle will start within 3 clocks. Now the four rules for a currently active memory cycle apply with condition number 1 already satisfied.
29	$\overline{LOCK}$	O	$\overline{LOCK}$. Indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active (LOW). The $\overline{LOCK}$ signal is activated by the "LOCK" prefix instruction and remains active until the completion of the next instruction. This signal is active LOW and floats to 3-state off in "hold acknowledge."

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
24, 25	QS_1, QS_0	O	Queue Status. Provides status to allow external tracking of the internal 8088 instruction queue. The queue status is valid during the CLK cycle after which the queue operation is performed.
34	–	O	Pin 34 is always HIGH in the maximum mode.

QS_1	QS_0	Characteristics
0 (LOW)	0	No Operation
0	1	First Byte of Opcode from Queue
1 (HIGH)	0	Empty the Queue
1	1	Subsequent Byte from Queue

DETAILED DESCRIPTION

The 8088 Compared to the 8086

- The queue length is 4 bytes in the 8088; whereas, the 8086 queue contains 6 bytes, or three words. The queue was shortened to prevent overuse of the bus by the BIU when prefetching instructions. This was required because of the additional time necessary to fetch instructions 8 bits at a time.
- To further optimize the queue, the prefetching algorithm was changed. The 8088 BIU will fetch a new instruction to load into the queue each time there is a 1 byte hole (space available) in the queue. The 8086 waits until a 2-byte space is available.
- The internal execution time of the instruction set is affected by the 8-bit interface. All 16-bit fetches and writes from/to memory take an additional four clock cycles. The CPU is also limited by the speed of instruction fetches. This latter problem only occurs when a series of simple operations occurs. When the more sophisticated instructions of the 8088 are being used, the queue has time to fill and the execution proceeds as fast as the execution unit will allow.

The 8088 and 8086 are completely software compatible by virtue of their identical execution units. Software that is system dependent may not be completely transferable, but software that is not system dependent will operate equally as well on an 8088 or an 8086.

The hardware interface of the 8088 contains the major differences between the two CPUs. The pin assignments are nearly identical, however, with the following functional changes:

- $A_8 - A_{15}$ – These pins are only address outputs on the 8088. These address lines are latched internally and remain valid throughout a bus cycle in a manner similar to the 8085 upper address lines.
- $\overline{BHE}$ has no meaning on the 8088 and has been eliminated.
- $\overline{SSO}$ provides the $\overline{S0}$ status information in the minimum mode. This output occurs on pin 34 in minimum mode only. DT/$\overline{R}$, IO/$\overline{M}$, and $\overline{SSO}$ provide the complete bus status in minimum mode.
- IO/$\overline{M}$ has been inverted to be compatible with the MCS-85 bus structure.
- ALE is delayed by one clock cycle in the minimum mode when entering HALT, to allow the status to be latched with ALE.

I/O Addressing

In the 8088, I/O operations can address up to a maximum of 64K I/O registers. The I/O address appears in the same format as the memory address on bus lines $A_{15} - A_0$. The address lines $A_{19} - A_{16}$ are zero in I/O operations. The variable I/O instructions, which use register DX as a pointer, have full address capability, while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space. I/O ports are addressed in the same manner as memory locations.

Designers familiar with the 8085 or upgrading an 8085 design should note that the 8085 addresses I/O with an 8-bit address on both halves of the 16-bit address bus. The 8088 uses a full 16-bit address of its lower 16 address lines.

Bus Operation

The 8088 address/data bus is broken into three parts – the lower eight address/data bits ($AD_0 - AD_7$), the middle eight address bits ($A_8 - A_{15}$), and the upper four address bits (A_{16}-A_{19}). The address/data bits and the highest four address bits are time multiplexed. This technique provides the most efficient use of pins on the processor, permitting the use of a standard 40 lead package. The middle eight address bits are not multiplexed; i.e., they remain valid throughout each bus cycle. In addition, the bus can be demultiplexed at the processor with a single address latch if a standard, non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T1, T2, T3, and T4. The address is emitted from the processor during T1 and data transfer occurs on the bus during T3 and T4. T2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "wait" states (Tw) are inserted between T3 and T4. Each inserted "wait" state is of the same duration as a CLK cycle. Periods can occur between 8088 driven bus cycles. These are referred to as "idle" states (Ti), or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T1 of any bus cycle, the ALE (address latch enable), signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$ are used by the bus controller, in maximum mode, to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S3 through S6 are multiplexed with high order address bits and are therefore valid during T2 through T4. S3 and S4 indicate which segment register was used for this bus cycle in forming the address according to the following table:

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data (extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

S5 is a reflection of the PSW interrupt enable bit. S6 is always equal to 0.

External Interface

Processor Reset and Initialization

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8088 RESET is required to be HIGH for greater than four clock cycles. The 8088 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 7 clock cycles. After this interval the 8088 operates normally, beginning with the instruction in absolute location FFFF0H. (See Figure 3.) The RESET input is internally synchronized to the processor clock. At initialization, the HIGH to LOW transition of RESET must occur no sooner than 50 μs after power up, to allow complete initialization of the 8088.

If INTR is asserted sooner than nine clock cycles after the end of RESET, the processor may execute one instruction before responding to the interrupt.

All 3-state outputs float to 3-state OFF during RESET. Status is active in the idle state for the first clock after RESET becomes active and then floats to 3-state OFF.

Interrupt Operations

Interrupt operations fall into two classes: software or hardware initiated. The software initiated interrupts and software aspects of hardware interrupts are specified in the instruction set description in the iAPX 88 book or the iAPX 86, 88 User's Manual. Hardware interrupts can be classified as nonmaskable or maskable.

Interrupts result in a transfer of control to a new program location. A 256 element table containing address pointers to the interrupt service program locations resides in absolute locations 0 through 3FFH (see Figure 3), which are reserved for this purpose. Each element in the table is 4 bytes in size and corresponds to an interrupt "type." An interrupting device supplies an 8-bit type number, during the interrupt acknowledge sequence, which is used to vector through the appropriate element to the new interrupt service program location.

Non-Maskable Interrupt (NMI)

The processor provides a single non-maskable interrupt (NMI) pin which has higher priority than the maskable interrupt request (INTR) pin. A typical use would be to activate a power failure routine. The NMI is edge-triggered on a LOW to HIGH transition. The activation of this pin causes a type 2 interrupt.

NMI is required to have a duration in the HIGH state of greater than two clock cycles, but is not required to be synchronized to the clock. Any higher going transition of NMI is latched on-chip and will be serviced at the end of the current instruction or between whole moves (2 bytes in the case of word moves) of a block type instruction. Worst case response to NMI would be for multiply, divide, and variable shift instructions. There is no specification on the occurrence of the low-going edge; it may occur before, during, or after the servicing of NMI. Another high-going edge triggers another response if it occurs after the start of the NMI procedure. The signal must be free of logical spikes in general and be free of bounces on the low-going edge to avoid triggering extraneous responses.

Maskable Interrupt (INTR)

The 8088 provides a single interrupt request input (INTR) which can be masked internally by software with the resetting of the interrupt enable (IF) flag bit. The interrupt request signal is level triggered. It is internally synchronized during each clock cycle on the high-going edge of CLK. To be responded to, INTR must be present (HIGH) during the clock period preceding the end of the current instruction or the end of a whole move for a block type instruction. During interrupt response sequence, further interrupts are disabled. The enable bit is reset as part of the response to any interrupt (INTR, NMI, software interrupt, or single step), although the FLAGS register which is automatically pushed onto the stack reflects the state of the processor prior to the interrupt. Until the old FLAGS register is restored, the enable bit will be zero unless specifically set by an instruction.

During the response sequence (See Figure 1), the processor executes two successive (back to back) interrupt acknowledge cycles. The 8088 emits the LOCK signal (maximum mode only) from T2 of the first bus cycle until T2 of the second. A local bus "hold" request will not be honored until the end of the second bus cycle. In the second bus cycle, a byte is fetched from the external interrupt system (e.g., 8259A PIC) which identifies the source (type) of the interrupt. This byte is multiplied by four and used as a pointer into the interrupt vector lookup table. An INTR signal left HIGH will be continually responded to within the limitations of the enable bit and sample period. The interrupt return instruction includes a flags pop which returns the status of the original interrupt enable bit when it restores the flags.

HALT

When a software HALT instruction is executed, the processor indicates that it is entering the HALT state in one of two ways, depending upon which mode is strapped. In minimum mode, the processor issues ALE, delayed by one clock cycle, to allow the system to latch the halt status. Halt status is available on IO/$\overline{M}$, DT/$\overline{R}$, and $\overline{SSO}$. In maximum mode, the processor issues appropriate HALT status on $\overline{S2}$, $\overline{S1}$, and $\overline{S0}$, and the 8288 bus controller issues one ALE. The 8088 will not leave the HALT state when a local bus hold is entered while in HALT. In this case, the processor reissues the HALT indicator at the end of the local bus hold. An interrupt request or RESET will force the 8088 out of the HALT state.

Read/Modify/Write (Semaphore) Operations via LOCK

The LOCK status information is provided by the processor when consecutive bus cycles are required during the execution of an instruction. This allows the processor to perform read/modify/write operations on memory (via the "exchange register with memory" instruction), without another system bus master receiving intervening memory cycles. This is useful in multiprocessor system configurations to accomplish "test and set lock" operations. The $\overline{LOCK}$ signal is activated (LOW) in the clock cycle following decoding of the LOCK prefix instruction. It is deactivated at the end of the last bus cycle of the instruction following the LOCK prefix. While $\overline{LOCK}$ is active, a request on a $\overline{RQ}$/$\overline{GT}$ pin will be recorded, and then honored at the end of the LOCK.

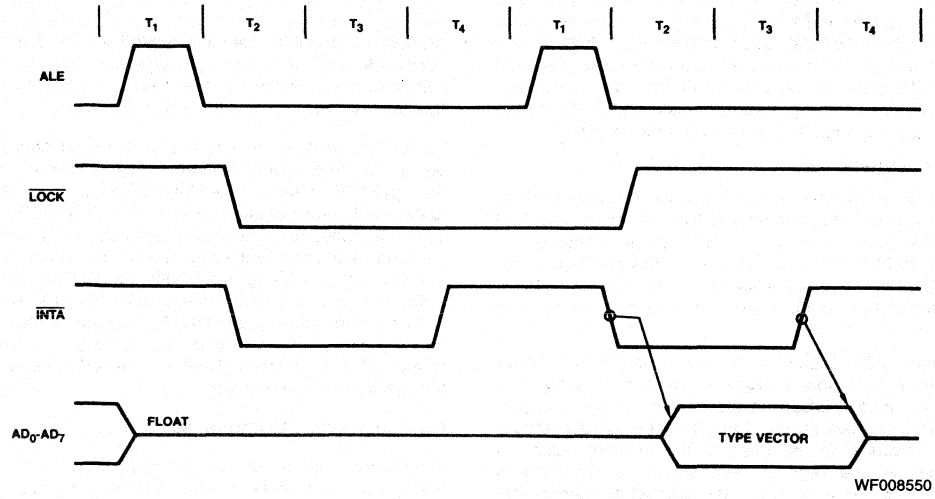


Figure 1. Interrupt Acknowledge Sequence

External Synchronization via TEST

As an alternative to interrupts, the 8088 provides a single software-testable input pin (TEST). This input is utilized by executing a WAIT instruction. The single WAIT instruction is repeatedly executed until the TEST input goes active (LOW). The execution of WAIT does not consume bus cycles once the queue is full.

If a local bus request occurs during WAIT execution, the 8088 3-states all output drivers. If interrupts are enabled, the 8088 will recognize interrupts and process them. The WAIT instruction is then refetched, and reexecuted.

Basic System Timing

In minimum mode, the MN/MX pin is strapped to V_{CC} and the processor emits bus control signals compatible with the 8085 bus structure. In maximum mode, the MN/MX pin is strapped to GND and the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals.

System Timing — Minimum System

The read cycle begins in T1 with the assertion of the address latch enable (ALE) signal. The trailing (low going) edge of this signal is used to latch the address information, which is valid on the address/data bus (AD0 – AD7) at this time, into the 8282/8283 latch. Address lines A8 through A15 do not need to be latched because they remain valid throughout the bus cycle. From T1 to T4 the IO/M signal indicates a memory or I/O operation. At T2 the address is removed from the address/data bus and the bus goes to a high impedance state. The read control signal is also asserted at T2. The read (RD) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later, valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver (8286/8287) is required to buffer the 8088 local bus, signals DT/R and DEN are provided by the 8088.

A write cycle also begins with the assertion of ALE and the emission of the address. The IO/M signal is again asserted to indicate a memory or I/O write operation. In T2, immediately following the address emission, the processor emits the data to be written into the addressed location. This data remains valid until at least the middle of T4. During T2, T3, and T_W, the processor asserts the write control signal. The write (WR) signal becomes active at the beginning of T2, as opposed to the read, which is delayed somewhat into T2 to provide time for the bus to float.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge (INTA) signal is asserted in place of the read (RD) signal and the address bus is floated. (See Figure 1.) In the second of two successive INTA cycles, a byte of information is read from the data bus, as supplied by the interrupt system logic (i.e. 8259A priority interrupt controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into the interrupt vector lookup table, as described earlier.

Bus Timing — Medium Complexity Systems

For medium complexity systems, the MN/MX pin is connected to GND and the 8288 bus controller is added to the system, as well as an 8282/8283 latch for latching the system address, and an 8286/8287 transceiver to allow for bus loading greater than the 8088 is capable of handling. Signals ALE, DEN, and DT/R are generated by the 8288 instead of the processor in this configuration, although their timing remains relatively the same. The 8088 status outputs (S2, S1, and S0) provide type of cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence, data is not valid at the leading edge of write. The 8286/8287 transceiver receives the usual T and OE inputs from the 8288's DT/R and DEN outputs.

The pointer into the interrupt vector table, which is passed during the second $\overline{INTA}$ cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8289A priority interrupt controller is positioned on the local bus, a TTL gate is required to disable the 8286/8287 transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll".

Memory Organization

The processor provides a 20-bit address to memory which locates the byte being referenced. The memory is organized as a linear array of up to 1 million bytes, addressed as 00000(H) to FFFFF(H). The memory is logically divided into code, data, extra data, and stack segments of up to 64K bytes each, with each segment falling on 16-byte boundaries. (See Figure 2.)

All memory references are made relative to base addresses contained in high speed segment registers. The segment types were chosen based on the addressing needs of programs. The segment register to be selected is automatically chosen according to the rules of the following table. All information in one segment type share the same logical attributes (e.g. code or data). By structuring memory into relocatable areas of similar characteristics and by automatically selecting segment registers, programs are shorter, faster, and more structured.

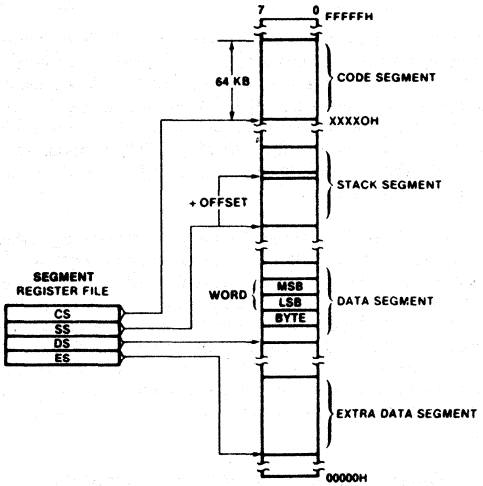


Figure 2. Memory Organization

Word (16-bit) operands can be located on even or odd address boundaries. For address and data operands, the least significant byte of the word is stored in the lower valued address location and the most significant byte in the next higher address location. The BIU will automatically execute two fetch or write cycles for 16-bit operands.

Certain locations in memory are reserved for specific CPU operations. (See Figure 3.) Locations from addresses FFFF0H through FFFFFH are reserved for operations including a jump to the initial system initialization routine. Following RESET, the CPU will always begin execution at location FFFF0H where the jump must be located. Locations 00000H through 003FFH are reserved for interrupt operations. Four-byte pointers consisting of a 16-bit segment address and a 16-bit offset address direct program flow to one of the 256 possible interrupt service routines. The pointer elements are assumed to have been stored at their respective places in reserved memory prior to the occurrence of interrupts.

Minimum and Maximum Modes

The requirements for supporting minimum and maximum 8088 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 8088 is equipped with a strap pin (MN/$\overline{MX}$) which defines the system configuration. The definition of a certain subset of the pins changes, dependent on the condition of the strap pin. When the MN/$\overline{MX}$ pin is strapped to GND, the 8088 defines pins 24 through 31 and 34 in maximum mode. When the MN/$\overline{MX}$ pin is strapped to V_{CC}, the 8088 generates bus control signals itself on pins 24 through 31 and 34.

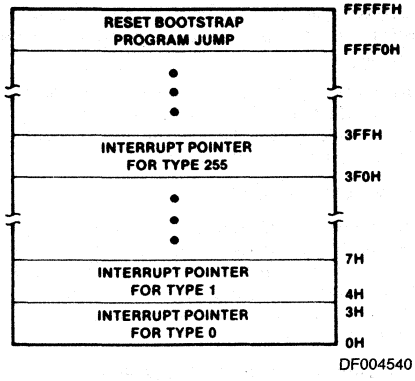


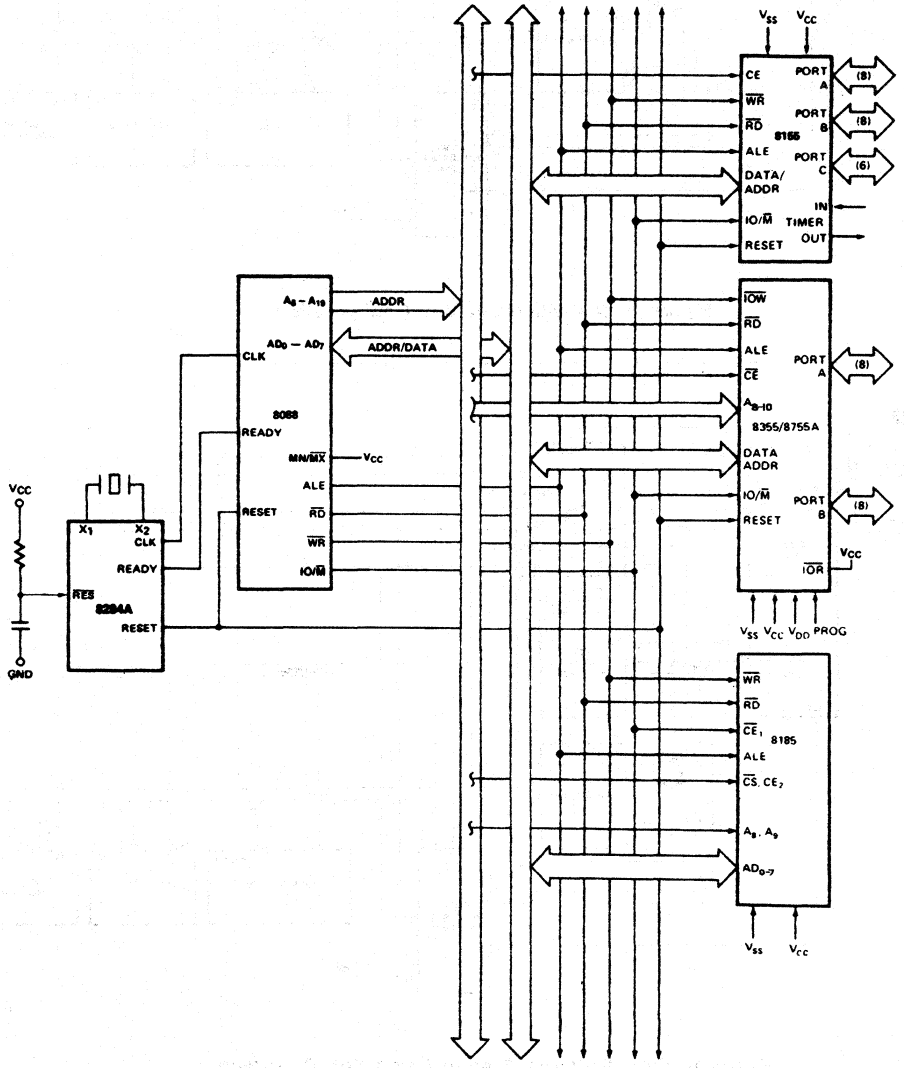
Figure 3. Reserved Memory Locations

Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when: relative to stack, destination of string operation, or explicitly overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

The minimum mode 8088 can be used with either a multiplexed or demultiplexed bus. The multiplexed bus configuration is compatible with the MCS-85™ multiplexed bus peripherals (8155, 8156, 8355, 8755A, and 8185). This configuration (see Figure 4) provides the user with a minimum chip count system. This architecture provides the 8088 processing power in a highly integrated form.

The demultiplexed mode requires one latch (for 64K addressability) or two latches (for a full megabyte of addressing). A third latch can be used for buffering if the address bus loading requires it. An 8286 or 8287 transceiver can also be used if data bus buffering is required. (See Figure 5.) The 8088 provides $\overline{DEN}$ and DT/$\overline{R}$ to control the transceiver, and ALE to latch the addresses. This configuration of the minimum mode provides the standard demultiplexed bus structure with heavy bus buffering and relaxed bus timing requirements.

The maximum mode employs the 8288 bus controller. (See Figure 6.) The 8288 decodes status lines $\overline{S0}$, $\overline{S1}$, and $\overline{S2}$, and provides the system with all bus control signals. Moving the bus control to the 8288 provides better source and sink current capability to the control lines, and frees the 8088 pins for extended large system features. Hardware lock, queue status, and two request/grant interfaces are provided by the 8088 in maximum mode. These features allow co-processors in local bus and remote bus configurations.



AF003450

Figure 4. Multiplexed Bus Configuration

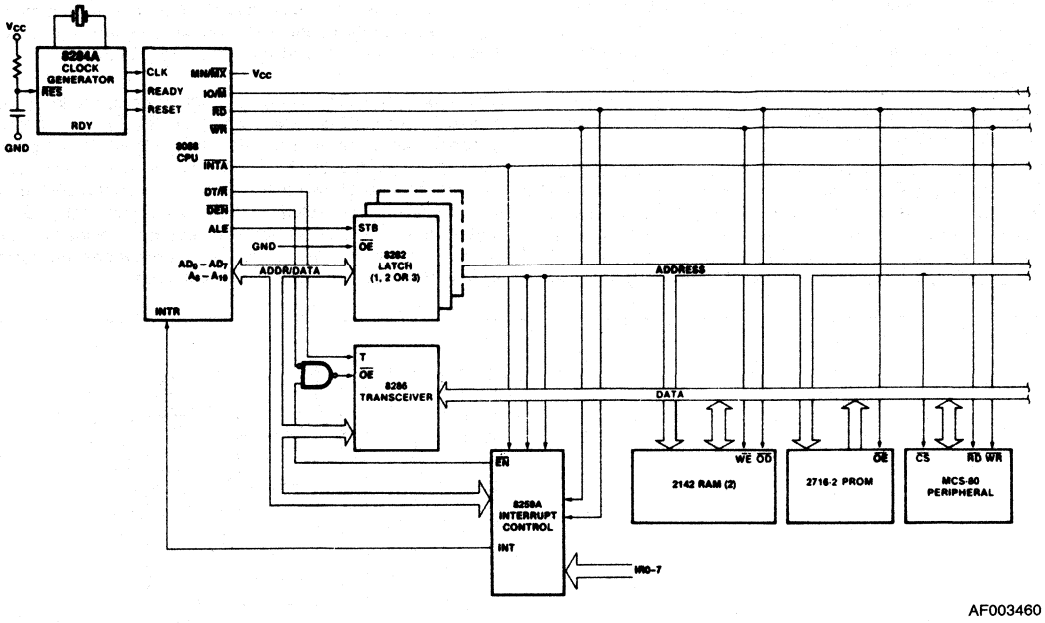


Figure 5. Demultiplexed Bus Configuration

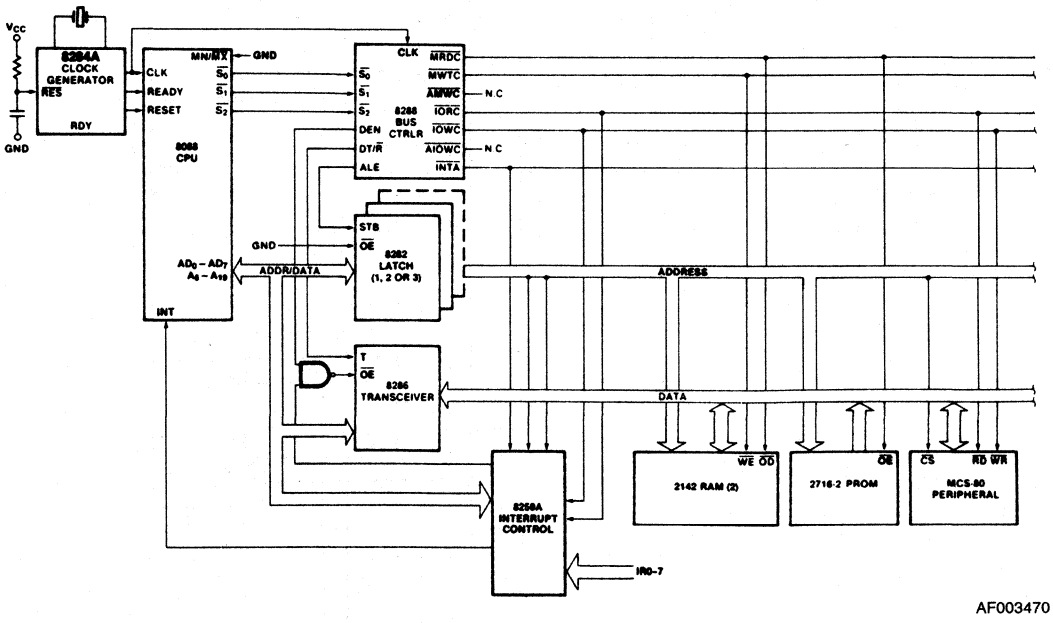


Figure 6. Fully Buffered System Using Bus Controller

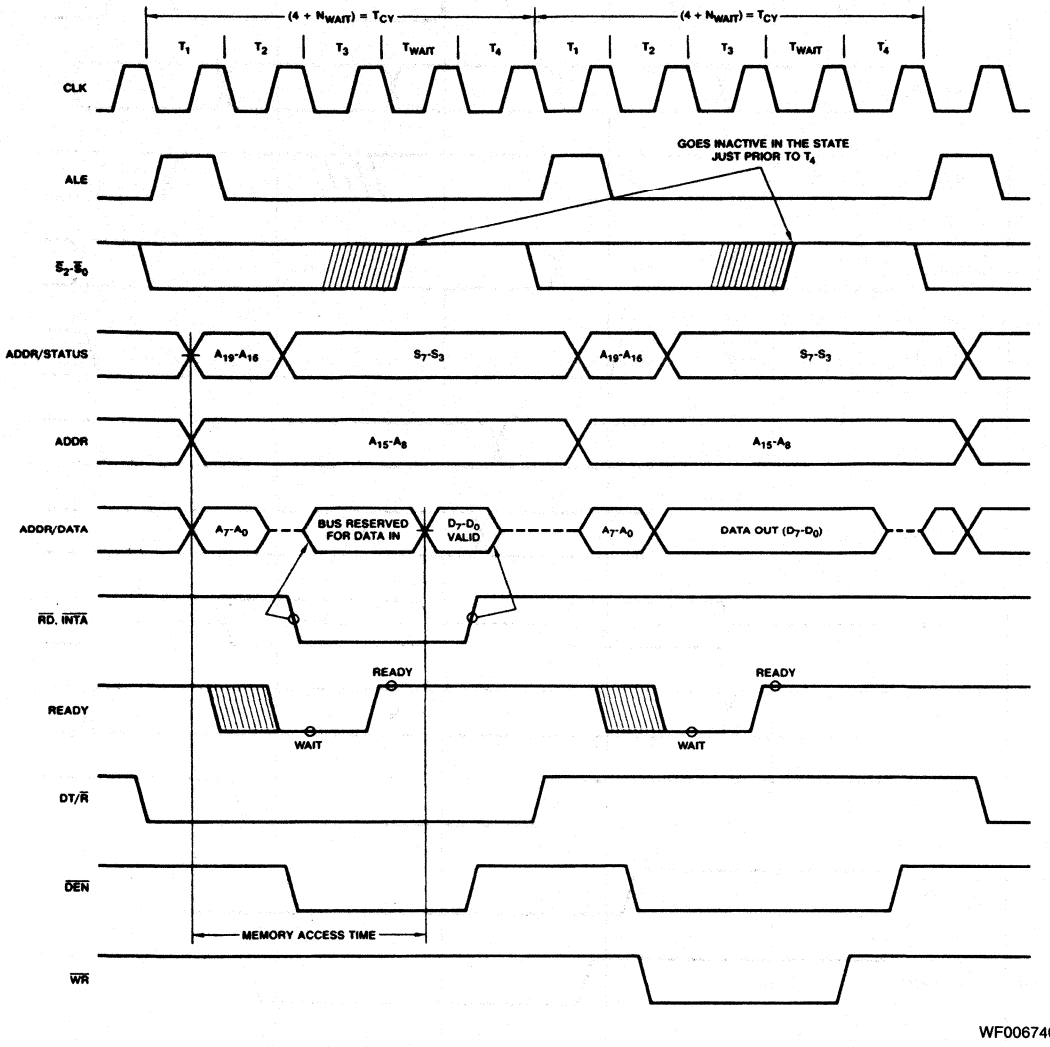


Figure 7. Basic System Timing

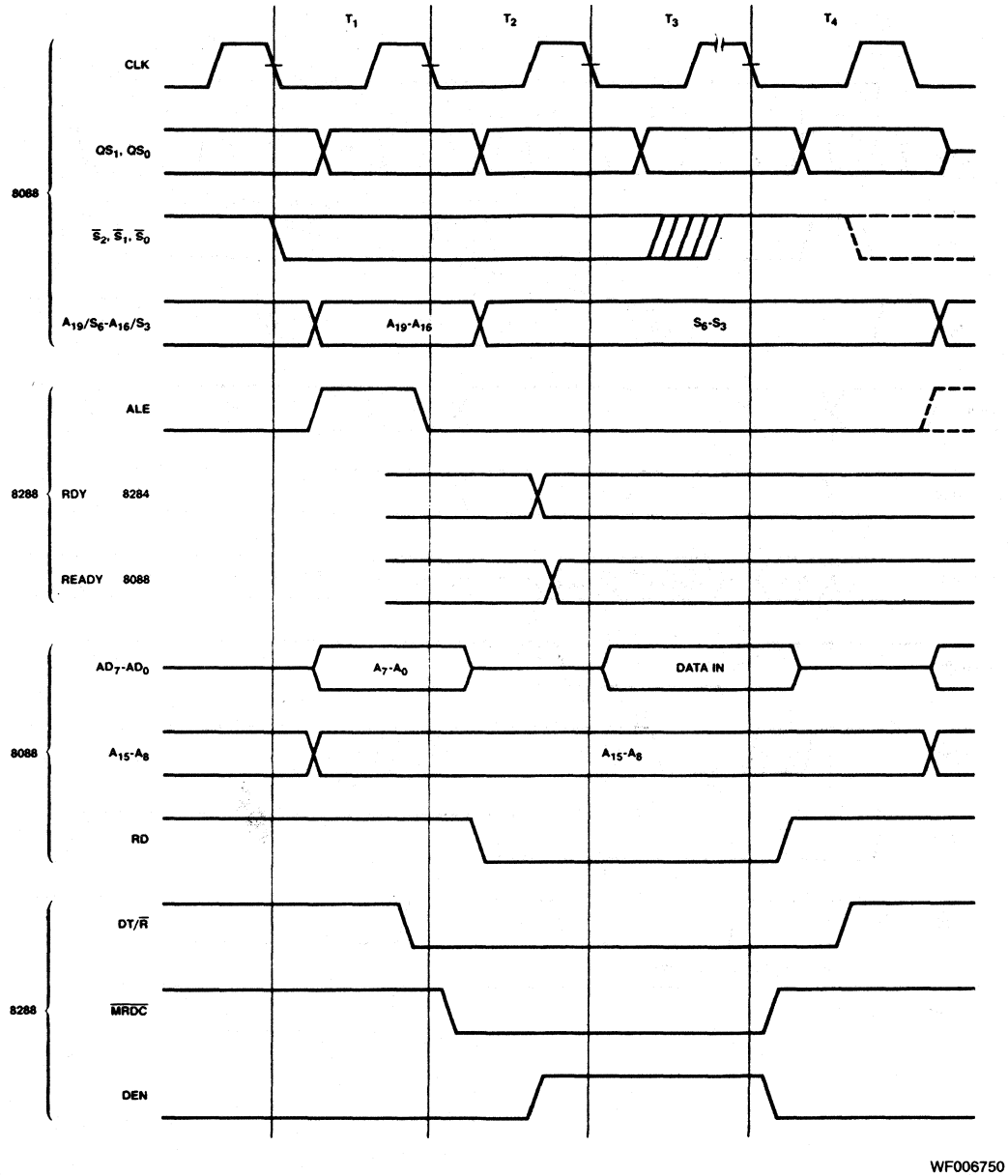


Figure 8. Medium Complexity System Timing

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on any Pin
with Respect to Ground ..... −1.0 to +7.0V
Power Dissipation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8088	0°C to 70°C	5.0V ±10%
8088-2 8088-1	0°C to 70°C	5.0V ±5%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} +0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2.0mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current			340	mA
I_{LI}	Input Leakage Current	$0V \leqslant V_{IN} \leqslant V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$0.45V \leqslant V_{OUT} \leqslant V_{CC}$		±10	μA
V_{CL}	Clock Input Low Voltage		−0.5	+0.6	V
V_{CH}	Clock Input High Voltage		3.9	V_{CC} +1.0	V
C_{IN}	Capacitance of Input Buffer (All input except AD_0–AD_7, RQ/GT)	fc = 1 MHz		15	pF
C_{IO}	Capacitance of I/O Buffer (AD_0–AD_7, RQ/GT)	fc = 1 MHz		15	pF

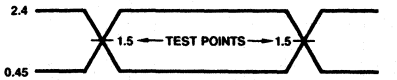
SWITCHING CHARACTERISTICS
MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284 (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284 (See Notes 1, 2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8088		118		68		53		ns
TCHRYX	READY Hold Time into 8088		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 3)		−8		−8		−10		ns
THVCH	HOLD Set-up Time		35		20		20		ns
TINVCH	INTR, NMI, $\overline{TEST}$ Set-up Time (See Note 2)		30		15		15		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

SWITCHING CHARACTERISTICS (Cont.)
TIMING RESPONSES

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLAV	Address Valid Delay	C_L = 100pF for all 8088 Outputs (in addition to internal loads)	10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TLHLL	ALE Width		TCLCH -20		TCLCH -10		TCLCH -10		ns
TCLLH	ALE Active Delay			80		50		40	ns
TCHLL	ALE Inactive Delay			85		55		45	ns
TLLAX	Address Hold Time to ALE Inactive		TCHCL -10		TCHCL -10		TCHCL -10		ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TWHDX	Data Hold Time After $\overline{WR}$		TCLCH -30		TCLCH -30		TCLCH -25		ns
TCVCTV	Control Active Delay 1		10	110	10	70	10	50	ns
TCHCTV	Control Active Delay 2		10	110	10	60	10	45	ns
TCVCTX	Control Inactive Delay		10	110	10	70	10	50	ns
TAZRL	Address Float to READ Active		0		0		0		ns
TCLRL	$\overline{RD}$ Active Delay		10	165	10	100	10	70	ns
TCLRH	$\overline{RD}$ Inactive Delay		10	150	10	80	10	60	ns
TRHAV	$\overline{RD}$ Inactive to Next Address Active		TCLCL -45		TCLCL -40		TCLCL -35		ns
TCLHAV	HLDA Valid Delay		10	160	10	100	10	60	ns
TRLRH	$\overline{RD}$ Width		2TCLCL -75		2TCLCL -50		2TCLCL -40		ns
TWLWH	$\overline{WR}$ Width		2TCLCL -60		2TCLCL -40		2TCLCL -35		ns
TAVAL	Address Valid to ALE Low		TCLCH -60		TCLCH -40		TCLCH -35		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

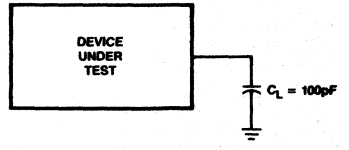
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006760

AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". The clock is driven at 4.3V and 0.25V. Timing measurements are made at 1.5V for both a logic "1" and "0."

SWITCHING TEST LOAD CIRCUIT



WF006770

C_L Includes JIG Capacitance.

SWITCHING CHARACTERISTICS (Cont.)
MAX MODE SYSTEM (USING 8288 BUS CONTROLLER) TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8088		8088-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLCL	CLK Cycle Period		200	500	125	500	100	500	ns
TCLCH	CLK Low Time		118		68		53		ns
TCHCL	CLK High Time		69		44		39		ns
TCH1CH2	CLK Rise Time	From 1.0 to 3.5V		10		10		10	ns
TCL2CL1	CLK Fall Time	From 3.5 to 1.0V		10		10		10	ns
TDVCL	Data in Set-up Time		30		20		5		ns
TCLDX	Data in Hold Time		10		10		10		ns
TR1VCL	RDY Set-up Time into 8284 (See Notes 1, 2)		35		35		35		ns
TCLR1X	RDY Hold Time into 8284 (See Notes 1,2)		0		0		0		ns
TRYHCH	READY Set-up Time into 8088		118		68		53		ns
TCHRYX	READY Hold Time into 8088		30		20		20		ns
TRYLCL	READY Inactive to CLK (See Note 4)		-8		-8		-10		ns
TINVCH	Set-up Time for Recognition (INTR, NMI, TEST) (See Note 2)		30		15		15		ns
TGVCH	RQ/GT Set-up Time		30		15		12		ns
TCHGX	RQ Hold Time into 8086		40		30		20		ns
TILIH	Input Rise Time (Except CLK)	From 0.8 to 2.0V		20		20		20	ns
TIHIL	Input Fall Time (Except CLK)	From 2.0 to 0.8V		12		12		12	ns

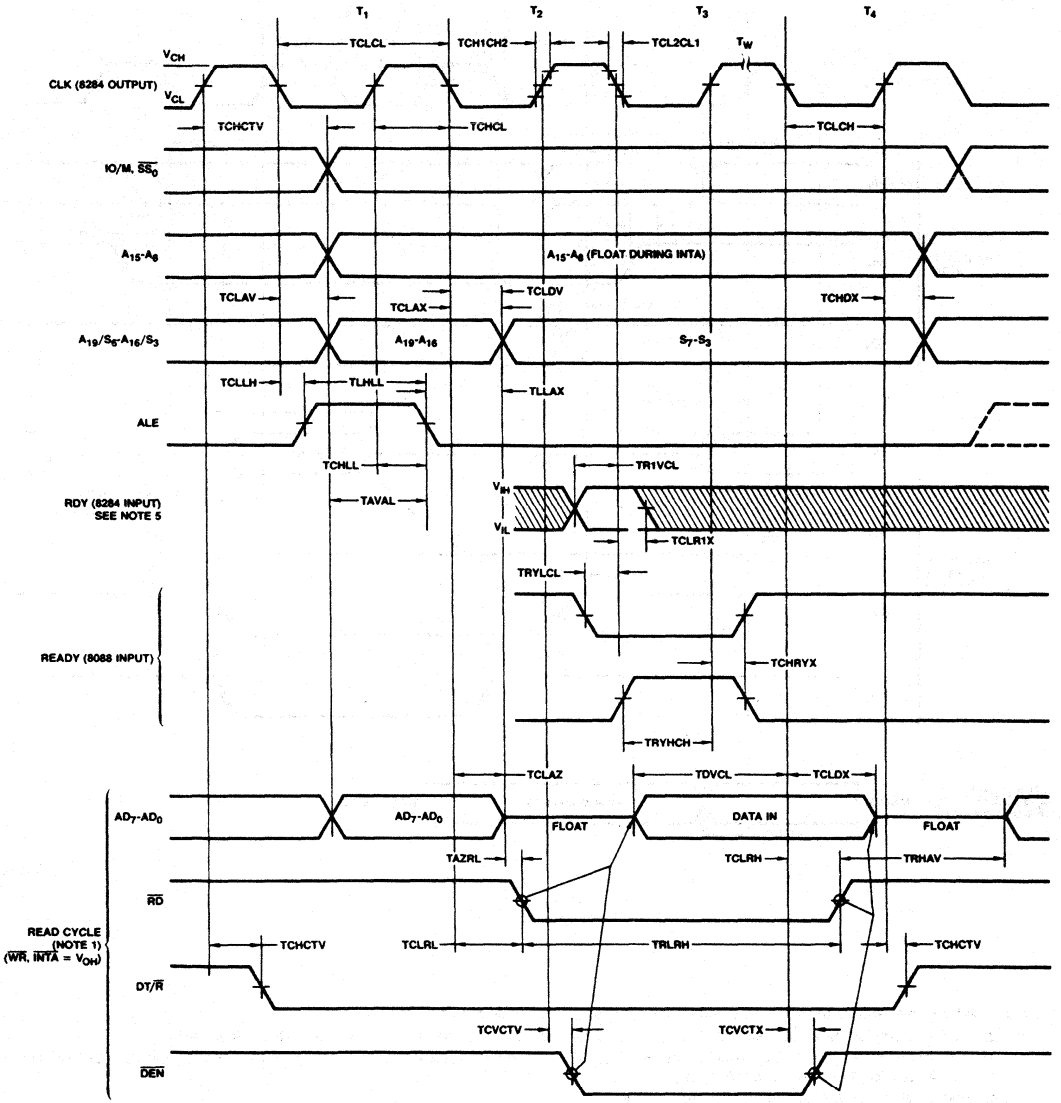
Notes: 1. Signal at 8284 or 8288 shown for reference only.
2. Set-up requirement for asynchronous signal only to guarantee recognition at next CLK.
3. Applies only to T_2 state (8ns into T_3 state).
4. Applies only to T_2 state (8ns into T_3 state).

SWITCHING CHARACTERISTICS (Cont.)
TIMING RESPONSES

Parameters	Description	Test Conditions	8088		8082-2		8088-1		Units
			Min	Max	Min	Max	Min	Max	
TCLML	Command Active Delay (See Note 1)	C_L = 100pF for all 8088 outputs (in addition to internal loads)	10	35	10	35	10	35	ns
TCLMH	Command Inactive Delay (See Note 1)		10	35	10	35	10	35	ns
TRYHSH	READY Active to Status Passive (See Note 3)			110		65		45	ns
TCHSV	Status Active Delay		10	110	10	60	10	45	ns
TCLSH	Status Inactive Delay		10	130	10	70	10	55	ns
TCLAV	Address Valid Delay		10	110	10	60	10	50	ns
TCLAX	Address Hold Time		10		10		10		ns
TCLAZ	Address Float Delay		TCLAX	80	TCLAX	50	10	40	ns
TSVLH	Status Valid to ALE High (See Note 1)			15		15		15	ns
TSVMCH	Status Valid to MCE High (See Note 1)			15		15		15	ns
TCLLH	CLK Low to ALE Valid (See Note 1)			15		15		15	ns
TCLMCH	CLK Low to MCE High (See Note 1)			15		15		15	ns
TCHLL	ALE Inactive Delay (See Note 1)			15		15		15	ns
TCLMCL	MCE Inactive Delay (See Note 1)			15		15		15	ns
TCLDV	Data Valid Delay		10	110	10	60	10	50	ns
TCHDX	Data Hold Time		10		10		10		ns
TCVNV	Control Active Delay (See Note 1)		5	45	5	45	5	45	ns
TCVNX	Control Inactive Delay (See Note 1)		10	45	10	45	10	45	ns
TAZRL	Address Float to Read Active		0		0		0		ns
TCLRL	RD Active Delay		10	165	10	100	10	70	ns
TCLRH	RD Inactive Delay		10	150	10	80	10	60	ns
TRHAV	RD Inactive to Next Address Active		TCLCL −45		TCLCL −40		TCLCL −35		ns
TCHDTL	Direction Control Active Delay (See Note 1)			50		50		50	ns
TCHDTH	Direction Control Inactive Delay (See Note 1)			30		30		30	ns
TCLGL	GT Active Delay			110		50	0	45	ns
TCLGH	GT Inactive Delay			85		50	0	45	ns
TRLRH	RD Width		2TCLCL −75		2TCLCL −50		2TCLCL −40		ns
TOLOH	Output Rise Time	From 0.8 to 2.0V		20		20		20	ns
TOHOL	Output Fall Time	From 2.0 to 0.8V		12		12		12	ns

SWITCHING WAVEFORMS

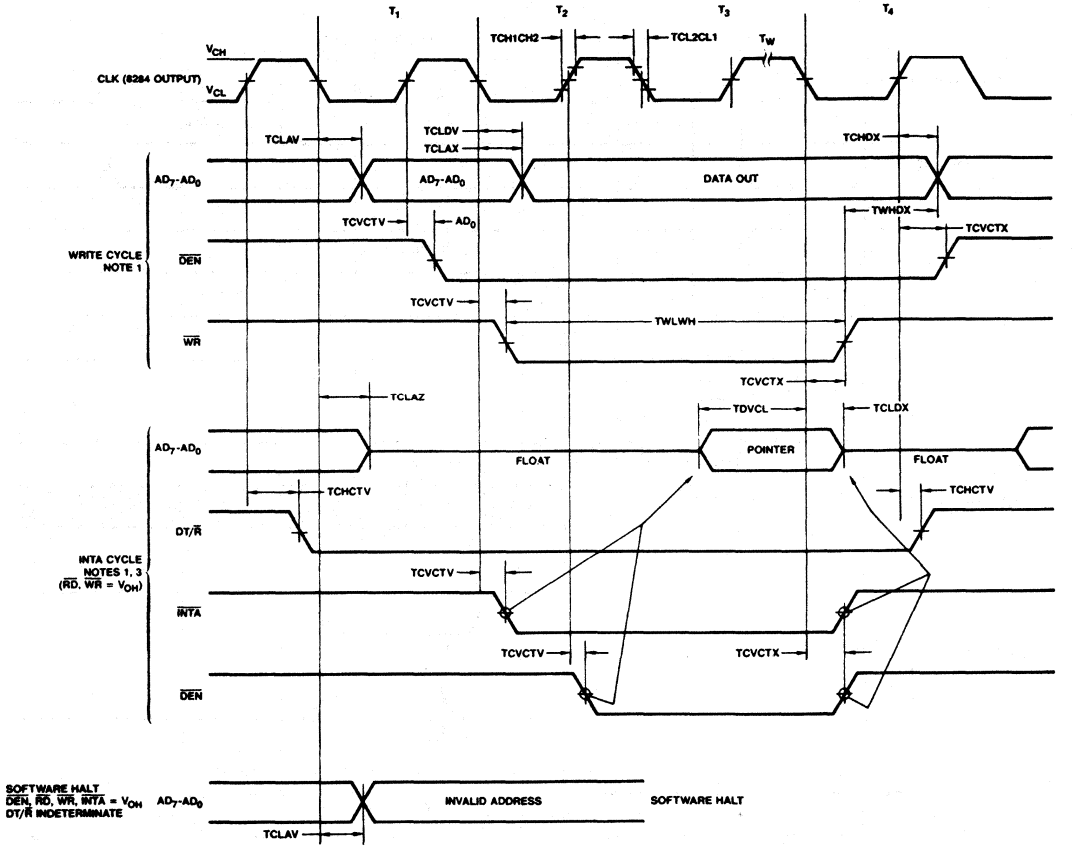
BUS TIMING – MINIMUM MODE SYSTEM



WF006790

SWITCHING WAVEFORMS (Cont.)

BUS TIMING – MINIMUM MODE SYSTEM (Cont.)

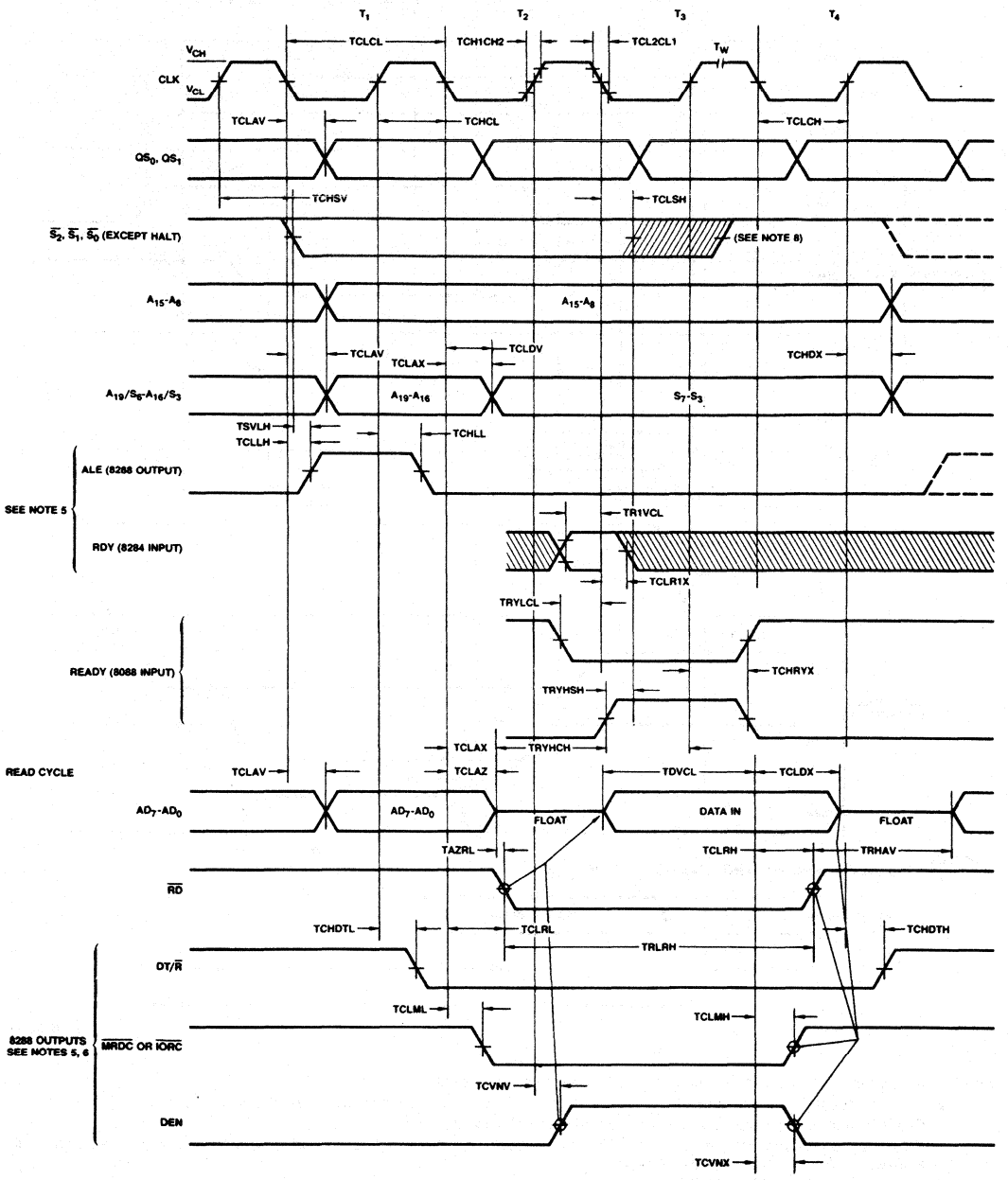


WF006780

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control signals are shown for the second INTA cycle.
4. Signals at 8284 are shown for reference only.
5. All timing measurements are made at 1.5V unless otherwise noted.

SWITCHING WAVEFORMS (Cont.)

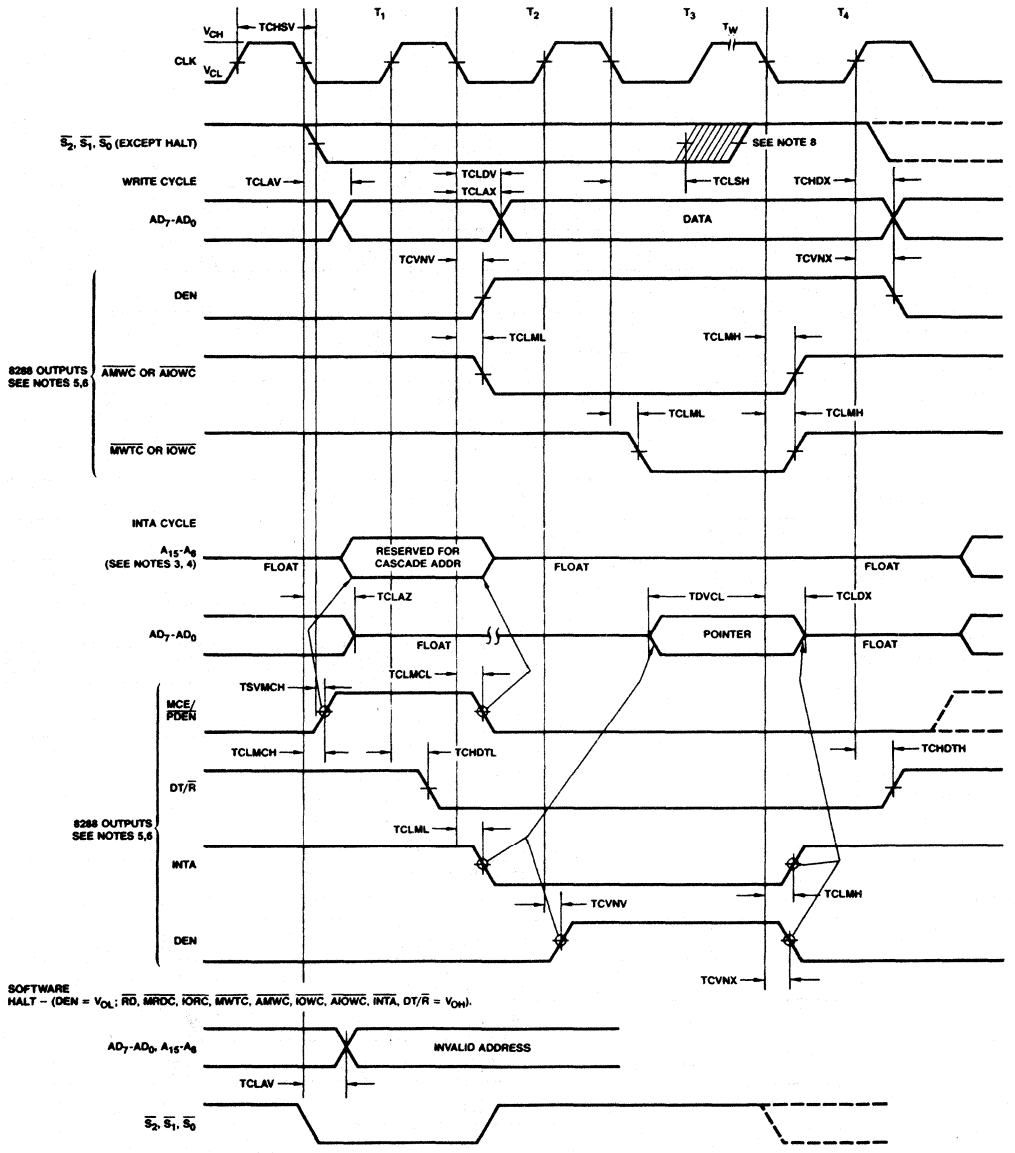
BUS TIMING – MAXIMUM MODE



WF006810

SWITCHING WAVEFORMS (Cont.)

BUS TIMING – MAXIMUM MODE SYSTEM (USING 8288)

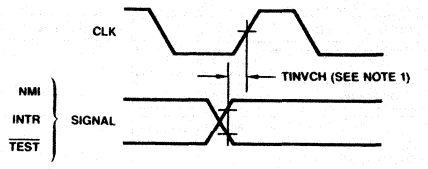


WF006800

Notes: 1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycles.
4. Two INTA cycles run back-to-back. The 8088 local ADDR/DATA bus is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284 or 8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$, and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5V unless otherwise noted.
8. Status inactive in state just prior to T_4.

SWITCHING WAVEFORMS (Cont.)

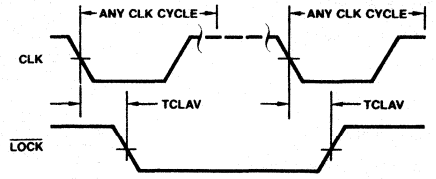
ASYNCHRONOUS SIGNAL RECOGNITION



WF006820

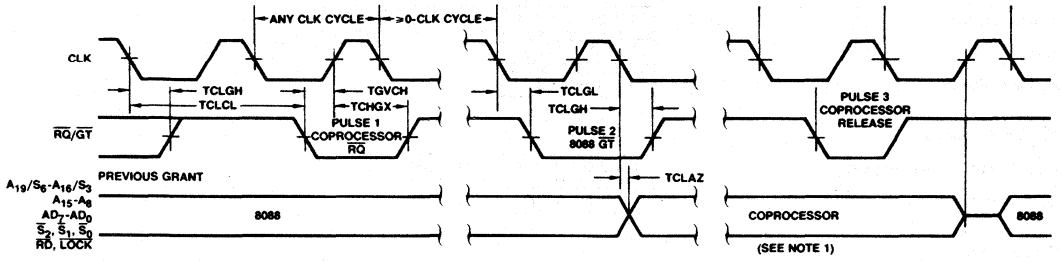
Note 1: Set-up requirements for asynchronous signals only to guarantee recognition at next CLK.

BUS LOCK SIGNAL TIMING (MAXIMUM MODE ONLY)



WF006830

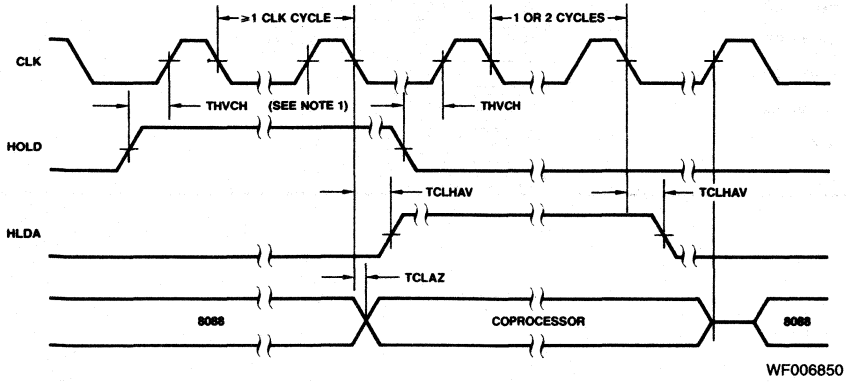
REQUEST/GRANT SEQUENCE TIMING (MAXIMUM MODE ONLY)



WF006840

Note 1: The coprocessor may not drive the buses outside the region shown without rising contention.

HOLD/HOLD ACKNOWLEDGE TIMING (MIMIMUM MODE ONLY)



WF006850

Note 1: All signals switch between V_{OH} and V_{OL} unless otherwise specified.

8086/8088
INSTRUCTION SET SUMMARY

DATA TRANSFER

MOV = Move	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Register/memory to/from register	1 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 1 0 0 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate to register	1 0 1 1 w reg	data	data if w = 1	
Memory to accumulator	1 0 1 0 0 0 0 w	addr-low	addr-high	
Accumulator to memory	1 0 1 0 0 0 1 w	addr-low	addr-high	
Register/memory to segment register	1 0 0 0 1 1 1 0	mod 0 reg r/m		
Segment register to register/memory	1 0 0 0 1 1 0 0	mod 0 reg r/m		
PUSH = Push:				
Register/memory	1 1 1 1 1 1 1 1	mod 1 1 0 r/m		
Register	0 1 0 1 0 reg			
Segment register	0 0 0 reg 1 1 0			
POP = Pop:				
Register/memory	1 0 0 0 1 1 1 1	mod 0 0 0 r/m		
Register	0 1 0 1 1 reg			
Segment register	0 0 0 reg 1 1 1			
XCHG = Exchange:				
Register/memory with register	1 0 0 0 0 1 1 w	mod reg r/m		
Register with accumulator	1 0 0 1 0 reg			
IN = Input from:				
Fixed port	1 1 1 0 0 1 0 w	port		
Variable port	1 1 1 0 1 1 0 w			
OUT = Ouput to:				
Fixed port	1 1 1 0 0 1 1 w	port		
Variable port	1 1 1 0 1 1 1 w			
XLAT = Transtate byte to AL	1 1 0 1 0 1 1 1			
LEA = Load EA to register	1 0 0 0 1 1 0 1	mod reg r/m		
LDS = Load pointer to DS	1 1 0 0 0 1 0 1	mod reg r/m		
LES = Load pointer to ES	1 1 0 0 0 1 0 0	mod reg r/m		
LANF = Load AH with flags	1 0 0 1 1 1 1 1			
SANF = Store AH into flags	1 0 0 1 1 1 1 0			
PUSHF = Push flags	1 0 0 1 1 1 0 0			
POPF = Pop flags	1 0 0 1 1 1 0 1			

INSTRUCTION SET SUMMARY (Cont.)

ARITHMETIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
ADD = Add				
Reg/memory with register to either	0 0 0 0 0 0 d w	mod reg r/m		
Immediate to register / memory	1 0 0 0 0 0 s w	mod 0 0 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 0 0 1 0 w	data	data if w = 1	
ADC = Add with carry:				
Reg/memory with register to either	0 0 0 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 s w	mod 0 1 0 r/m	data	data if s:w = 01
Immediate to accumulator	0 0 0 1 0 1 0 w	data	data if w = 1	
INC = Increment:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 0 r/m		
Register	0 1 0 0 0 reg			
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1			
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1			
SUB = Subtract:				
Reg/memory and register to either	0 0 1 0 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 1 0 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 1 0 1 1 0 w	data	data if w = 1	
SBB = Subtract with borrow:				
Reg/memory and register to either	0 0 0 1 1 0 d w	mod reg r/m		
Immediate from register/memory	1 0 0 0 0 0 s w	mod 0 1 1 r/m	data	data if s:w = 01
Immediate from accumulator	0 0 0 1 1 1 0 w	data	data if w = 1	
DEC = Decrement:				
Register/memory	1 1 1 1 1 1 1 w	mod 0 0 1 r/m		
Register	0 1 0 0 1 reg			
NEG Change sign	1 1 1 1 0 1 1 w	mod 0 1 1 r/m		
CMP = Compare:				
Register/memory with register	0 0 1 1 1 0 1 w	mod reg r/m		
Register with register/memory	0 0 1 1 1 0 0 w	mod reg r/m		
Immediate with register/memory	1 0 0 0 0 0 s w	mod 1 1 1 r/m	data	data if s:w = 01
Immediate with accumulator	0 0 1 1 1 1 0 w	data	data if w = 1	
AAS ASCII adjust for subtract	0 0 1 1 1 1 1 1			
DAS Decimal adjust for subtract	0 0 1 0 1 1 1 1			
MUL Mulitiply (unsigned)	1 1 1 1 0 1 1 w	mod 1 0 0 r/m		
IMUL Integer multiply (signed):	1 1 1 1 0 1 1 w	mod 1 0 1 r/m		
AAM ASCII adjust for multiply	1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0		
DIV Divide (unsigned):	1 1 1 1 0 1 1 w	mod 1 1 0 r/m		
IDIV Integer divide (signed)	1 1 1 1 0 1 1 w	mod 1 1 1 r/m		
AAD ASCH adjust for divide	1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0		
CBW Convert byte to word	1 0 0 1 1 0 0 0			
CWD Convert word to double word	1 0 0 1 1 0 0 1			

INSTRUCTION SET SUMMARY (Cont.)

LOGIC

	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
NOT Invert	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		
SHL/SAL Shift logical/arithmetic left	1 1 0 1 0 0 v w	mod 1 0 0 r/m		
SHR Shift logical right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
SAR Shift arithmetic right	1 1 0 1 0 0 v w	mod 1 1 1 r/m		
ROL Rotate left	1 1 0 1 0 0 v w	mod 0 0 0 r/m		
ROR Rotate right	1 1 0 1 0 0 v w	mod 0 0 1 r/m		
RCL Rotate through carry flag left	1 1 0 1 0 0 v w	mod 0 1 0 r/m		
RCR Rotate through carry right	1 1 0 1 0 0 v w	mod 0 1 1 r/m		
AND = And:				
Reg/memory and register to either	0 0 1 0 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	
TEST = And function to flags, no result:				
Register/memory and register	1 0 0 0 0 1 0 w	mod reg r/m		
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	
OR = Or:				
Reg/memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	
XOR = Exclusive or:				
Reg/memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	

STRING MANIPULATION:

REP = Repeat	1 1 1 1 0 0 1 z
MOVS = Move byte/word	1 0 1 0 0 1 0 w
CMPS = Compare byte/word	1 0 1 0 0 1 1 w
SCAS = Scan byte/word	1 0 1 0 1 1 1 w
LODS = Load byte/wd to AL/AX	1 0 1 0 1 1 0 w
STOS = Stor byte/wd from AL/A	1 0 1 0 1 0 1 w

INSTRUCTION SET SUMMARY (Cont.)

CONTROL TRANSFER

CALL = Call	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low	disp-high	
indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0 r/m		
Direct intersegment	1 0 0 1 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1 r/m		
JMP = Unconditional jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low	disp-high	
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0 r/m		
Direct intersegment	1 1 1 0 1 0 1 0	offset-low	offset-high	
		seg-low	seg-high	
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1 r/m		
RET = Return from CALL:				
Within segment	1 1 0 0 0 0 1 1			
Within seg adding immed to SP	1 1 0 0 0 0 1 0	data-low	data-high	
Intersegment	1 1 0 0 1 0 1 1			
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low	data-high	
JE/JZ = Jump on equal/zero	0 1 1 1 0 1 0 0	disp		
JL/JNGE = Jump on less/not greater or equal	0 1 1 1 1 1 0 0	disp		
JLE/JNG = Jump on less or equal/not greater	0 1 1 1 1 1 1 0	disp		
JB/JNAE = Jump on below/not above or equal	0 1 1 1 0 0 1 0	disp		
JBE/JNA = Jump on below or equal/not above	0 1 1 1 0 1 1 0	disp		
JP/JPE = Jump on parity/parity even	0 1 1 1 1 0 1 0	disp		
JO = Jump on overflow	0 1 1 1 0 0 0 0	disp		
JS = Jump on sign	0 1 1 1 1 0 0 0	disp		
JNE/JNZ = Jump on not equal/not zero	0 1 1 1 0 1 0 1	disp		
JNL/JGE = Jump on not less/greater or equal	0 1 1 1 1 1 0 1	disp		
JNLE/JG = Jump on not less or equal/greater	0 1 1 1 1 1 1 1	disp		
JNB/JAE = Jump on not below/above or equal	0 1 1 1 0 0 1 1	disp		
JNBE/JA = Jump on not below or equal/above	0 1 1 1 0 1 1 1	disp		
JNP/JPO = Jump on not par/par odd	0 1 1 1 1 0 1 1	d isp		
JNO = Jump on not overflow	0 1 1 1 0 0 0 1	disp		
JNS = Jump on not sign	0 1 1 1 1 0 0 1	disp		
LOOP = Loop CX times	1 1 1 0 0 0 1 0	disp		
LOOPZ/LOOPE = Loop while zero/equal	1 1 1 0 0 0 0 1	disp		
LOOPNZ/LOOPNE = Loop while not zero/equal	1 1 1 0 0 0 0 0	disp		
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1	disp		

INSTRUCTION SET SUMMARY (Cont.)

CONTROL TRANSFER (Cont.)

INT = Interrupt	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Type specified	1 1 0 0 1 1 0 1	type		
Type 3	1 1 0 0 1 1 0 0			
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0			
IRET = Interrupt return	1 1 0 0 1 1 1 1			

PROCESSOR CONTROL

CLC = Clear carry	1 1 1 1 1 0 0 0	
CMC = Complement carry	1 1 1 1 0 1 0 1	
STC = Set carry	1 1 1 1 1 0 0 1	
CLD = Clear direction	1 1 1 1 1 1 0 0	
STD = Set direction	1 1 1 1 1 1 0 1	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	
STI = Set interrupt	1 1 1 1 1 0 1 1	
HLT = Halt	1 1 1 1 0 1 0 0	
WAIT = Wait	1 0 0 1 1 0 1 1	
ESC = Processor Extension Escape	1 0 0 1 1 x x x	mod x x x r/m
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	

Footnotes:

AL = 8-bit accumulator
AX = 16-bit accumulator
CX = Count register
DS = Data segment
ES = Extra segment
Above/below refers to unsigned value.
Greater = more positive.
Less = less positive (more negative) signed values
if d = 1 then "to" reg; if d = 0 then "from" reg
w = 1 then word instruction; if w = 0 then byte instruction

if mod = 11 then r/m is treated as a REG field
if mod = 00 then DISP = 0*, disp-low and disp-high are absent
if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent
if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP
if r/m = 001 then EA = (BX) + (DI) + DISP
if r/m = 010 then EA = (BP) + (SI) + DISP
if r/m = 011 then EA = (BP) + (DI) + DISP
if r/m = 100 then EA = (SI) + DISP
if r/m = 101 then EA = (DI) + DISP
if r/m = 110 then EA = (BP) + DISP*
if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

if s:w = 01 then 16 bits of immediate data form the operand.
if s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand.
if v = 0 then "count" = 1; if v = 1 then "count" in (CL)
x = don't care
z is used for string primitives for comparison with Z.F Flag.

SEGMENT OVERRIDE PREFIX

0	0	1	reg	1	1	0

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register files as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = X:X:X:X:(OF):(DF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)

82284

Clock Driver and Ready Interface
for iAPX 286 Processors

PRELIMINARY

DISTINCTIVE CHARACTERISTICS

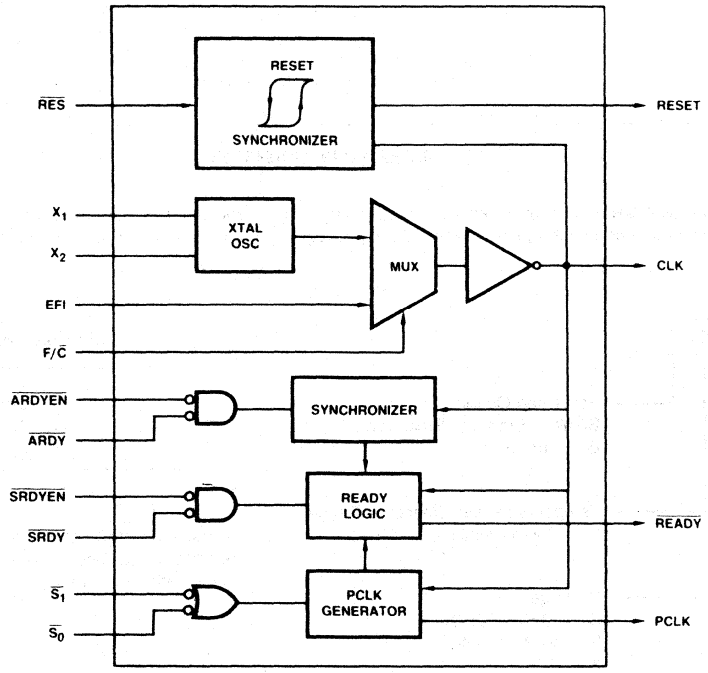
- Generates system clock for iAPX 286 processors
- Uses crystal or TTL signal for frequency source
- Provides local $\overline{\text{READY}}$ and Multibus* $\overline{\text{READY}}$ synchronization
- Generates system reset output from Schmitt Trigger input
- 18-pin package
- Single +5V power supply

GENERAL DESCRIPTION

The 82284 is a clock generator/driver which provides clock signals for iAPX 286 processors and support components.

The device contains logic to supply $\overline{\text{READY}}$ to the CPU from either asynchronous or synchronous sources. It also generates a synchronous reset signal from an asynchronous input with hysteresis.

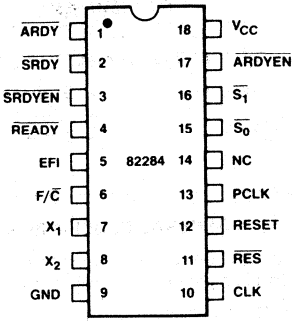
Figure 1. 82284 Block Diagram



05917A-1

*Multibus is a registered trademark of Intel Corporation.

Figure 2. 82284 Pin Configuration



05917A-2

Note: Pin 1 is marked for orientation.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: device number, speed option (if applicable), package type, operating range and screening option (if desired).

82284 D C B

- Screening Option
  - Blank – Standard Processing
  - B – Burn-In
- Temperature (see Operating Range)
  - C – Commercial (0 to +70°C)
- Package
  - D = 18-Pin CERDIP
  - P = 18-Pin Plastic DIP
  - X = Dice
- Clock Driver and Ready Interface for iAPX 286 Processors

Valid Combinations

D82284
D82284B
P82284
P82284B

Consult the AMD sales office in your area to determine if a device is currently available in the combination you wish.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	$\overline{ARDY}$	I	**Asynchronous Ready** $\overline{ARDY}$ is an active LOW input used to terminate the current bus cycle. The $\overline{ARDY}$ input is qualified by $\overline{ARDYEN}$. Inputs to $\overline{ARDY}$ may be applied asynchronously to CLK. Setup and hold times are given to assure a guaranteed response to synchronous inputs.
2	$\overline{SRDY}$	I	**Synchronous Ready** $\overline{SRDY}$ is an active LOW input used to terminate the current bus cycle. The $\overline{SRDY}$ input is qualified by the $\overline{SRDYEN}$ input. Setup and hold times must be satisfied for proper operation.
3	$\overline{SRDYEN}$	I	**Synchronous Ready Enable** $\overline{SRDYEN}$ is an active LOW input which qualifies $\overline{SRDY}$. $\overline{SRDYEN}$ selects $\overline{SRDY}$ as the source for $\overline{READY}$ to the CPU for the current bus cycle. Setup and hold times must be satisfied for proper operation.
4	$\overline{READY}$	O	**Ready** $\overline{READY}$ is an active LOW output which signals the current bus cycle is to be completed. The $\overline{SRDY}$, $\overline{SRDYEN}$, $\overline{ARDY}$, $\overline{ARDYEN}$, $\overline{S_1}$, $\overline{S_0}$ and $\overline{RES}$ inputs control $\overline{READY}$ as explained later in the $\overline{READY}$ generator section. $\overline{READY}$ is an open collector output requiring an external 910 ohm pull-up resistor
5	EFI	I	**External Frequency In** The EFI input drives CLK when F/$\overline{C}$ is strapped HIGH. The EFI input frequency must be twice the processor's internal clock frequency.
6	F/$\overline{C}$	I	**Frequency/Crystal Select** F/$\overline{C}$ is a strapping option used to select the source for the CLK output. When F/$\overline{C}$ is strapped LOW, the internal crystal drives CLK. When F/$\overline{C}$ is strapped HIGH, the EFI input drives the CLK output.
7,8	X_1,X_2	I	**Crystal In** These are the pins to which a parallel resonant fundamental mode crystal is attached for the internal oscillator. When F/$\overline{C}$ is strapped LOW, the oscillator will drive the CLK output at the crystal frequency. The crystal frequency must be twice the processor's internal clock frequency.
9	GND		System Ground: 0V
10	CLK	O	**System Clock** CLK output is used by the processor and any support devices which must be synchronized with the processor. The frequency of the CLK output is twice the processor's internal clock frequency. CLK can drive both TTL and MOS level inputs.
11	$\overline{RES}$	I	**Reset In** $\overline{RES}$ is an active LOW input which generates the system reset signal RESET. Signals to $\overline{RES}$ may be applied asynchronously to CLK. A Schmitt Trigger input is provided on $\overline{RES}$, so that an RC circuit can be used to provide a time delay. Setup and hold times are given to assure a guaranteed response to synchronous inputs.
12	RESET	O	**Reset** RESET is an active HIGH output which is derived from the $\overline{RES}$ input. RESET is used to force the system into an initial state. When RESET is active, $\overline{READY}$ will be active (LOW).
13	PCLK	O	**Peripheral Clock** PCLK is an output which provides a 50% duty cycle clock with one half the frequency of CLK. PCLK will be in phase with the processor's internal clock following the first bus cycle after the processor has been reset.
14			No Connection
15,16	$\overline{S_0}$,$\overline{S_1}$	I	**Status** These inputs prepare the 82284 for a subsequent bus cycle. $\overline{S_0}$ and $\overline{S_1}$ synchronize PCLK to the internal processor clock and control $\overline{READY}$. These inputs have pullup resistors to keep them HIGH if nothing is driving them. Setup and hold times must be satisfied for proper operation.
17	$\overline{ARDYEN}$	I	**Asynchronous Ready Enable** $\overline{ARDYEN}$ is an active LOW input which qualifies the $\overline{ARDY}$ input. $\overline{ARDYEN}$ selects $\overline{ARDY}$ as the source of $\overline{READY}$ for the current bus cycle. Inputs to $\overline{ARDYEN}$ may be applied asynchronously to CLK. Setup and hold times are given to assure a guaranteed response to synchronous inputs.
18	V_{CC}	I	Supply Power: +5V

3

INTRODUCTION

The 82284 generates the clock, ready, and reset signals required for iAPX 286 processors and support components. The 82284 is packaged in an 18-pin DIP and contains a crystal controlled oscillator, MOS clock generator, peripheral clock generator, Multibus ready synchronization logic and system reset generation logic.

CLOCK GENERATOR

The CLK output provides the basic timing control for an iAPX 286 system. CLK has output characteristics sufficient to drive MOS devices. CLK is generated by either an internal crystal oscillator or an external source as selected by the F/$\overline{C}$ strapping option. When F/$\overline{C}$ is LOW, the crystal oscillator drives the CLK output. When F/$\overline{C}$ is HIGH, the EFI input drives the CLK output.

The 82284 provides a second clock output (PCLK) for peripheral devices. PCLK is CLK divided by two. PCLK has a duty cycle of 50% and TTL output drive characterisitcs. PCLK is normally synchronized to the internal processor clock.

After reset, the PCLK signal may be out of phase with the internal processor clock. The $\overline{S}_1$ and $\overline{S}_0$ signals of the first bus cycle are used to synchronize PCLK to the internal processor clock. The phase of the PCLK output changes by extending its HIGH time beyond one system clock (see waveforms). PCLK is forced HIGH when either $\overline{S}_0$ or $\overline{S}_1$ was active (LOW) for the two previous CLK cycles. PCLK continues to oscillate when both S_0 and S_1 are HIGH.

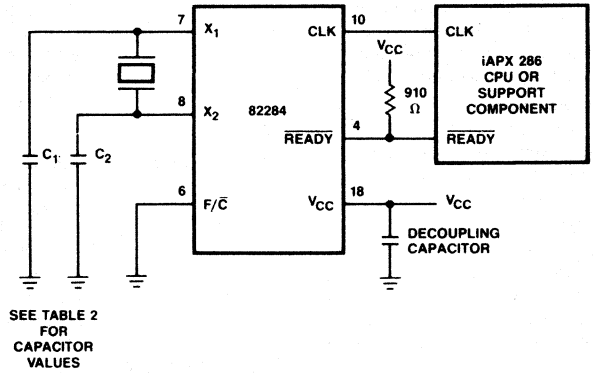
Since the phase of the internal processor clock will not change except during reset, the phase of PCLK will not change except during the first bus cycle after reset.

OSCILLATOR

The oscillator circuit of the 82284 is a linear Pierce oscillator which requires an external parallel resonant fundamental mode crystal. The output of the oscillator is internally buffered. The crystal frequency chosen should be twice the processor's internal clock frequency. The crystal should have a typical load capacitance of 32pF.

X_1 and X_2 are the oscillator crystal connections. For stable operation of the oscillator, two loading capacitors are recommended, as shown in Figure 3. The sum of the board capacitance and loading capacitance should equal the values shown. It is advisable to limit stray board capacitances (not including the effect of the loading capacitors or crystal capacitance) to less than 10pF between the X_1 and X_2 pins. V_{CC} and GND pins should be decoupled as close to the 82284 as possible.

Figure 3. Recommended Crystal and READY Connections



05917A-3

Table 2. 82284 Crystal Loading Capacitance Values

Crystal Frequency	C_1 Capacitance (pin 7)	C_2 Capacitance (pin 8)
1 to 8MHz	60pF	40pF
8 to 16MHz	25pF	15pF

Note: Capacitance values must include stray board capacitance.

RESET OPERATION

The reset logic provides the RESET output to force the system into a known, initial state. When the $\overline{RES}$ input is active (LOW), the RESET output becomes active (HIGH). $\overline{RES}$ is synchronized internally at the falling edge of CLK before generating the RESET output (see waveforms). Synchronization of the $\overline{RES}$ input introduces a one or two CLK delay before affecting the RESET output.

At power up, a system does not have a stable V_{CC} and CLK. To prevent spurious activity, $\overline{RES}$ should be asserted until V_{CC} and CLK stabilize at their operating values. iAPX 286 processors and support components also require their RESET inputs be HIGH a minimum of 16 CLK cycles. An RC network, as shown in Figure 4, will keep $\overline{RES}$ LOW long enough to satisfy both needs.

A Schmitt Trigger input with hysteresis on $\overline{RES}$ assures a single transition of RESET with an RC circuit on $\overline{RES}$. The hysteresis separates the input voltage level at which the circuit output switches from HIGH to LOW from the input voltage level at which the circuit output switches from LOW to HIGH. The $\overline{RES}$ HIGH to LOW input transition voltage is lower than the $\overline{RES}$ LOW to HIGH input transition voltage. As long as the slope of the $\overline{RES}$ input voltage remains in the same direction (increasing or decreasing) around the $\overline{RES}$ input transition voltage, the RESET output will make a single transition.

READY OPERATION

The 82284 accepts two ready sources for the system ready signal which terminates the current bus cycle. Either a synchronous ($\overline{SRDY}$) or asynchronous ready ($\overline{ARDY}$) source may be used. Each ready input has an enable ($\overline{SRDYEN}$ and $\overline{ARDYEN}$) for selecting the type of ready source required to terminate the current bus cycle. An address decoder would normally select one of the enable inputs.

$\overline{READY}$ is enabled (LOW) if either $\overline{SRDY}$ + $\overline{SRDYEN}$ = 0 or $\overline{ARDY}$ + $\overline{ARDYEN}$ = 0 when sampled by the 82284 $\overline{READY}$ generation logic. $\overline{READY}$ will remain active for at least two CLK cycles.

The $\overline{READY}$ output has an open-collector driver allowing other ready circuits to be wire OR'ed with it, as shown in Figure 3. The $\overline{READY}$ signal of an iAPX 286 system requires an external 910 ohm ±5% pull-up resistor. To force the $\overline{READY}$ signal inactive (HIGH) at the start of a bus cycle, the $\overline{READY}$ output floats when either $\overline{S}_1$ or $\overline{S}_0$ are sampled LOW at the falling edge of CLK. Two system clock periods are allowed for the pull-up resistor to pull the $\overline{READY}$ signal to V_{IH}. When RESET is active, $\overline{READY}$ is forced active one CLK later (see waveforms).

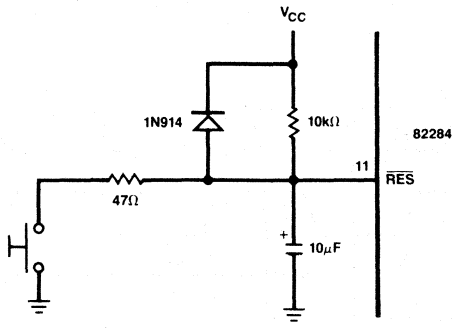
Figure 5 illustrates the operation of $\overline{SRDY}$ and $\overline{SRDYEN}$. These inputs are sampled on the falling edge of CLK when $\overline{S}_1$ and $\overline{S}_0$ are inactive and PCLK is HIGH. $\overline{READY}$ is forced active when both $\overline{SRDY}$ and $\overline{SRDYEN}$ are sampled as LOW.

Figure 6 shows the operation of $\overline{ARDY}$ and $\overline{ARDYEN}$. These inputs are sampled by an internal synchronizer at each falling edge of CLK. The output of the synchronizer is then sampled when PCLK is HIGH. If the synchronizer resolved both the $\overline{ARDY}$ and $\overline{ARDYEN}$ inputs to have been LOW, $\overline{READY}$ becomes LOW. When both $\overline{ARDY}$ and $\overline{ARDYEN}$ have been resolved as active, the $\overline{SRDY}$ and $\overline{SRDYEN}$ inputs are ignored.

$\overline{READY}$ remains active until either $\overline{S}_1$ or $\overline{S}_0$ are sampled LOW, or the ready inputs are sampled as inactive.

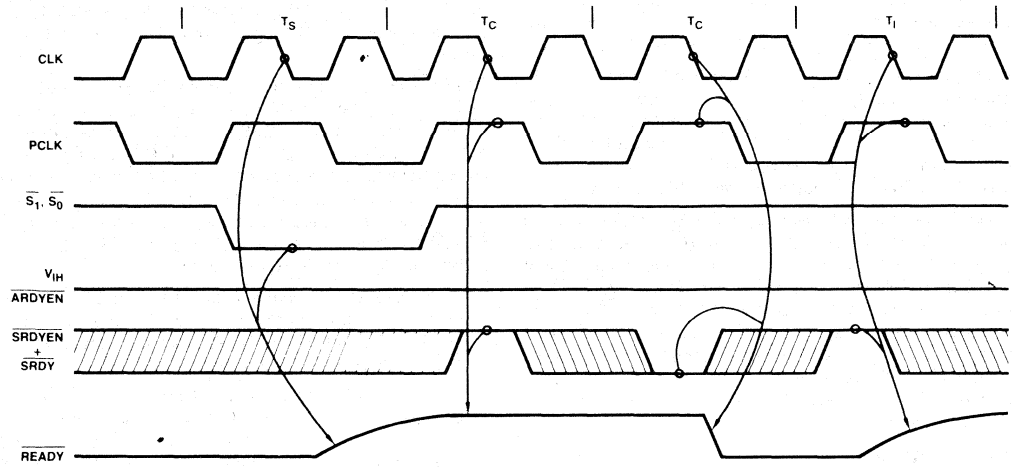
3

Figure 4. Typical RC $\overline{RES}$ Timing Circuit



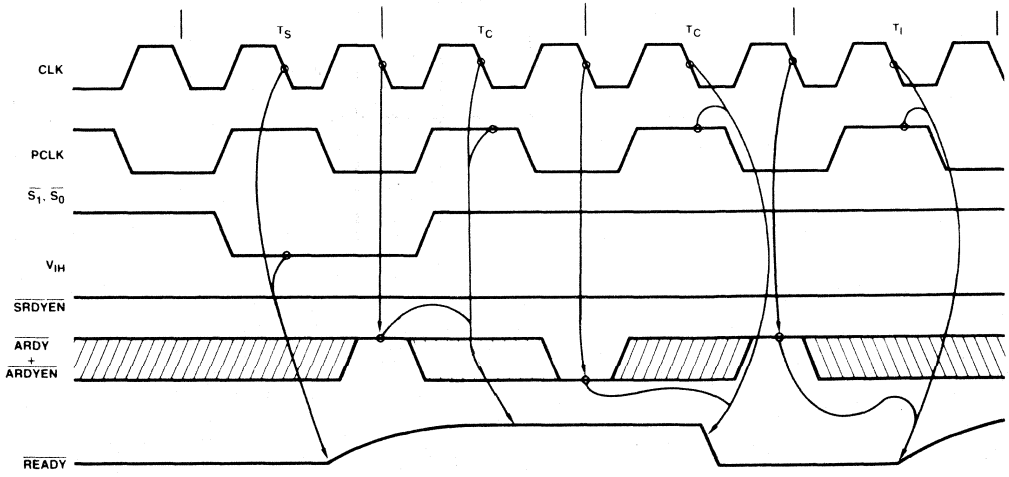
05917A-4

Figure 5. Synchronous Ready Operation



05917A-5

Figure 6. Asynchronous Ready Operation



05917A-6

ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias ..... 0 to +70°C
Storage Temperature ..... −65 to +150°C
All Output and Supply Voltages ..... −0.5V to +7V
All Input Voltages ..... −1.0V to +5.5V
Power Dissipation 1W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGE

Commercial (C) Devices
Ambient Temperature ..... 0 to +70°C
Supply Voltage ..... +4.5 to +5.5V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS OVER OPERATING RANGE unless otherwise specified

		6MHz		8MHz			
Symbol	Parameter	−6 Min	−6 Max	Min	Max	Unit	Test Condition
V_{IL}	Input LOW Voltage		.8		.8	V	
V_{IH}	Input HIGH Voltage	2.0		2.0		V	
V_{IHR}	$\overline{RES}$ and EFI Input HIGH Voltage	2.6		2.6		V	
V_{HYS}	$\overline{RES}$ Input Hysteresis	0.25		0.25		V	
V_{OL}	RESET, PCLK Output LOW Voltage		.45		.45	V	I_{OL} = 5mA
V_{OH}	RESET, PCLK Output HIGH Voltage	2.4		2.4		V	I_{OH} = −1mA
V_{OLR}	$\overline{READY}$, Output LOW Voltage		.45		.45	V	I_{OL} = 7mA
V_{OLC}	CLK Output LOW Voltage		.45		.45	V	I_{OL} = 5mA
V_{OHC}	CLK Output HIGH Voltage	4.0		4.0		V	I_{OH} = −800μA
V_C	Input Forward Clamp Voltage		−1.0		−1.0	V	I_C = −5mA
I_F	Forward Input Current		−.5		−.5	mA	V_F = .45V
I_R	Reverse Input Current		50		50	μA	V_R = V_{CC} Max
I_{CC}	Power Supply Current		145		145	mA	
C_I	Input Capacitance		10		10	pF	F_C = 1MHz

AC CHARACTERISICS (T_A = 0 to 70 C, V_{CC} = 5V, ±10%)

AC timings are referenced to 0.8V and 2.0V points of signals as illustrated in data sheet waveforms, unless otherwise noted.

		6MHz		8MHz			
Symbol	Parameter	-6 Min	-6 Max	Min	Max	Unit	Test Condition
1	EFI to CLK Delay		35		30	ns	At 1.5V Note 1
2	EFI LOW Time	40		25		ns	At 1.5V Notes 1 and 7
3	EFI HIGH Time	35		25		ns	At 1.5V Notes 1 and 7
4	CLK Period	83	500	62	500	ns	
5	CLK LOW Time	20		15		ns	At 1.0V Notes 1, 2 and 8
6	CLK HIGH Time	25		25		ns	At 3.6V Notes 1, 2 and 8
7	CLK Rise Time		10		10	ns	1.0 to 3.6V Note 1
8	CLK Fall Time		10		10	ns	3.6 to 1.0V Note 1
9	Status Setup Time	28		22		ns	Note 1
10	Status Hold Time	1		1		ns	Note 1
11	$\overline{SRDY}$ or $\overline{SRDYEN}$ Setup Time	25		15		ns	Note 1
12	$\overline{SRDY}$ or $\overline{SRDYEN}$ Hold Time	0		0		ns	Note 1
13	$\overline{ARDY}$ or $\overline{ARDYEN}$ Setup Time	5		0		ns	Notes 1 and 3
14	$\overline{ARDY}$ or $\overline{ARDYEN}$ Hold Time	30		30		ns	Notes 1 and 3
15	$\overline{RES}$ Setup Time	25		20		ns	Notes 1 and 3
16	$\overline{RES}$ Hold Time	10		10		ns	Notes 1 and 3
17	$\overline{READY}$ Inactive Delay	5		5		ns	At 0.8V Note 4
18	$\overline{READY}$ Active Delay	0	33	0	24	ns	At 0.8V Note 4
19	PCLK Delay	0	45	0	45	ns	Note 5
20	RESET Delay	5	50	5	34	ns	Note 5
21	PCLK LOW Time	t_4 - 20		t_4 - 20		ns	Notes 5 and 6
22	PCLK HIGH Time	t_4 - 20		t_4 - 20		ns	Notes 5 and 6

Notes:
1. CLK loading: C_L = 150pF.
2. With the internal crystal oscillator using recommended crystal and capacitive loading, or with the EFI input meeting specifications t_2 and t_3. Use a parallel-resonant, fundamental mode crystal. The recommended crystal loading for CLK frequencies of 8 16MHz are 25pF from pin X_1 to ground, and 15pF from pin X_2 to ground. These recommended values are ± 5pF and include all stray capacitance. Decouple V_{CC} and GND as close to the 82284 as possible.
3. This is an asynchronous input. This specification is given for testing purposes only, to assure recognition at specific CLK edge.
4. $\overline{READY}$ loading: I_{OL} = 7mA, C_L = 150pF. In system application, use 910 ohm ± 5% pull-up resistor to meet 80286, 80286-6 and 80286-4 timing requirements.
5. PCLK and RESET loading: C_L = 75pF. PCLK also has 750 ohm pull-up.
6. t_4 refers to any allowable CLK period.
7. When driving the 82284 with EFI, provide minimum EFI HIGH and LOW times as follows:

CLK Output Frequency	8MHz CLK	12MHz CLK	16MHz CLK*
Min Required EFI HIGH Time	52ns	35ns	25ns
Min Required EFI LOW Time	52ns	40ns	25ns

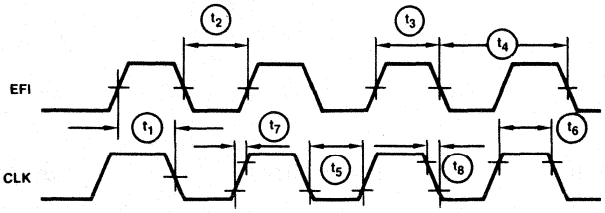
*At CLK frequencies above 12MHz, CLK output HIGH and LOW times are guaranteed only when using a crystal with recommended capacitive loading per Table 2, not when driving component from EFI. All features of the 82284 remain functional whether EFI or a crystal is used to drive the 82284.

8. When using a crystal (with recommended loading capacitance per Table 2) appropriate for the speed of the 80286, CLK output HIGH and LOW times are guaranteed to meet 80286 requirements.

0591
Refer to page 7-1 for Essential Information on Military Devi

WAVEFORMS

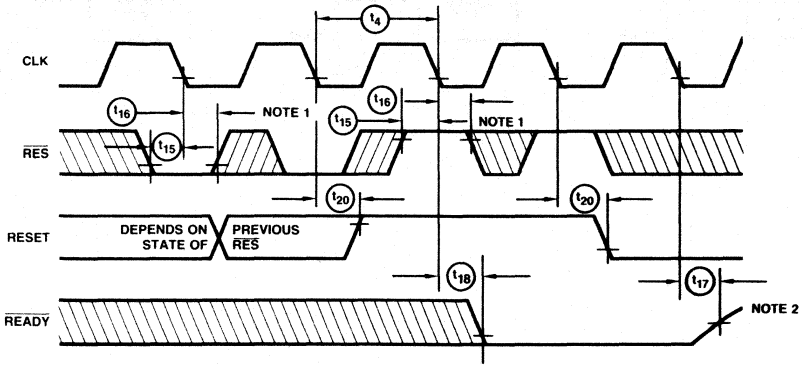
CLK AS A FUNCTION OF EFI



05917A-9

Note: The EFI input LOW and HIGH times as shown are required to guarantee the CLK LOW and HIGH times shown.

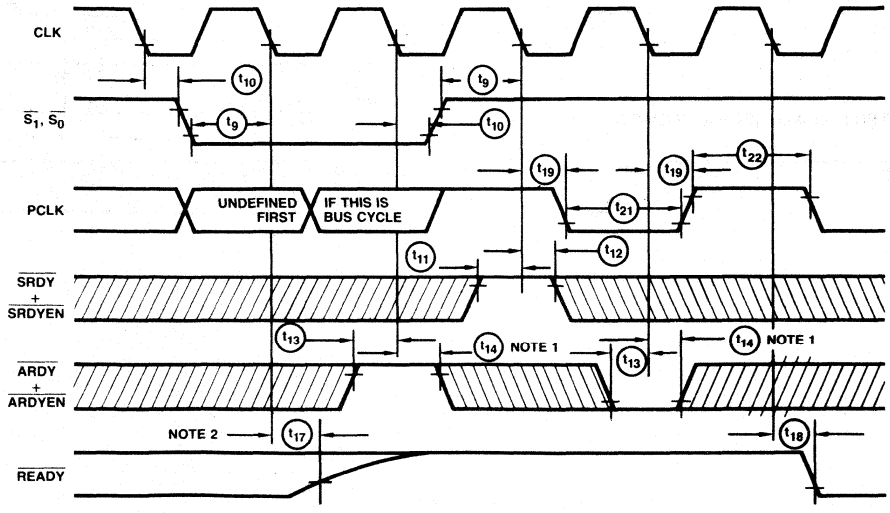
RESET AND $\overline{\text{READY}}$ TIMING AS A FUNCTION OF $\overline{\text{RES}}$ WITH $\overline{S_1}$ AND $\overline{S_0}$ HIGH



05917A-10

Notes: 1. This is an asynchronous input. The setup and hold times shown are required to guarantee the response shown.
2. Tie 910 ohm ±5% pull-up resistor to the $\overline{\text{READY}}$ output.

$\overline{\text{READY}}$ AND PCLK TIMING WITH $\overline{\text{RES}}$ HIGH

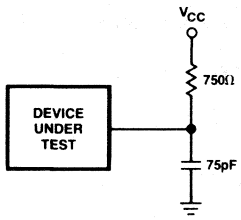


05917A-11

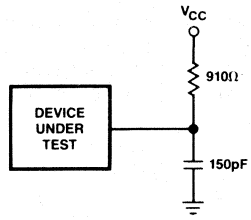
Notes: 1. This is an asynchronous input. The setup and hold times shown are required to guarantee the response shown.
2. Tie 910 ohm ±5% pull-up resistor to the $\overline{\text{READY}}$ output.

SWITCHING TEST CIRCUITS AND WAVEFORMS

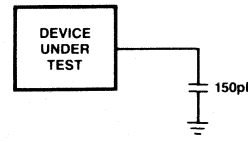
PCLK OUTPUT



$\overline{\text{READY}}$ OUTPUT

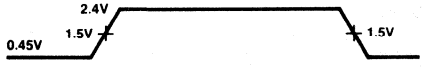


OTHER OUTPUTS

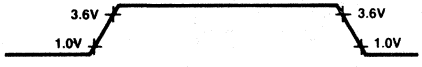


05917A-7

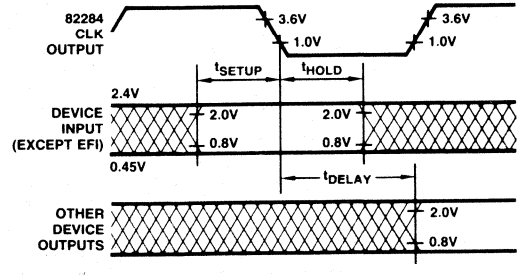
EFI DRIVE AND MEASUREMENT POINTS



CLK OUTPUT MEASUREMENT POINTS



AC SETUP, HOLD AND DELAY TIME MEASUREMENT – GENERAL



05917A-8

82C288

BUS CONTROLLER
FOR iAPX 286 PROCESSORS
PRELIMINARY

DISTINCTIVE CHARACTERISTICS

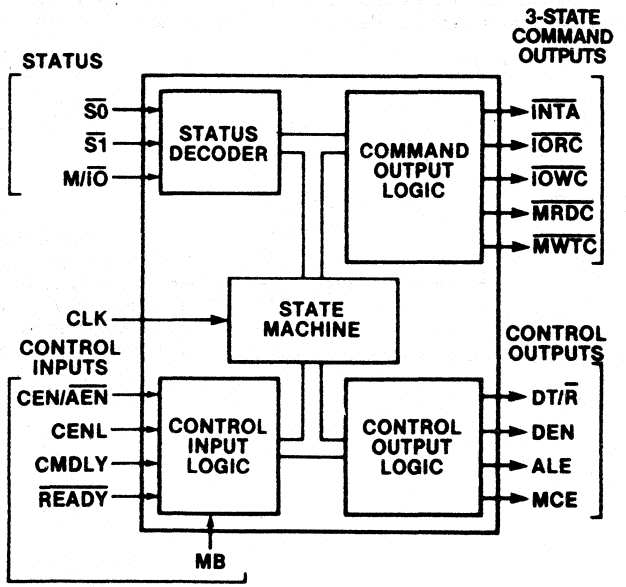
- Provides Commands and Control for Local and System Bus
- Offers Wide Flexibility in System Configurations
- Flexible Command Timing
- Optional Multibus* Compatible Timing
- Control Drivers with 16 ma I_{OL} and 3-State Command Drivers with 32 ma I_{OL}
- Single +5V Supply
- Low Power Operation:
  - I_{CCSB} = 10μA
  - I_{CCOP} = 1mA/MHz

GENERAL DESCRIPTION

The 82C288 Bus Controller is a 20-pin CMOS component for use in iAPX 286 microsystems. The bus controller provides command and control outputs with flexible timing options. Separate command outputs are used for memory and I/O devices. The data bus is controlled with separate data enable and direction control signals.

Two modes of operation are possible via a strapping option: Multibus compatible bus cycles and high speed bus cycles.

BLOCK DIAGRAM



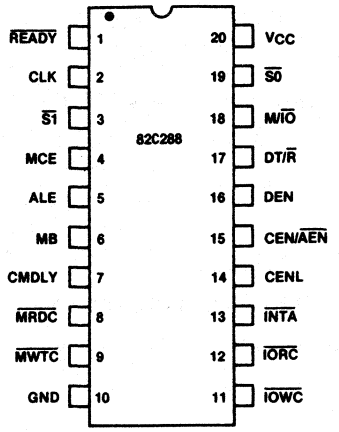
BD004000

*Multibus is a registered trademark of Intel Corporation.

Order # 06100A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

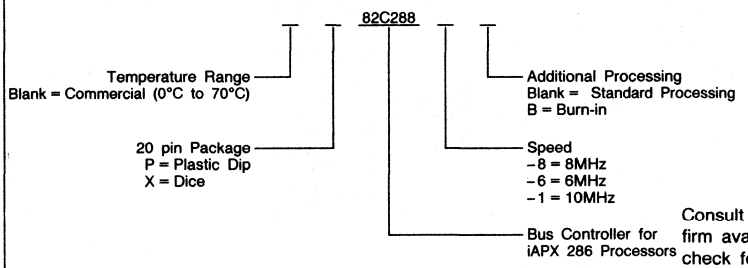


CD005622

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
82C288-6	P
82C288-8	P

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

*A "C" in the middle of the device type denotes CMOS version of the product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1	$\overline{READY}$	I	Ready. $\overline{READY}$ is an active LOW input that indicates the end of the current bus cycle. The 82284 drives $\overline{READY}$ LOW during RESET to force the 82C288 into the Idle state. Multibus mode requires at least one wait state to allow the command outputs to become active. Set-up and hold times must be met for proper operation.
2	CLK	I	System Clock. CLK provides the basic timing control for the 82C288. Its frequency is twice the processor's internal clock frequency. The falling edge of this input signal establishes when inputs are sampled and command and control outputs change.
3,19	$\overline{S1}$, $\overline{S0}$	I	Bus Cycle Status. $\overline{S0}$ and $\overline{S1}$ are active LOW inputs that start a bus cycle and, along with M/$\overline{IO}$, define the type of bus cycle. (See Table 1 for iAPX 286 bus cycle status definitions.) A bus cycle is started when either $\overline{S1}$ or $\overline{S0}$ is sampled LOW at the falling edge of CLK. These inputs have internal pull-up resistors to hold them HIGH when not being driven. Set-up and hold times must be met for proper operation.
4	MCE	O	Master Cascade Enable. MCE signals that a cascade address from a master 8259A interrupt controller may be placed onto the CPU address bus for latching by the address latches under ALE control. The CPU's address bus may then be used to broadcast the cascade address to slave interrupt controllers so only one of them will respond to the interrupt acknowledge cycle. This control output is active HIGH. MCE is only active during interrupt acknowledge cycles and is not affected by any control input. Using MCE to enable cascade address drivers requires latches which save the cascade address on the falling edge of ALE.
5	ALE	O	Address Latch Enable. ALE is an active HIGH output that controls the address latches used to hold an address stable during a bus cycle. ALE is not issued for the halt bus cycle and is not affected by any control inputs.
6	MB	I	Multibus Mode Select. MB determines the timing of the command and control outputs. When HIGH, the bus controller operates with Multibus-compatible timings. When LOW, the bus controller optimizes the command and control output timing for short bus cycles. The function of the CEN/$\overline{AEN}$ input pin is selected by this pin. MB is intended to be a strapping option and not dynamically changed; it may be connected to V_{CC} or GND.
7	CMDLY	I	Command Delay. CMDLY is an active HIGH input that allows the delaying of a command start. If sampled HIGH, the command output is not activated and CMDLY is again sampled at the next CLK cycle. When sampled LOW, the selected command is enabled. If $\overline{READY}$ is detected LOW before the command output is activated, the 82C288 will terminate the bus cycle, even if no command was issued. Set-up and hold times must be satisfied for proper operation. This input may be connected to GND if no delays are required before starting a command. This input has no effect on control outputs.
8	$\overline{MRDC}$	O	Memory Read Command. $\overline{MRDC}$ is an active LOW control output that instructs the memory device to place data onto the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{READY}$ controls when it becomes inactive.
9	$\overline{MWTC}$	O	Memory Write Command. $\overline{MWTC}$ is an active LOW command output that instructs a memory device to read the data on the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{READY}$ controls when it becomes inactive.
10	GND		System Ground: 0V.
11	$\overline{IOWC}$	O	I/O Write Command. $\overline{IOWC}$ is an active LOW command output that instructs an I/O device to read the data on the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{READY}$ controls when it becomes inactive.
12	$\overline{IORC}$	O	I/O Read Command. $\overline{IORC}$ is an active LOW command output that instructs an I/O device to place data onto the data bus. The MB and CMDLY inputs control when this output becomes active. $\overline{READY}$ controls when it becomes inactive.
13	$\overline{INTA}$	O	Interrupt Acknowledge. $\overline{INTA}$ is an active LOW control output that tells an interrupting device that its interrupt request is being acknowledged. The MB and CMDLY inputs control when this output becomes active. $\overline{READY}$ controls when it becomes inactive.
14	CENL	I	Common Enable Latched. CENL is a select signal which enables the bus controller to respond to the current bus cycle being initiated. CENL is an active HIGH input latched internally at the end of each T_S cycle. CENL is used to select the appropriate bus controller for each bus cycle in a system where the CPU has more than one bus it can use. This input may be connected to V_{CC} to select this 82C288 for all transfers. No control inputs affect CENL. Set-up and hold times must be met for proper operation.
15	CEN/$\overline{AEN}$	I	Command Enable/Address Enable. CEN/$\overline{AEN}$ controls the command and DEN outputs of the bus controller. This input may be asynchronous to CLK. Set-up and hold times are given to assure a guaranteed response to synchronous inputs. This input may be connected to V_{CC} or GND. When MB is HIGH, this pin has the $\overline{AEN}$ function. $\overline{AEN}$ is an active LOW input which indicates that the CPU has been granted use of a shared bus and the bus controller command outputs may exit 3-state OFF and become inactive (HIGH). $\overline{AEN}$ HIGH indicates that the CPU does not have control of the shared bus and forces the command outputs into 3-state OFF and DEN inactive (LOW). $\overline{AEN}$ would normally be controlled by an 82289 bus arbiter which activates $\overline{AEN}$ when that arbiter owns the bus to which the bus controller is attached. When MB is LOW, this pin has the CEN function. CEN is an unlatched active HIGH input which allows the bus controller to activate its command and DEN outputs. With MB LOW, CEN LOW forces the command and DEN outputs inactive but does not tristate them.
16	DEN	O	Data Enable. DEN determines when data transceivers connected to the local data bus should be enabled. DEN is an active HIGH control. DEN is delayed for write cycles in the Multibus mode.
17	DT/$\overline{R}$	O	Data Transmit/Receive. DT/$\overline{R}$ establishes the direction of data flow to or from the local data bus. When HIGH, this control output indicates that a write bus cycle is being performed. A LOW indicates a read bus cycle. DEN is always inactive when DT/$\overline{R}$ changes states. This output is HIGH when no bus cycle is active. DT/$\overline{R}$ is not affected by any of the control inputs.
18	M/$\overline{IO}$	I	Memory or I/O Select. M/$\overline{IO}$ determines whether the current bus cycle is in the memory space or I/O space. When LOW, the current bus cycle is in the I/O space. Set-up and hold times must be met for proper operation.
20	V_{CC}		Supply Power: +5V.

Table 1. iAPX 286 Bus Cycle Status Definitions

M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Type of Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	I/O Read
0	1	0	I/O Write
0	1	1	None; idle
1	0	0	Halt or shutdown
1	0	1	Memory read
1	1	0	Memory write
1	1	1	None; idle

DETAILED DESCRIPTION

Introduction

The 82C288 bus controller is used in iAPX 286 systems to provide address latch control, data transceiver control, and standard level-type command outputs. The command outputs are timed and have sufficient drive capabilities for large TTL buses and meet all IEEE-796 requirements for Multibus. A special Multibus mode is provided to satisfy all address/data set-up and hold time requirements. Command timing may be tailored to special needs via a CMDLY input to determine the start of a command and $\overline{READY}$ to determine the end of a command.

Connection to multiple buses are supported with a latched enable input (CENL). An address decoder can determine which, if any, bus controller should be enabled for the bus cycle. This input is latched to allow an address decoder to take full advantage of the pipelined timing on the iAPX 286 local bus.

Buses shared by several bus controllers are supported. An $\overline{AEN}$ input prevents the bus controller from driving the shared bus command and data signals except when enabled by an external bus arbiter such as the 82289.

Separate DEN and DT/$\overline{R}$ outputs control the data transceivers for all buses. Bus contention is eliminated by disabling DEN before changing DT/$\overline{R}$. The DEN timing allows sufficient time for tristate bus drivers to enter 3-state OFF before enabling other drivers onto the same bus.

The term CPU refers to any iAPX 286 processor or support component which may become an iAPX 286 local bus master and thereby drive the 82C288 status inputs.

Processor Cycle Definition

Any CPU which drives the local bus uses an internal clock which is one half the frequency of the system clock (CLK) (see Figure 1). Knowledge of the phase of the local bus master internal clock is required for proper operation of the iAPX 286 local bus. The local bus master informs the bus controller of its internal clock phase when it asserts the status signals. Status signals are always asserted beginning in Phase 1 of the local bus master's internal clock.

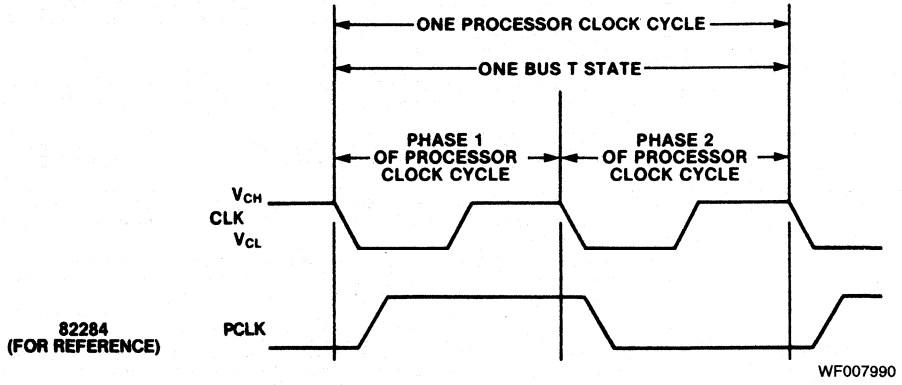


Figure 1. CLK Relationship to the Processor Clock and Bus T-States

Bus State Definition

The 82C288 bus controller has three bus states (see Figure 2): Idle (T_I), Status (T_S), and Command (T_C). Each bus state is two CLK cycles long. Bus state phases correspond to the internal CPU processor clock phases.

The T_I bus state occurs when no bus cycle is currently active on the iAPX 286 local bus. This state may be repeated indefinitely. When control of the local bus is being passed between masters, the bus remains in the T_I state.

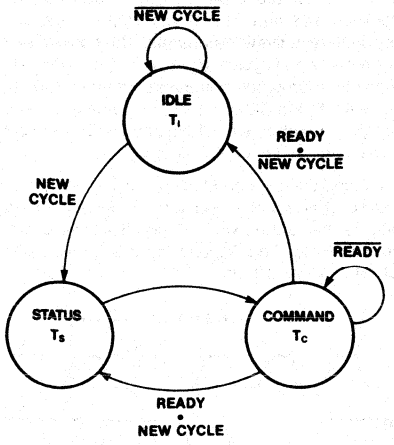


Figure 2. 82C288 Bus States

Bus Cycle Definition

The $\overline{S1}$ and $\overline{S0}$ inputs signal the start of a bus cycle. When either input becomes LOW, a bus cycle is started. The T_S bus state is defined to be the two CLK cycles during which either $\overline{S1}$ or $\overline{S0}$ are active (see Figure 3). These inputs are sampled by the 82C288 at every falling edge of CLK. When either $\overline{S1}$ or $\overline{S0}$ are sampled LOW, the next CLK cycle is considered the second phase of the internal CPU clock cycle.

The local bus enters the T_C bus state after the T_S state. The shortest bus cycle may have one T_S state and one T_C state. Longer bus cycles are formed by repeating T_C states. A repeated T_C bus state is called a wait state.

The $\overline{READY}$ input determines whether the current T_C bus state is to be repeated. The $\overline{READY}$ input has the same timing and effect for all bus cycles. $\overline{READY}$ is sampled at the end of each T_C bus state to see if it is active. If sampled HIGH, the T_C bus state is repeated. This is called inserting a wait state. The control and command outputs do not change during wait states.

When $\overline{READY}$ is sampled LOW, the current bus cycle is terminated. Note that the bus controller may enter the T_S bus state directly from T_C if the status lines are sampled active at the next falling edge of CLK.

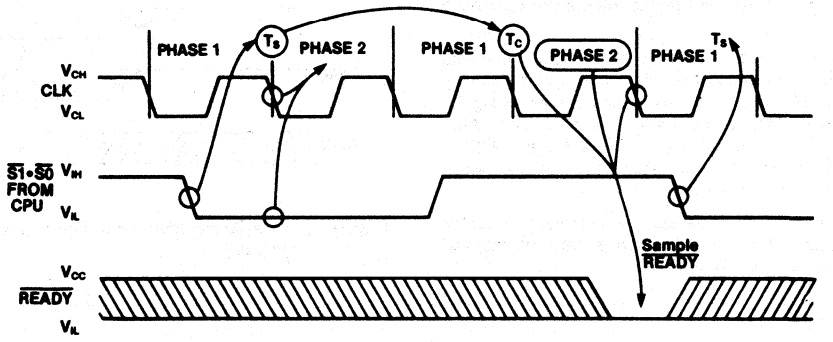


Figure 3. Bus Cycle Definition

Table 2. Command and Control Outputs for Each Type of Bus Cycle

Type of Bus Cycle	M/$\overline{IO}$	$\overline{S1}$	$\overline{S0}$	Command Activated	DT/$\overline{R}$ State	ALE, DEN Issued?	MCE Issued?
Interrupt Acknowledge	0	0	0	$\overline{INTA}$	LOW	YES	YES
I/O Read	0	0	1	$\overline{IORC}$	LOW	YES	NO
I/O Write	0	1	0	$\overline{IOWC}$	HIGH	YES	NO
None; Idle	0	1	1	None	HIGH	NO	NO
Halt/Shutdown	1	0	0	None	HIGH	NO	NO
Memory Read	1	0	1	$\overline{MRDC}$	LOW	YES	NO
Memory Write	1	1	0	$\overline{MWTC}$	HIGH	YES	NO
None; Idle	1	1	1	None	HIGH	NO	NO

Operating Modes

Two types of buses are supported by the 82C288-Multibus and non-Multibus. When the MB input is strapped HIGH, Multibus timing is used. In Multibus mode, the 82C288 delays command and data activation to meet IEEE-796 requirements on address to command active and write data to command active set-up timing. Multibus mode requires at least one wait state in the bus cycle since the command outputs are delayed. The non-Multibus mode does not delay any outputs and does not require wait states. The MB input affects the timing of the command and DEN outputs.

Command and Control Outputs

The type of bus cycle performed by the local bus master is encoded in the M/$\overline{IO}$, $\overline{S1}$, and $\overline{S0}$ inputs. Different command and control outputs are activated depending on the type of bus cycle. Table 2 indicates the cycle decode done by the 82C288 and the effect on command, DT/$\overline{R}$, ALE, DEN, and MCE outputs.

Bus cycles come in three forms: read, write, and halt. Read bus cycles include memory read, I/O read, and interrupt acknowledge. The timing of the associated read command outputs ($\overline{MRDC}$, $\overline{IORC}$ and $\overline{INTA}$), control outputs (ALE, DEN, DT/$\overline{R}$) and control inputs (CEN/$\overline{AEN}$, CENL, CMDLY, MB, and $\overline{READY}$) are identical for all read bus cycles. Read cycles differ only in which command output is activated. The MCE control output is only asserted during interrupt acknowledge cycles.

Write bus cycles activate different control and command outputs with different timing than read bus cycles. Memory write and I/O write are write bus cycles whose timing for command outputs ($\overline{MWTC}$ and $\overline{IOWC}$), control outputs (ALE, DEN, DT/$\overline{R}$) and control inputs (CEN/$\overline{AEN}$, CENL, CMDLY, MB, and $\overline{READY}$) are identical. They differ only in which command output is activated.

Halt bus cycles are different because no command or control output is activated. All control inputs are ignored until the next bus cycle is started via $\overline{S1}$ and $\overline{S0}$.

Figures 4 – 8 show the basic command and control output timing for read and write bus cycles. Halt bus cycles are not shown since they activate no outputs. The basic idle-read-idle and idle-write-idle bus cycles are shown. The signal label CMD represents the appropriate command output for the bus cycle. For Figures 4 – 8, the CMDLY input is connected to GND and CENL to V_{CC}. The effects of CENL and CMDLY are described later in the section on control inputs.

Figures 4, 5 and 6 show non-Multibus cycles. MB is connected to GND while CEN is connected to V_{CC}. Figure 4 shows a read cycle with no wait states while Figure 5 shows a write cycle with one wait state. The $\overline{READY}$ input is shown to illustrate how wait states are added.

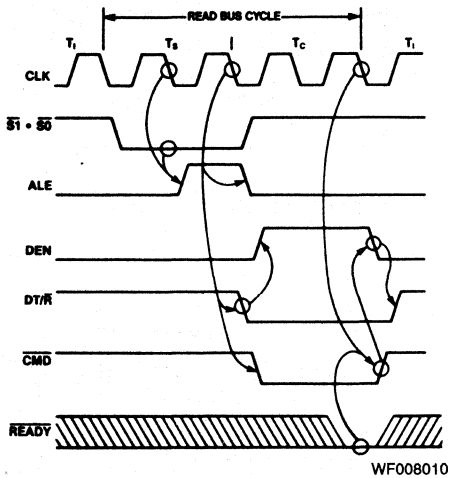


Figure 4. Idle-Read-Idle Bus Cycles with MB = 0

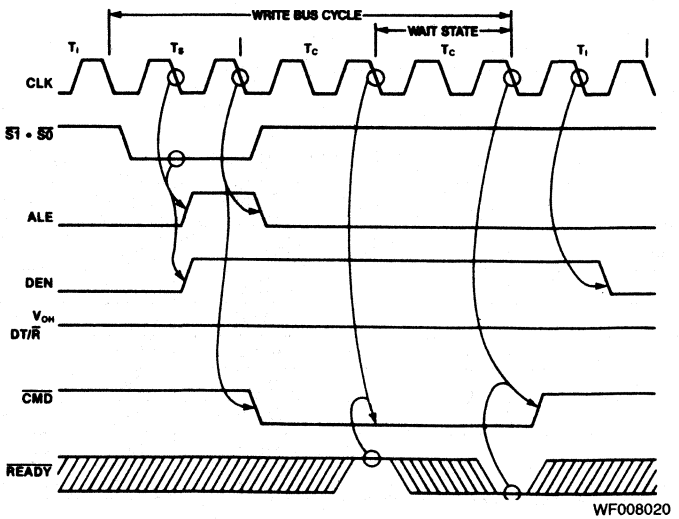


Figure 5. Idle-Write-Idle Bus Cycles with MB = 0

Bus cycles can occur back-to-back with no T_I bus states between T_C and T_S. Back-to-back cycles do not affect the timing of the command and control outputs. Command and control outputs always reach the states shown for the same clock edge (within T_S, T_C, or following bus state) of a bus cycle.

A special case in control timing occurs for back-to-back write cycles with MB = 0. In this case, DT/$\overline{R}$ and DEN remain HIGH between the bus cycles (see Figure 6). The command and ALE output timing does not change.

Figures 7 and 8 show a Multibus cycle with MB = 1. $\overline{AEN}$ and CMDLY are connected to GND. The effects of CMDLY and $\overline{AEN}$ are described later in the section on control inputs. Figure 7 shows a read cycle with one wait state and Figure 8 shows a write cycle with two wait states. The second wait state of the write cycle is shown only for example purposes and is not required. The $\overline{READY}$ input is shown to illustrate how wait states are added.

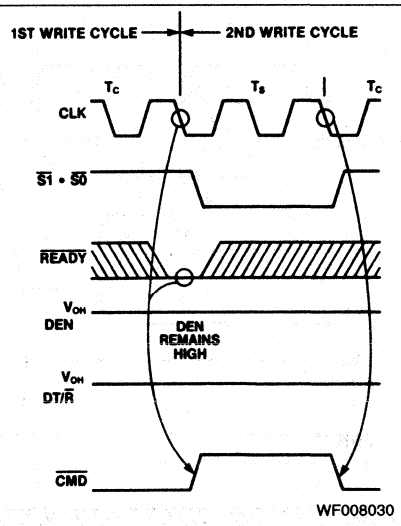


Figure 6. Write-Write Bus Cycles with MB = 0

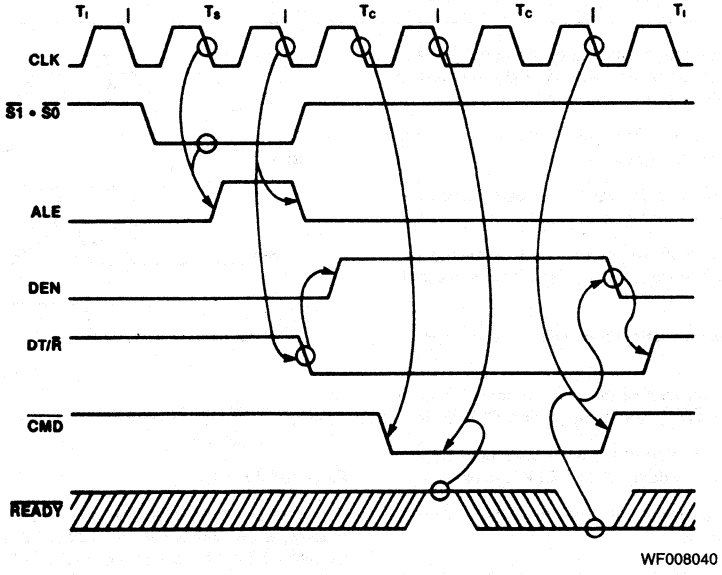


Figure 7. Idle-Read-Idle Bus Cycles with MB = 1

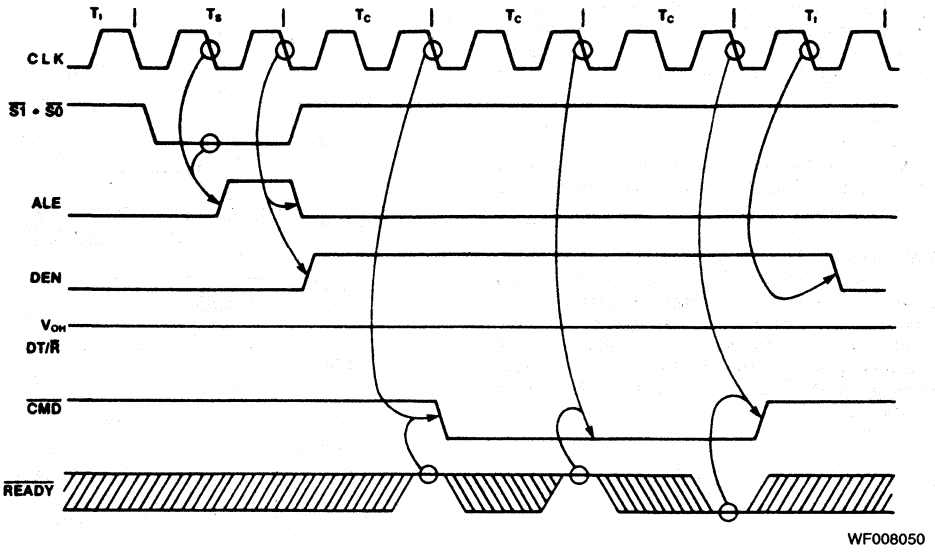


Figure 8. Idle-Write-Idle Bus Cycles with MB = 1

The MB control input affects the timing of the command and DEN outputs. These outputs are automatically delayed in Multibus mode to satisfy three requirements:

1) 50ns minimum set-up time for valid address before any command output becomes active.

2) 50ns minimum set-up time for valid write data before any write command output becomes active.

3) 65ns maximum time from when any read command becomes inactive until the slave's read data drivers reach 3-state OFF.

Three signal transitions are delayed by MB = 1 as compared to MB = 0:

1) The HIGH-to-LOW transition of the read command outputs ($\overline{IORC}$, $\overline{MRDC}$, and $\overline{INTA}$) are delayed one CLK cycle.

2) The HIGH-to-LOW transition of the write command outputs ($\overline{IOWC}$ and $\overline{MWTC}$) are delayed two CLK cycles.

3) The LOW-to-HIGH transition of DEN for write cycles is delayed one CLK cycle.

Back-to-back bus cycles with MB = 1 do not change the timing of any of the command or control outputs. DEN always becomes inactive between bus cycles with MB = 1.

Except for a halt or shutdown bus cycle, ALE will be issued during the second half of T_S for any bus cycle. ALE becomes inactive at the end of the T_S to allow latching the address to keep it stable during the entire bus cycle. The address outputs may change during Phase 2 of any T_C bus state. ALE is not affected by any control input.

Figure 9 shows how MCE is timed during interrupt acknowledge ($\overline{INTA}$) bus cycles. MCE is one CLK cycle longer than ALE to hold the cascade address from a master 8259A valid after the falling edge of ALE. With the exception of the MCE control output, an $\overline{INTA}$ bus cycle is identical in timing to a read bus cycle. MCE is not affected by any control input.

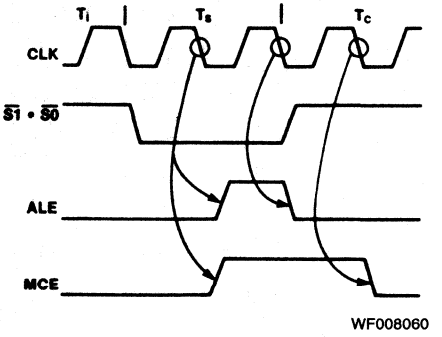


Figure 9. MCE Operation for an $\overline{INTA}$ Bus Cycle

Control Inputs

The control inputs can alter the basic timing of command outputs, allow interfacing to multiple buses, and share a bus between different masters. For many iAPX 286 systems, each CPU will have more than one bus which may be used to perform a bus cycle. Normally, a CPU will only have one bus controller active for each bus cycle. Some buses may be shared by more than one CPU (i.e. Multibus) requiring only one of them use the bus at a time.

Systems with multiple and shared buses use two control input signals of the 82C288 bus controller, CENL and $\overline{AEN}$ (see Figure 10). CENL enables the bus controller to control the current bus cycle. The $\overline{AEN}$ input prevents a bus controller from driving its command outputs. $\overline{AEN}$ HIGH means that another bus controller may be driving the shared bus.

In Figure 10, two buses are shown: a local bus and a Multibus. Only one bus is used for each CPU bus cycle. The CENL inputs of the bus controllers select which bus controller is to perform the bus cycle. An address decoder determines which

bus to use for each bus cycle. The 82C288 connected to the shared Multibus must be selected by CENL and be given access to the Multibus by $\overline{AEN}$ before it will begin a Multibus operation.

CENL must be sampled HIGH at the end of the T_S bus state (see timing waveforms) to enable the bus controller to activate its command and control outputs. If sampled LOW, the commands and DEN will not go active and DT/$\overline{R}$ will remain HIGH. The bus controller will ignore the CMDLY, CEN, and $\overline{READY}$ inputs until another bus cycle is started via $\overline{S1}$ and $\overline{S0}$. Since an address decoder is commonly used to identify which bus is required for each bus cycle, CENL is latched to avoid the need for latching its input.

The CENL input can affect the DEN control output. When MB = 0, DEN normally becomes active during Phase 2 of T_S in write bus cycles. This transition occurs before CENL is sampled. If CENL is sampled LOW, the DEN output will be forced LOW during T_C as shown in the timing waveforms.

When MB = 1, CEN/$\overline{AEN}$ becomes $\overline{AEN}$. $\overline{AEN}$ controls when the bus controller command outputs enter and exit 3-state OFF. $\overline{AEN}$ is intended to be driven by a bus arbiter, like the 82289, which assures only one bus controller is driving the shared bus at any time. When $\overline{AEN}$ makes a LOW-to-HIGH transition, the command outputs immediately enter 3-state OFF and DEN is forced inactive. An inactive DEN should force the local data transceivers connected to the shared data bus into 3-state OFF (see Figure 10). The LOW-to-HIGH transition of $\overline{AEN}$ should only occur during T_I or T_S bus states.

The HIGH-to-LOW transition of $\overline{AEN}$ signals that the bus controller may now drive the shared bus command signals. Since a bus cycle may be active or be in the process of starting, $\overline{AEN}$ can become active during any T-state. $\overline{AEN}$ LOW immediately allows DEN to go to the appropriate state. Three CLK edges later, the command outputs will go active (see timing waveforms). The Multibus requires this delay for the address and data to be valid on the bus before the commands become active.

When MB = 0, CEN/$\overline{AEN}$ becomes CEN. CEN is an asynchronous input which immediately affects the command and DEN outputs. When CEN makes a HIGH-to-LOW transition, the commands and DEN are immediately forced inactive. When CEN makes a LOW-to-HIGH transition, the commands and DEN outputs immediately go to the appropriate state (see timing waveforms). $\overline{READY}$ must still become active to terminate a bus cycle if CEN remains LOW for a selected bus controller (CENL was latched HIGH).

Some memory or I/O systems may require more address or write data set-up time to command active than provided by the basic command output timing. To provide flexible command timing, the CMDLY input can delay the activation of command outputs. The CMDLY input must be sampled LOW to activate the command outputs. CMDLY does not affect the control outputs ALE, MCE, DEN, and DT/$\overline{R}$.

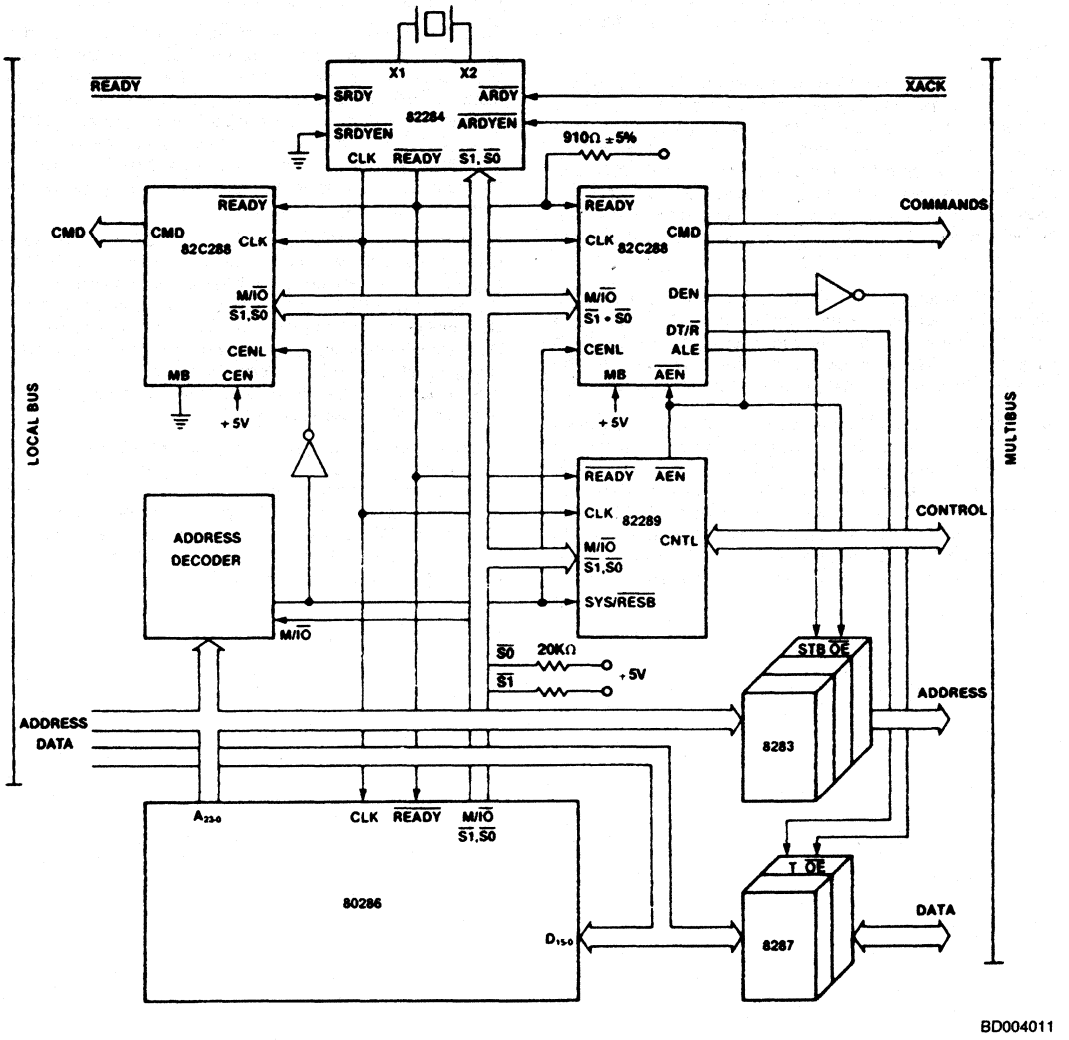


Figure 10. System Use of $\overline{AEN}$ and CENL

CMDLY is first sampled on the falling edge of the CLK ending T_S. If sampled HIGH, the command output is not activated, and CMDLY is again sampled on the next falling edge of CLK. Once sampled LOW, the proper command output becomes active immediately if MB = 0. If MB = 1, the proper command goes active no earlier than shown in Figures 7 and 8.

$\overline{READY}$ can terminate a bus cycle before CMDLY allows a command to be issued. In this case no commands are issued and the bus controller will deactivate DEN and DT/$\overline{R}$ in the same manner as if a command had been issued.

Waveforms Discussion

The waveforms show the timing relationships of inputs and outputs and do not show all possible transitions of all signals in all modes. Instead, all signal timing relationships are shown via the general cases. Special cases are shown when needed. The waveforms provide some functional descriptions of the 82C288; however, most functional descriptions are provided in Figures 3 through 9.

To find the timing specification for a signal transition in a particular mode, first look for a special case in the waveforms. If no special case applies, then use a timing specification for the same or related function in another mode.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to GND ..... −0.5V to +7.0V
Power Dissipation ..... 1 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Ambient Temperature ..... 0 to +70°C
Supply Voltage ..... +4.5 to +5.5V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input LOW Voltage		−.5	.8	V
V_{IH}	Input HIGH Voltage		2.0	V_{CC} + .5	V
V_{ILC}	CLK Input LOW Voltage		−.5	.6	V
V_{IHC}	CLK Input HIGH Voltage		3.8	V_{CC} + .5	V
V_{OL}	Output LOW Voltage Command Outputs Control Outputs	 I_{OL} = 32mA Note 1 I_{OL} = 16mA Note 2		 .45 .45	 V V
V_{OH}	Output HIGH Voltage Command Outputs Control Outputs	 I_{OH} = −5mA Note 1 I_{OH} = −1mA Note 2	 2.4 2.4		 V V
I_F	Input Current ($\overline{S0}$, $\overline{ST}$ and M/$\overline{IO}$ inputs)	Vf = 4.5V		−0.3	mA
I_{IL}	Input Leakage current (all other inputs)	$0V \leqslant V_{IN} \leqslant V_{CC}$		±10	μA
I_{LO}	Output Leakage Current	$.45V \leqslant V_{OUT} \leqslant V_{CC}$		±10	μA
I_{CCSB}	Standby Power Supply Current	V_{CC} = 5.5V, V_{IN} = V_{CC} or GND, Outputs open		10	μA
I_{CCOP}	Operating Power Supply Current	V_{CC} = 5.5V, Outputs open		1	mA/MHz

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V, V_{IN} = +5V or GND)

C_I	Input Capacitance	F_c = 1MHz		10	pF
C_O	Output Capacitance	F_c = 1MHz		20	pF

Notes: 1. Command Outputs are $\overline{INTA}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{MRDC}$, $\overline{MWRC}$.
2. Control Outputs are DT/$\overline{R}$, DEN, ALE and MCE.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

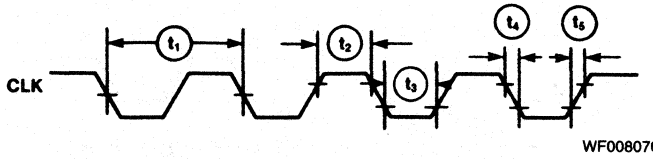
Switching timings are referenced to 0.8V and 2.0V points of signals as illustrated in data sheet waveforms, unless otherwise noted.

No.	Description	Test Conditions	82C288-6 (6MHz)		82C288-8 (8MHz)		Units
			Min	Max	Min	Max	
1	CLK Period		83	250	62	250	ns
2	CLK HIGH Time	at 3.8V	25	235	20	235	ns
3	CLK LOW Time	at 0.6V	20	225	15	230	ns
4	CLK Rise Time	1.0V to 3.5V		10		10	ns
5	CLK Fall Time	3.5V to 1.0V		10		10	ns
6	M/$\overline{IO}$ and Status Set-up Time		28		22.5		ns
7	M/$\overline{IO}$ and Status Hold Time		0		0		ns
8	CENL Set-up Time		30		20		ns
9	CENL Hold Time		0		0		ns
10	$\overline{READY}$ Set-up Time		50		38.5		ns
11	$\overline{READY}$ Hold Time		35		25		ns
12	CMDLY Set-up Time		25		20		ns
13	CMDLY Hold Time		0		0		ns
14	$\overline{AEN}$ Set-up Time	Note 3	30		25		ns
15	$\overline{AEN}$ Hold Time	Note 3	0		0		ns
16	ALE, MCE Active Delay from CLK	Note 4	3	25	3	15	ns
17	ALE, MCE Inactive Delay from CLK	Note 4		35		20	ns
18	DEN (Write) Inactive from CENL	Note 4		35		35	ns
19	DT/$\overline{R}$ LOW from CLK	Note 4		40		20	ns
20	DEN (Read) Active from DT/$\overline{R}$	Note 4	10	50	10	40	ns
21	DEN (Read) Inactive Dly from CLK	Note 4	3	40	3	35	ns
22	DT/$\overline{R}$ HIGH from DEN Inactive	Note 4	5	45	10	40	ns
23	DEN (Write) Active Delay from CLK	Note 4		35		30	ns
24	DEN (Write) Inactive Dly from CLK	Note 4	3	35	3	30	ns
25	DEN Inactive from CEN	Note 4		40		25	ns
26	DEN Active from CEN	Note 4		35		25	ns
27	DT/$\overline{R}$ HIGH from CLK (when CEN = LOW)	Note 4		50		50	ns
28	DEN Active from $\overline{AEN}$	Note 4		35		30	ns
29	$\overline{CMD}$ Active Delay from CLK	Note 5	3	40	3	20	ns
30	$\overline{CMD}$ Inactive Delay from CLK	Note 5	3	30	3	20	ns
31	$\overline{CMD}$ Inactive from CEN	Note 5		35		25	ns
32	$\overline{CMD}$ Active from CEN	Note 5		45		25	ns
33	$\overline{CMD}$ Inactive Enable from $\overline{AEN}$	Note 5		40		40	ns
34	$\overline{CMD}$ Float Delay from $\overline{AEN}$	Note 6		40		40	ns

Notes: 3. $\overline{AEN}$ and MB are asynchronous inputs. This specification is for testing purposes only, to assure recognition at a specific CLK edge.
4. Control output load: C_L = 150pF.
5. Command output load: C_L = 300pF.
6. Float condition occurs when output current is less then I_{LO} in magnitude.

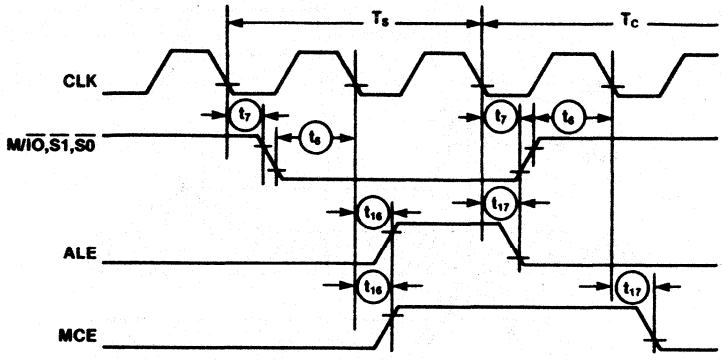
SWITCHING WAVEFORMS

CLK CHARACTERISTICS



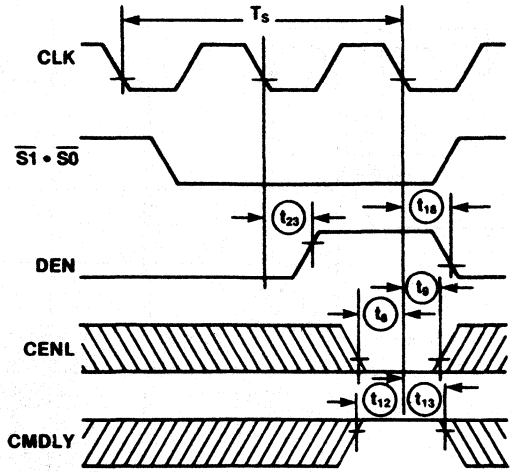
WF008070

STATUS, ALE, MCE CHARACTERISTICS



WF008080

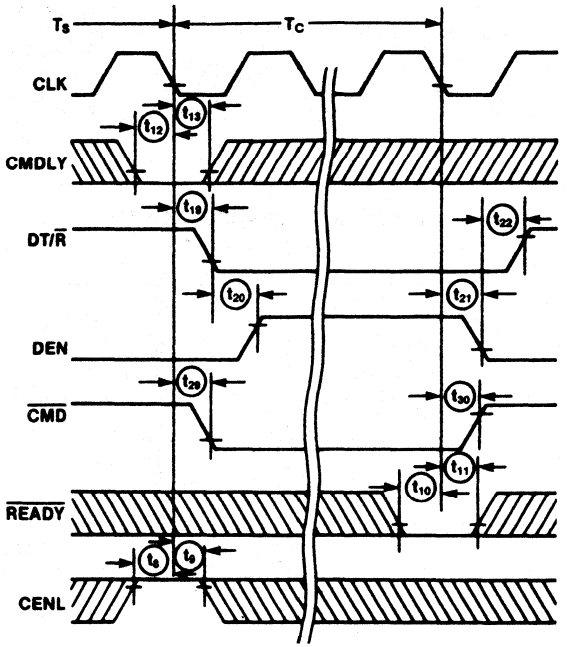
CENL, CMDLY, DEN CHARACTERISTICS WITH MB = 0 and CEN = 1 DURING WRITE CYCLE



WF008090

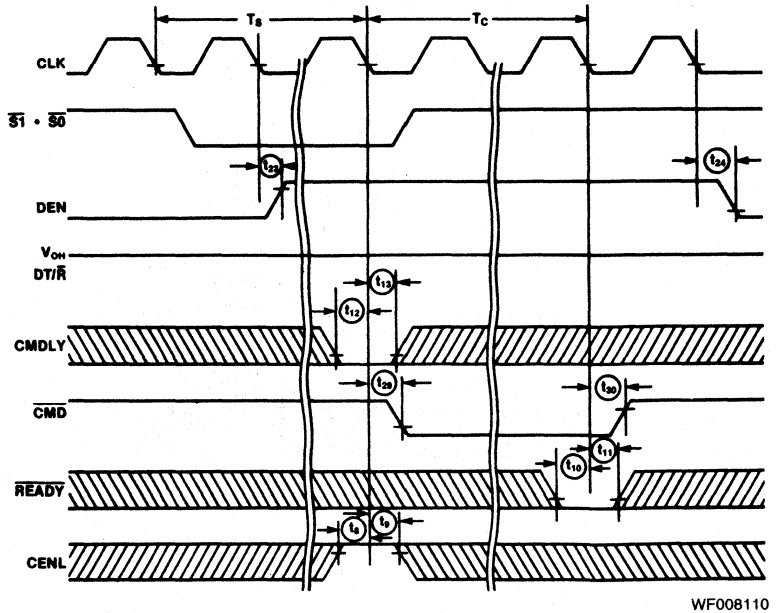
SWITCHING WAVEFORMS (Continued)

READ CYCLE CHARACTERISTICS WITH MB = 0 and CEN = 1



WF008100

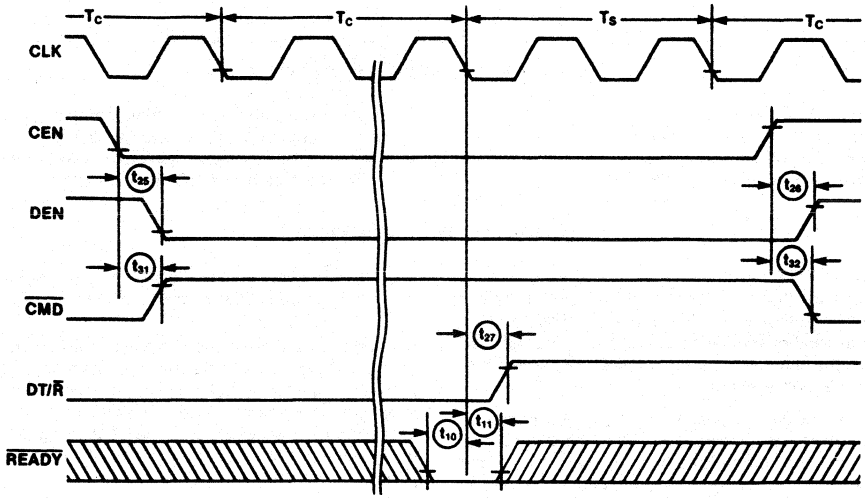
WRITE CYCLE CHARACTERISTICS WITH MB = 0 AND CEN = 1



WF008110

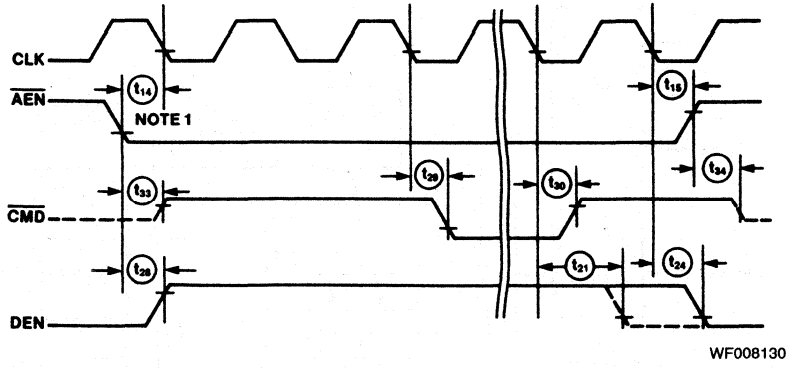
SWITCHING WAVEFORMS (Continued)

CEN CHARACTERISTICS WITH MB = 0



WF008120

AEN CHARACTERISTICS WITH MB = 1



WF008130

Note 1: AEN is an asynchronous input. AEN set-up and hold time is specified to guarantee the response shown in the waveforms.

8231A*

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- 2, 3 and 4MHz operation
- Fixed point 16- and 32-bit operations
- Floating point 32-bit operations
- Add, Subtract, Multiply and Divide
- Trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation
- Float to fixed and fixed to float conversions
- DMA or programmed I/O data transfers

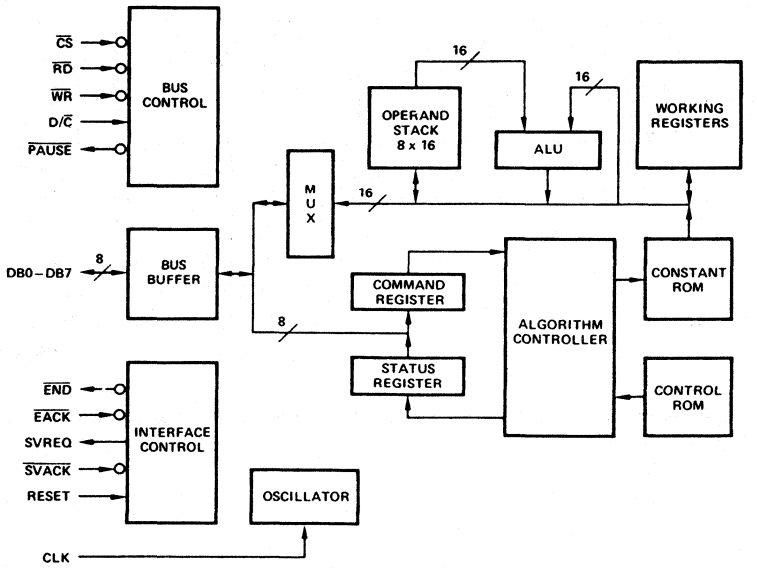
GENERAL DESCRIPTION

The 8231A Arithmetic Processing Unit (APU) is a monolithic MOS/LSI device that provides high performance fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. It may be used to enhance the computational capability of a wide variety of processor-oriented systems.

All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and a command is issued to perform operations on the data in the stack. Results are then available to be retrieved from the stack, or additional commands may be entered.

Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.

BLOCK DIAGRAM



BD003650

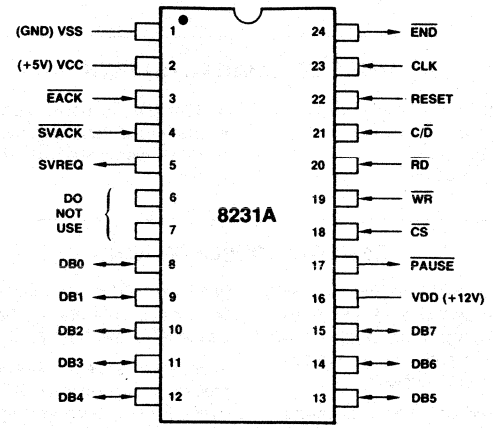
*The 8231A is an AMD invented device more commonly referred to as the Am9511A. See the Am9511A datasheet in Section 2 for additional specifications.

04101A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40

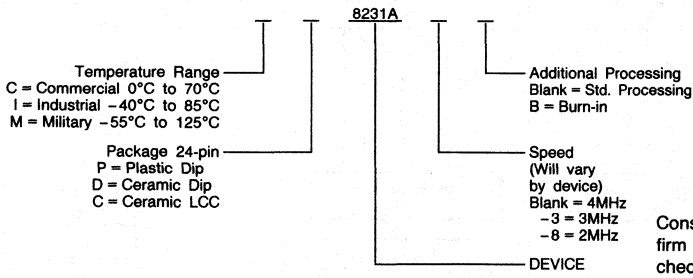


CD005460

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8231A	D
8231A-3	D, ID
8231A-3B	MD
8231A-8	D, ID
8231A-8B	MD

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

8232*

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- Single (32-bit) and double (64-bit) precision capability
- Add, subtract, multiply and divide functions
- Compatible with proposed IEEE format
- Easy interfacing to microprocessors
- 8-bit data bus
- 12V and 5V power supplies
- Stack oriented operand storage
- Direct memory access or programmed I/O Data Transfers

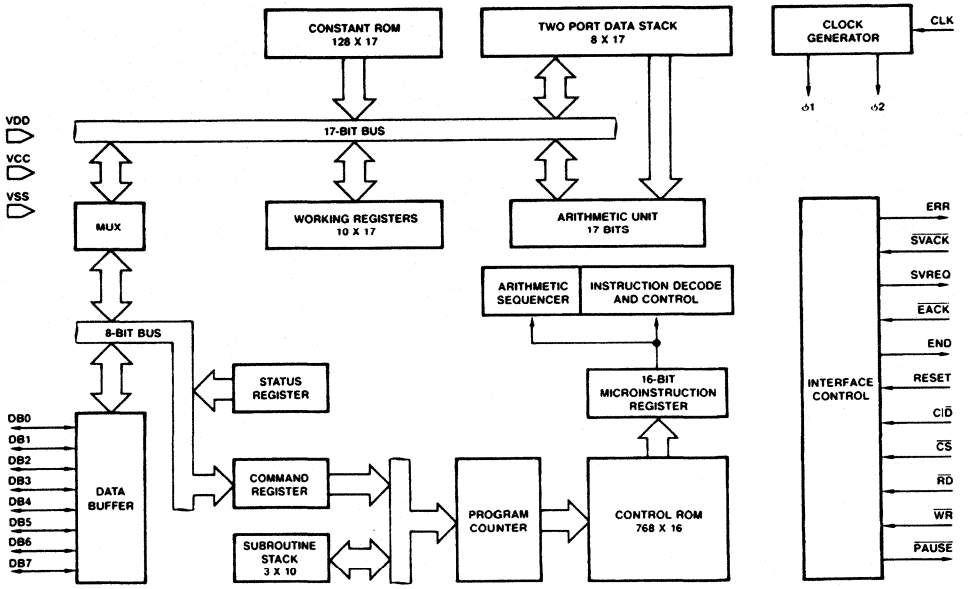
GENERAL DESCRIPTION

The 8232 is high performance floating-point processor unit (FPU). It provides a single precision (32-bit) and double precision (64-bit) add, subtract, multiply and divide operations. It can be easily interfaced to enhance the computational capabilities of the host microprocessor.

The operand, result, status and command information transfers take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack by the host processor and a command is issued to perform an operation on the data stack. The results of this operation are available to the host processor by popping the stack.

Information transfers between the 8232 and the host processor can be handled by using programmed I/O or direct memory access techniques. After completing an operation, the 8232 activates an "end of execution" signal that can be used to interrupt the host processor.

BLOCK DIAGRAM



BD003690

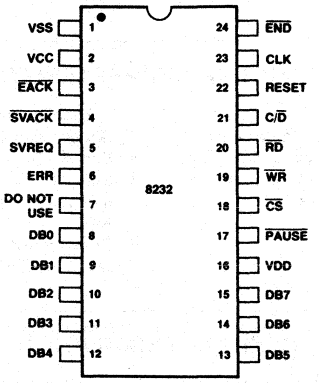
*The 8232 is an AMD invented device more commonly referred to as the Am9512. See the Am9512 datasheet in Section 2 for additional specifications.

04100A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-24

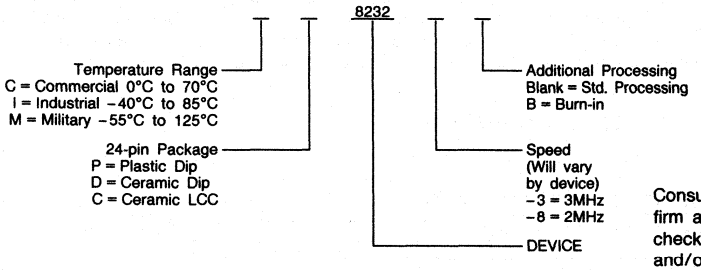


CD005440

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8232-8	D, ID
8232-8B	MD
8232-3	D, ID
8232-3B	MD

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

8237A

Multimode DMA Controller
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Four independent DMA channels, each with separate registers for Mode Control, Current Address, Base Address, Current Word Count and Base Word Count
- Transfer modes: Block, Demand, Single Word, Cascade
- Independent autoinitialization of all channels
  - Memory-to-memory transfers
  - Memory block initialization
  - Address increment or decrement
  - Master system disable
  - Enable/disable control of individual DMA requests
- Directly expandable to any number of channels
- End of Process input for terminating transfers
- Software DMA requests
- Independent polarity control for DREQ and DACK signals
- Compressed timing option speeds transfers – up to 2M words/second

GENERAL DESCRIPTION

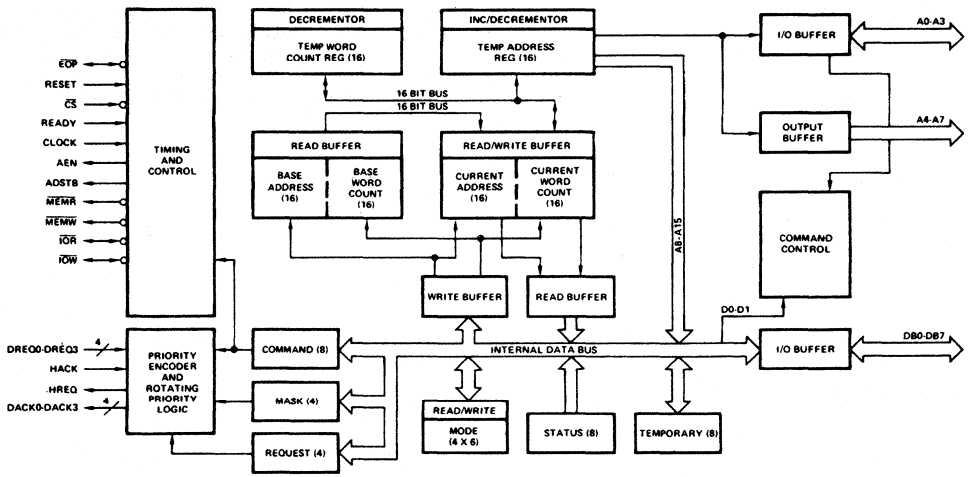
The 8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information to or from the system memory. Memory-to-memory transfer capability is also provided. The 8237A offers a wide variety of programmable control features to enhance data throughout and system optimization and to allow dynamic reconfiguration under program control.

The 8237A is designed to be used in conjunction with an external 8-bit address register such as the Am74LS373. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process (EOP).

Each channel has a full 64K address and word count capability. An external EOP signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

BLOCK DIAGRAM



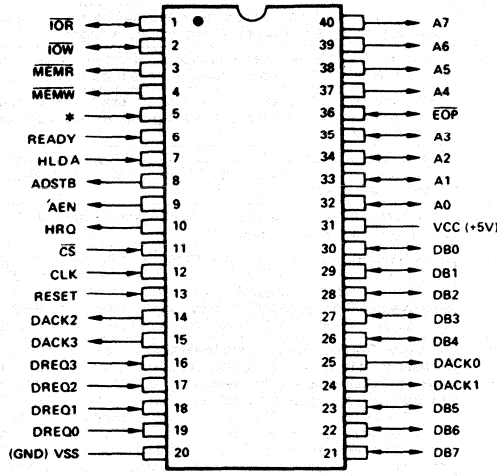
BD003700

Figure 1.

03726A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40-1, P-40-1



CD005471

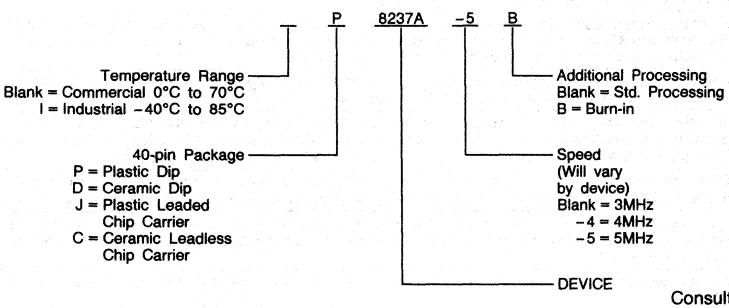
Figure 2.

Note: Pin 1 is marked for orientation

Also available in PLCC. See Section 7 for pinout details.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8237A 8237AB 8237A-4 8237A-4B	P, D, ID
8237A-5 8237A-5B	P, D, ID
8237A 8237A-4	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
31	V_{CC}		Power: +5 volt supply.
20	V_{SS}		Ground: Ground.
12	CLK	I	Clock Input: Clock Input controls the internal operations of the 8237A and its rate of data transfers. The input may be driven at up to 3 MHz for the standard 8237A and up to 5 MHz for the 8237A-5.
11	$\overline{CS}$	I	Chip Select: Chip Select is an active low input used to select the 8237A as an I/O device during the Idle cycle. This allows CPU communication on the data bus.
13	RESET	I	Reset: Reset is an active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
6	READY	I	Ready: Ready is an input used to extend the memory read and write pulses from the 8237A to accommodate slow memories or I/O peripheral devices. Ready must not make transitions during its specified set-up/hold time.
7	HLDA	I	Hold Acknowledge: The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.
19-16	DREQ0-DREQ3	I	DMA Request: The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active.
30-26, 23-21	DB0-DB7	I/O	Data Bus: The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the 8237A control registers. During DMA cycles the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the 8237A on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
1	$\overline{IOR}$	I/O	I/O Read: I/O Read is a bidirectional active low three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the 8237A to access data from a peripheral during a DMA Write transfer.
2	$\overline{IOW}$	I/O	I/O Write: I/O Write is a bidirectional active low three-state line. In the Idle cycle, it is an Input control signal used by the CPU to load information into the 8237A. In the Active cycle, it is an output control signal used by the 8237A to load data to the peripheral during a DMA Read transfer.
36	$\overline{EOP}$	I/O	End of Process: End of Process is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional $\overline{EOP}$ pin. The 8237A allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{EOP}$ Input low with an external $\overline{EOP}$ signal. The 8237A also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{EOP}$ signal which is output through the $\overline{EOP}$ Line. The reception of $\overline{EOP}$, either internal or external, will cause the 8237A to terminate the service, reset the request, and, if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{EOP}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains unchanged. During memory-to-memory transfers, $\overline{EOP}$ will be output when the TC for channel 1 occurs. $\overline{EOP}$ should be tied high with a pull-up resistor if it is not used to prevent erroneous end of process inputs.
32-35	A0-A3	I/O	Address: The four least significant address lines are bidirectional three-state signals. In the Idle cycle they are inputs and are used by the CPU to address the register to be loaded or read. In the Active cycle they are outputs and provide the lower 4 bits of the output address.
37-40	A4-A7	O	Address: The four most significant address lines are three-state outputs and provide 4 bits of address. These lines are enabled only during the DMA service.
10	HRQ	O	Hold Request: This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes 8237A to issue the HRQ. After HRQ goes active, at least one clock cycle (TCY) must occur before HLDA goes active.
25, 24, 14, 15	DACK0-DACK3	O	DMA Acknowledge: DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.
9	AEN	O	Address Enable: Address Enable enables the 8-bit latch containing the upper 8 address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers. AEN is active HIGH.
8	ADSTB	O	Address Strobe: The active high, Address Strobe is used to strobe the upper address byte into an external latch.
3	$\overline{MEMR}$	O	Memory Read: The Memory Read signal is an active low three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
4	$\overline{MEMW}$	O	Memory Write: The Memory Write is an active low three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.

03726A

Refer to page 7-1 for Essential Information on Military Devices

DETAILED DESCRIPTION

The 8237A block diagram includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The 8237A contains 344 bits of internal memory in the form of registers. Figure 3 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Figure 3. 8237A Internal Registers

The 8237A contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the 8237A. The Program Command Control block decodes the various commands given to the 8237A by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input. In 8237A systems this input will usually be the $\phi 2$ TTL clock from an 8224 or CLK from an 8085AH or 8284A. For 8085AH-2 systems above 3.9 MHz, the 8085 CLK(OUT) does not satisfy 8237A-5 clock LOW and HIGH time requirements. In this case, an external clock should be used to drive the 8237A-5.

DMA Operation

The 8237A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 8237A can assume seven separate states, each composed of one full clock period. State I (SI) is the inactive state. It is entered when the 8237A has no valid DMA requests pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State S0 (S0) is the first state of a DMA service. The 8237A has requested a hold but the processor has not yet returned an acknowledge. The 8237A may still be programmed until it receives HLDA from the CPU. An acknowledge from the CPU will signal that DMA transfers may begin. S1, S2, S3 and S4 are the working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 8237A. Note that the data is transferred directly from the I/O device to memory (or vice versa) with $\overline{IOR}$ and $\overline{MEMW}$ (or $\overline{MEMR}$ and $\overline{IOW}$) being active at the same time. The data is not read into or driven out of the 8237A in I/O-to-memory or memory-to-I/O DMA transfers.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

Idle Cycle

When no channel is requesting service, the 8237A will enter the Idle cycle and perform "SI" states. In this cycle the 8237A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample $\overline{CS}$, looking for an attempt by the microprocessor to write or read the internal registers of the 8237A. When $\overline{CS}$ is low and HLDA is low, the 8237A enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0 – A3 are inputs to the device and select which registers will be read or written. The $\overline{IOR}$ and $\overline{IOW}$ lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip-flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip-flop is reset by Master Clear or Reset. A separate software command can also reset this flip-flop.

Special software commands can be executed by the 8237A in the Program Condition. These commands are decoded as sets of addresses with the $\overline{CS}$ and $\overline{IOW}$. The commands do not make use of the data bus. Instructions include Clear First/Last Flip-Flop and Master Clear.

Active Cycle

When the 8237A is in the Idle cycle and a non-masked channel requests a DMA service, the device will output an HRQ to the microprocessor and enter the Active cycle. It is in this cycle that the DMA service will take place, in one of four modes:

Single Transfer Mode – In Single Transfer mode the device is programmed to make one transfer only. The word count will be decremented and the address decremented or incremented following each transfer. When the word count "rolls over" from zero to FFFFH, a Terminal Count (TC) will cause an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HRQ will go inactive and release the bus to the system. It will again go active and, upon receipt of a new HLDA, another single transfer will be performed, in 8080A, 8085AH, 8088, or 8086 system this will ensure one full machine cycle execution between DMA transfers. Details of timing between the 8237A and other bus control protocols will depend upon the characteristics of the microprocessor involved.

Block Transfer Mode – In Block Transfer mode the device is activated by DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFH, or an external End of Process (EOP) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialization will occur at the end of the service if the channel has been programmed for it.

Demand Transfer Mode – In Demand Transfer mode the device is programmed to continue making transfers until a TC or external $\overline{EOP}$ is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the

microprocessor is allowed to operate, the intermediate values of address and word count are stored in the 8237A Current Address and Current Word Count registers. Only an $\overline{EOP}$ can cause an Autoinitialize at the end of the service. $\overline{EOP}$ is generated either by TC or by an external signal.

Cascade Mode – This mode is used to cascade more than one 8237A together for simple system expansion. The HRQ and HLDA signals from the additional 8237A are connected to the DREQ and DACK signals of a channel of the initial 8237A. This allows the DMA requests of the additional device to propagate through the priority network circuitry of the preceding device. The priority chain is preserved and the new device must wait for its turn to acknowledge requests. Since the cascade channel of the initial 8237A is used only for prioritizing the additional device, it does not output any address or control signals of its own. These could conflict with the outputs of the active channel in the added device. The 8237A will respond to DREQ and DACK but all other outputs except HRQ will be disabled. The ready input is ignored.

Figure 4 shows two additional devices cascaded into an initial device using two of the previous channels. This forms a two level DMA system. More 8237As could be added at the second level by using the remaining channels of the first level. Additional devices can also be added by cascading into the channels of the second level devices, forming a third level.

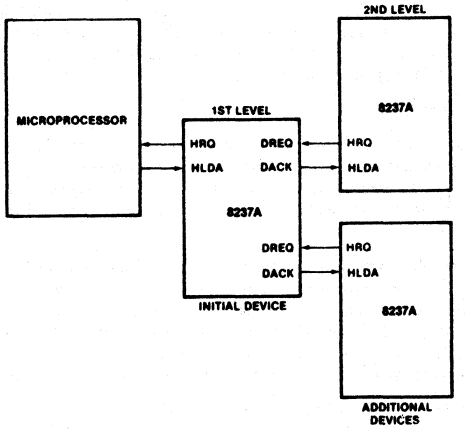


Figure 4. Cascaded 8237As

Transfer Types

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from and I/O device to the memory by activating $\overline{MEMW}$ and $\overline{IOR}$. Read transfers move data from memory to an I/O device by activating $\overline{MEMR}$ and $\overline{IOW}$. Verify transfers are pseudo transfers. The 8237A operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive. The ready input is ignored in verify mode.

Memory-to-Memory – To perform block moves of data from one memory address space to another with a minimum of program effort and time, the 8237A includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 to 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The 8237A requests a DMA service in the normal manner. After HLDA is true, the device, using four state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 8237A internal Temporary register. Channel 1 then performs a four-state transfer of the data from the Temporary register to memory using the address is its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 current Word Count is decremented. When the word count of channel 1 goes to FFFFH, a TC is generated causing an $\overline{EOP}$ output terminating the service.

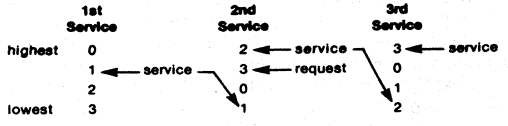
Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

The 8237A will respond to external $\overline{EOP}$ signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. The timing of memory-to-memory transfers is found in Figure 12. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

Autoinitialize – By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Autoinitialize initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word count registers of that channel following $\overline{EOP}$. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not altered when the channel is in Autoinitialize. Following Autoinitialize the channel is ready to perform another DMA service, without CPU intervention, as soon as a valid DREQ is detected. In order to Autoinitialize both channels in a memory-to-memory transfer, both word counts should be programmed identically. If interrupted externally, $\overline{EOP}$ pulses should be applied in both bus cycles.

Priority – The 8237A has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interferring with that service until it is completed.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing – In order to achieve even greater throughput where system characteristics permit, the 8237A can compress the transfer time to two clock cycles. From Figure 11 it can be seen that state S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states will still occur

when A8 – A15 need updating (see Address Generation). Timing for compressed transfers is found in Figure 14.

Address Generation – In order to reduce pin count, the 8237A multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output by the 8237A directly. Lines A0 – A7 should be connected to the address bus. Figure 11 shows the time relationships between CLK, AEN, ADSTB, DB0 – DB7 and A0 – A7.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 8237A executes S1 states only when updating of A8 – A15 in the latch is necessary. This means for long services, S1 states and Address Strobes may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

REGISTER DESCRIPTION

Current Address Register – Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize takes place only after an $\overline{EOP}$.

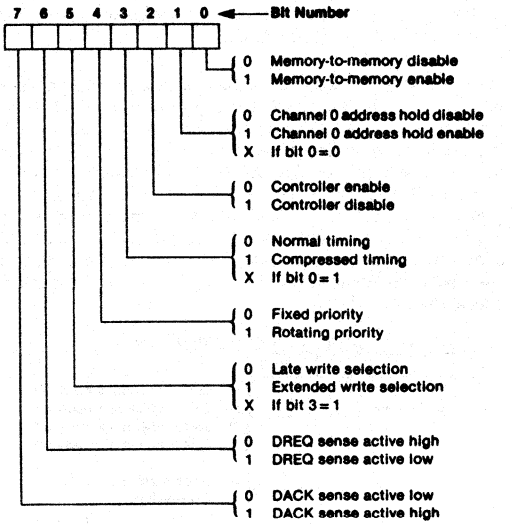
Current Word Register – Each channel has a 16-bit Current Word Count register. This register determines the number of transfers to be performed. The actual number of transfers will be one more than the number programmed in the Current Word Count register (i.e., programming a count of 100 will result in 101 transfers). The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes from zero to FFFFH, a TC will be generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service it may also be reinitialized by an Autoinitialization back to its original value. Autoinitialize can occur only when an $\overline{EOP}$ occurs. If it is not Autoinitialized, this register will have a count of FFFFH after TC.

Base Address and Base Word Count Registers – Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original value of their associated current registers. During Autoinitialize these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

Command Register – This 8-bit register controls the operation of the 8237A. It is programmed by the microprocessor in the Program Condition and is cleared by Reset or a Master Clear instruction. The following table lists the function of the command bits. See Figure 6 for address coding.

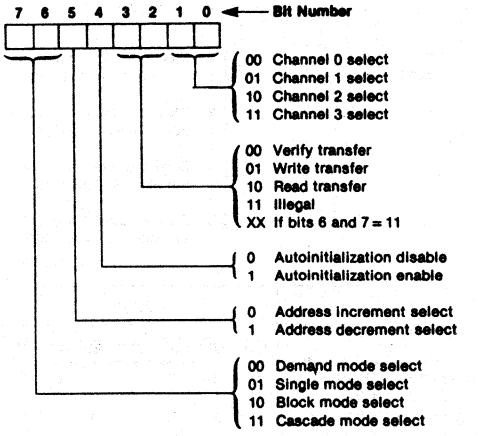
Mode Register – Each channel has a 6-bit Mode register associated with it. When the register is being written to by the microprocessor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written.

Command Register



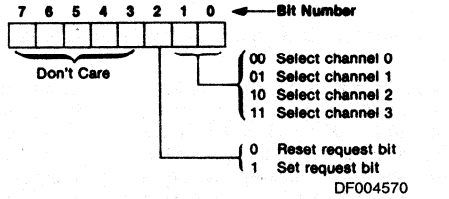
DF004550

Mode Register



DF004560

Request Register

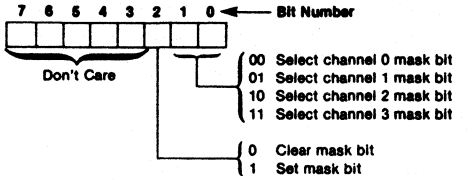


DF004570

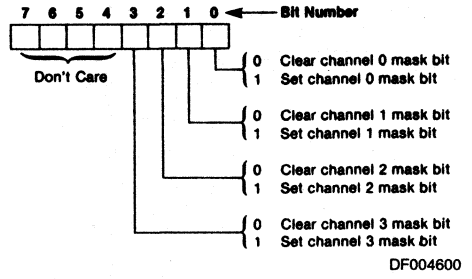
Request Register – The 8237A can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the 4-bit Request register. These are nonmaskable and subject to

prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external $\overline{EOP}$. The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 5 for register address coding. In order to make a software request, the channel must be in Block Mode.

Mask Register – Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an $\overline{EOP}$ if the channel is not programmed for Autoinitialize. Each bit of the 4-bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 5 for instruction addressing.



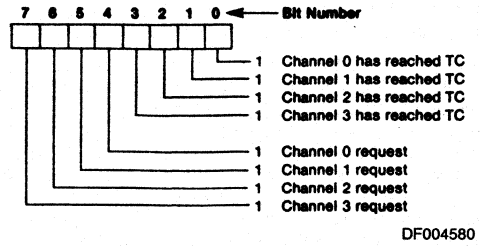
All four bits of the Mask register may also be written with a single command.



Register	Operation	Signals						
		$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Figure 5. Definition of Register Codes

Status Register – The Status register is available to be read out of the 8237A by the microprocessor. It contains information about the status of the devices at this point. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0 – 3 are set every time a TC is reached by that channel or an external $\overline{EOP}$ is applied. These bits are cleared upon Reset and on each Status Read. Bits 4 – 7 are set whenever their corresponding channel is requesting service.



Temporary Register – The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands – These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The three software commands are:

Clear First/Last Flip-Flop: This command is executed prior to writing or reading new address or word count information to the 8237A. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 8237A will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Figure 6 lists the address codes for the software commands:

Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

Figure 6. Software Command Codes

Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0-DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W)0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W)0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W)0-W7
			0	0	1	0	1	1	1	1	W8-W15

PROGRAMMING

The 8237A will accept programming from the host processor any time that HLDA is inactive; this is true even if HRQ is active. The responsibility of the host is to assure that programming and HLDA are mutually exclusive. Note that a problem can occur if a DMA request occurs, on an unmasked channel while the 8237A is being programmed. For instance, the CPU may be starting to reprogram the two byte Address register of channel 1 when channel 1 receives a DMA request. If the 8237A is enabled (bit 2 in the command register is 0) and channel 1 is unmasked, a DMA service will occur after only one byte of the Address register has been reprogrammed. This can be avoided by disabling the controller (setting bit 2 in the command register) or masking the channel before programming any other registers. Once the programming is complete, the controller can be enabled/unmasked.

After power-up it is suggested that all internal locations, especially the Mode registers, be loaded with some valid value. This should be done even if some channels are unused.

APPLICATIONS INFORMATION

Figure 8 shows a convenient method for configuring a DMA system with the 8237A controller and an 8080A/8085AH microprocessor system. The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 8237A takes control of the address bus, the data bus and the control bus. The address for the first transfer operation comes out in two bytes – the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into the 8282 8-bit latch to complete the full 16 bits of the address bus. The 8282 is a high-speed, 8-bit, three-state latch in a 20-pin package. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 8237A is used.

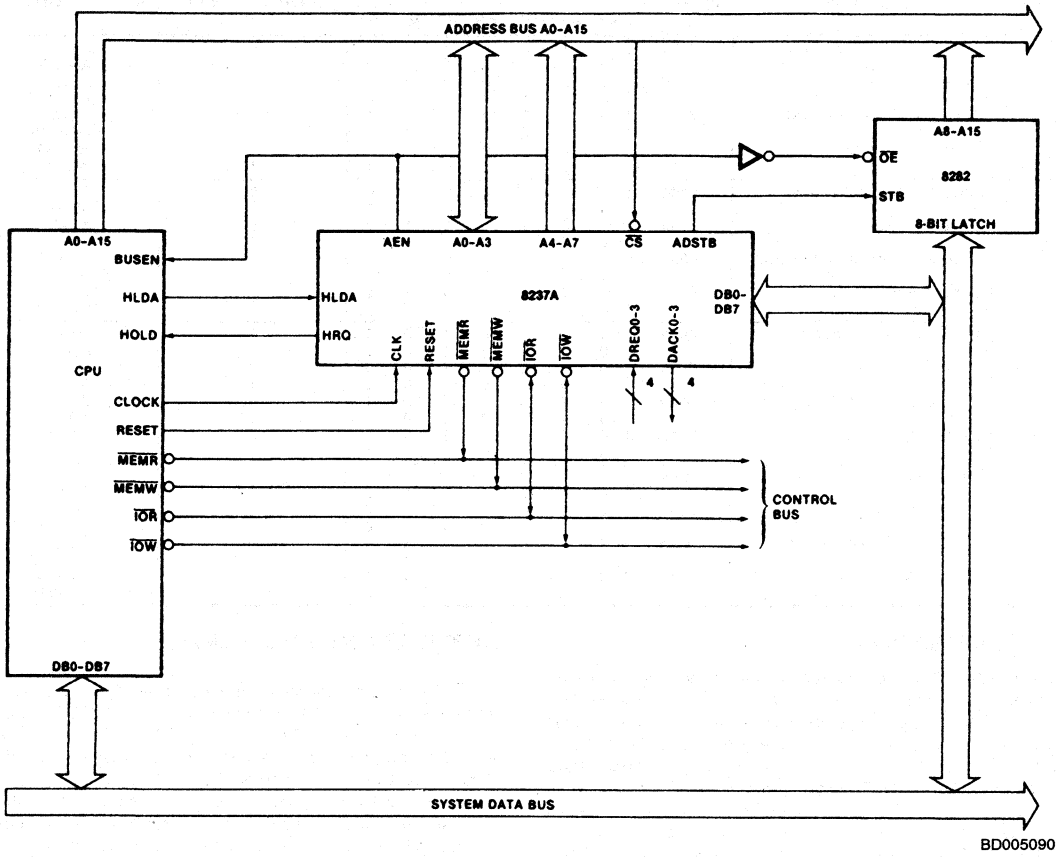


Figure 8. 8237A System Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Supply Voltage ..... −0.5 to +7.0V
All Signal Voltages
with Respect to Ground ..... −0.5 to +7.0V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A or T_C	V_{CC}
8237A 8237A-4 8237A-5	T_A = 0° to 70°C	5V ±5%

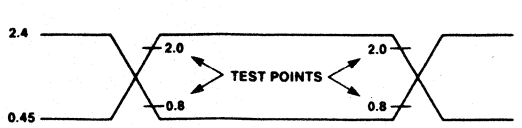
Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{OH}	Output HIGH Voltage	I_{OH} = −200μA	2.4			Volts
		I_{OH} = −100μA, (HRQ Only)	3.3			
V_{OL}	Output LOW Voltage	I_{OL} = 2.0mA (Data bus)			.45	Volts
		I_{OL} = 3.2mA (Other outputs)			.45	
V_{IH}	Input HIGH Voltage		2.0		V_{CC} + 0.5	Volts
V_{IL}	Input LOW Voltage		−0.5		0.8	Volts
I_{LD}	Input Load Current	GND ≤ V_{IN} ≤ V_{CC}	−10		+10	μA
I_{OLK}	Output Leakage Current	V_{CC} ≤ V_{OUT} ≤ GND + .40	−10		+10	μA
I_{CC}	Supply Current	T_A = +25°C		65	130	mA
		T_A = 0°C		75	150	
C_{OUT}	Output Capacitance	f_c = 1.0MHz, Inputs = 0V		4	8	pF
C_{IN}	Input Capacitance			8	15	pF
C_{IO}	I/O Capacitance			10	18	pF

Notes:
1. Typical values are for T_A = 25°C, nominal supply voltage and nominal processing parameters.
2. Input timing parameters assume transition times of 20ns or less. Waveform measurement points for both input and output signals are 2.0V for High and 0.8V for Low, unless otherwise noted.
3. Output loading is 1 Standard TTL gate 50pF capacitance unless noted otherwise.
4. The new $\overline{IOW}$ or $\overline{MEMW}$ pulse width for normal write will be t_{CY}-100ns and for extended write will be $2t_{CY}$-100ns. The net $\overline{IOR}$ or $\overline{MEMR}$ pulse width for nomal read will be $2t_{CY}$-50ns and for compressed read will be t_{CY}-50ns.
5. t_{DQ} is specified for two different output HIGH levels. t_{DQ1} is measured at 2.0V. t_{DQ2} is measured at 3.3V. The value for t_{DQ2} assumes an external 3.3kΩ pull-up resistor connected from HRQ to V_{CC}.
6. DREQ should be held active until DACK is returned.
7. DREQ and DACK signals may be active High or active Low. Timing diagrams assume the active High mode.
8. Output loading on the data bus is 1 Standard TTL gate plus 15pF for the minimum value and 1 Standard TTL gate plus 100pF for the maximum value.
9. Successive read and/or write operations by the external processor to program or examine the controller must be timed to allow at least 600ns for the 8237A, at least 450ns for the 8237A-4, and 400ns for the 8237A-5 as recovery time between active read or write pulses.
10. Parameters are listed in alphabetical order.
11. Pin 5 is an input that should always be at a logic High level. An internal pull-up resister will establish a logic High when the pin is left floating. Alternatively, pin 5 may be tied to V_{CC}.
12. Signals $\overline{READ}$ and $\overline{WRITE}$ refer to $\overline{IOR}$ and $\overline{MEMW}$ respectively for peripheral-to-memory DMA operations and to $\overline{MEMR}$ and $\overline{IOW}$ respectively for memory-to-peripheral DMA operations.
13. If N wait states are added during the write-to-memory half of a memory-to-memory transfer, this parameter will increase by N (t_{CY}).
14. Because $\overline{EOP}$ high from clock high is load dependent, users wishing to test these parameters should use a 2K pull-up resistor and a tester with 50pf or less load capacitance. Time constant RC = 120ns is added to the specified number in the data sheet for testing.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF009540

A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0." (Note 2)

03726A

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Note 1)
Active Cycle (Notes 2, 3, 10, 11 and 12)

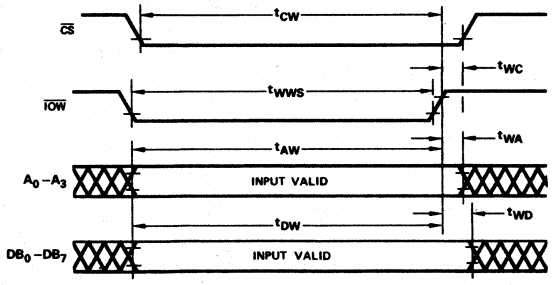
Parameters	Description	8237A		8237A-4		8237A-5		Units
		Min	Max	Min	Max	Min	Max	
t_{AEL}	AEN HIGH from CLK LOW (S_1) Delay Time		300		225		200	ns
t_{AET}	AEN LOW from CLK HIGH (S_1) Delay Time		200		150		130	ns
t_{AFAB}	ADR Active to Float Delay from CLK HIGH		150		120		90	ns
t_{AFC}	$\overline{READ}$ or $\overline{WRITE}$ Float from CLK HIGH		150		120		120	ns
t_{AFDB}	DB Active to Float Delay from CLK HIGH		250		190		170	ns
t_{AHR}	ADR from $\overline{READ}$ HIGH Hold Time	t_{CY}-100		t_{CY}-100		t_{CY}-100		ns
t_{AHS}	DB from ADSTB LOW Hold Time	50		40		30		ns
t_{AHW}	ADR from WRITE HIGH Hold Time	t_{CY}-50		t_{CY}-50		t_{CY}-50		ns
t_{AK}	DACK Valid from CLK LOW Delay Time		250		220		170	ns
	$\overline{EOP}$ HIGH from CLK HIGH Delay Time		250		190		170	
	$\overline{EOP}$ LOW to CLK HIGH Delay Time		250		190		170	
t_{ASM}	ADR Stable form CLK HIGH		250		190		170	ns
t_{ASS}	DB to ADSTB LOW Set-up Time	100		100		100		ns
t_{CH}	Clock High Time (Transitions $\leqslant$ 10ns)	120		100		80		ns
t_{CL}	Clock Low Time (Transitions $\leqslant$ 10ns)	150		110		68		ns
t_{CY}	CLK Cycle Time	320		250		200		ns
t_{DCL}	CLK HIGH to $\overline{READ}$ or $\overline{WRITE}$ LOW Delay (Note 4)		270		200		190	ns
t_{DCTR}	$\overline{Read}$ HIGH from CLK HIGH (S_4) Delay Time (Note 4)		270		210		190	ns
t_{DCTW}	$\overline{WRITE}$ HIGH from CLK HIGH (S_4) Delay Time (Note 4)		200		150		130	ns
t_{DQ1}	HRQ Valid from CLK HIGH Delay Time (Note 5)		160		120		120	ns
t_{DQ2}			250		190		120	
t_{EPS}	$\overline{EOP}$ LOW from CLK LOW Set-up Time	60		45		40		ns
t_{EPW}	$\overline{EOP}$ Pulse Width	300		225		220		ns
t_{FAAB}	ADR Float to Active Delay from CLK HIGH		250		190		170	ns
t_{FAC}	$\overline{READ}$ or $\overline{WRITE}$ Active from CLK HIGH		200		150		150	ns
t_{FADB}	DB Float to Active Delay from CLK HIGH		300		225		200	ns
t_{HS}	HLDA Valid to CLK HIGH Set-up Time	100		75		75		ns
t_{IDH}	Input Data from $\overline{MEMR}$ HIGH Hold Time	0		0		0		ns
t_{IDS}	Input Data to $\overline{MEMR}$ HIGH Set-up Time	250		190		170		ns
t_{ODH}	Output Data from $\overline{MEMW}$ HIGH Hold Time	20		20		10		ns
t_{ODV}	Output Data Valid to $\overline{MEMW}$ HIGH (Note 13)	200		125		130		ns
t_{QS}	DREQ to CLK LOW (S_1, S_4) Set-up Time	120		90		0		ns
t_{RH}	CLK to READY LOW Hold Time	20		20		20		ns
t_{RS}	READY to CLK LOW Set-up Time	100		60		75		ns
t_{STL}	ADSTB HIGH from CLK HIGH Delay Time		200		150		130	ns
t_{STT}	ADSTB LOW from CLK HIGH Delay Time		140		110		90	ns

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Note 1)
Program Condition (Idle Cycle) (Notes 2, 3, 10 and 11)

Parameters	Description	8237A		8237A-4		8237A-5		Units
		Min	Max	Min	Max	Min	Max	
t_{AR}	ADR Valid or $\overline{CS}$ LOW to $\overline{READ}$ LOW	50		50		50		ns
t_{AW}	ADR Valid to $\overline{WRITE}$ HIGH Set-up Time	200		150		130		ns
t_{CW}	CS LOW to $\overline{WRITE}$ HIGH Set-up Time	200		150		130		ns
t_{DW}	Data Valid to $\overline{WRITE}$ HIGH Set-up Time	200		150		130		ns
t_{RA}	ADR or $\overline{CS}$ Hold from $\overline{READ}$ HIGH	0		0		0		ns
t_{RDE}	Data Access from $\overline{READ}$ LOW (Note 8)		200		200		140	ns
t_{DRF}	DB Float Delay from $\overline{READ}$ HIGH	20	100	20	100	0	70	ns
t_{RSTD}	Power Supply HIGH to RESET LOW Set-up Time	500		500		500		μs
t_{RSTS}	RESET to First $\overline{IOWR}$	2TCY		2TCY		2TCY		ns
t_{RSTW}	RESET Pulse Width	300		300		300		ns
t_{RW}	$\overline{READ}$ Width	300		250		200		ns
t_{WA}	ADR from $\overline{WRITE}$ HIGH Hold Time	20		20		20		ns
t_{WC}	$\overline{CS}$ HIGH from $\overline{WRITE}$ HIGH Hold Time	20		20		20		ns
t_{WD}	Data from $\overline{WRITE}$ HIGH Hold Time	30		30		30		ns
t_{WWS}	Write Width	200		200		160		ns

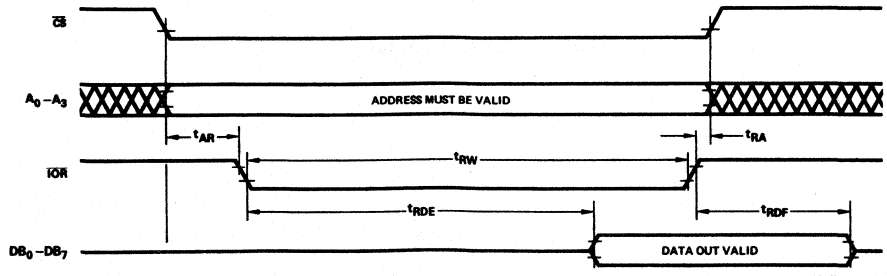
SWITCHING WAVEFORMS

Timing Diagram 1. Program Condition Write Timing (Note 9)



WF006470

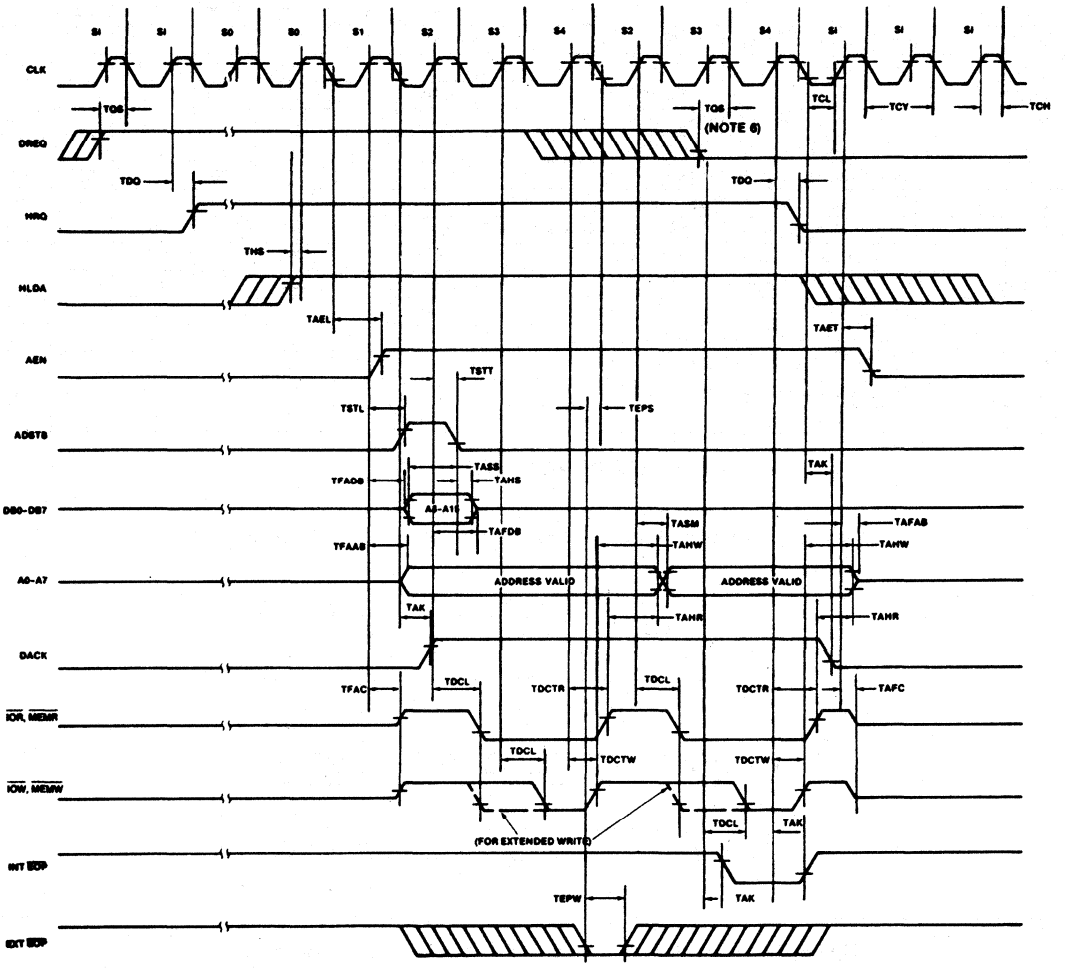
Timing Diagram 2. Program Condition Read Cycle (Note 9)



WF006480

SWITCHING WAVEFORMS (Continued)

DMA TRANSFER TIMING



WF009550

Figure 9. DMA Transfer

SWITCHING WAVEFORMS (Continued)

MEMORY-TO-MEMORY TRANSFER TIMING

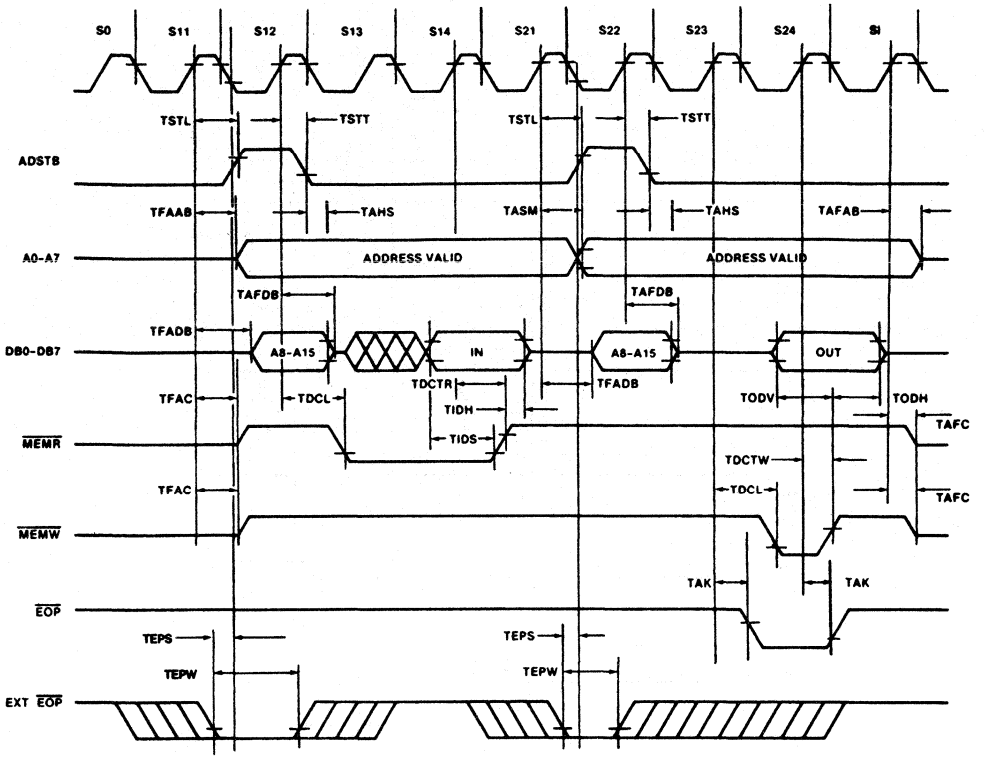


Figure 10. Memory-to-Memory Transfer

READY TIMING

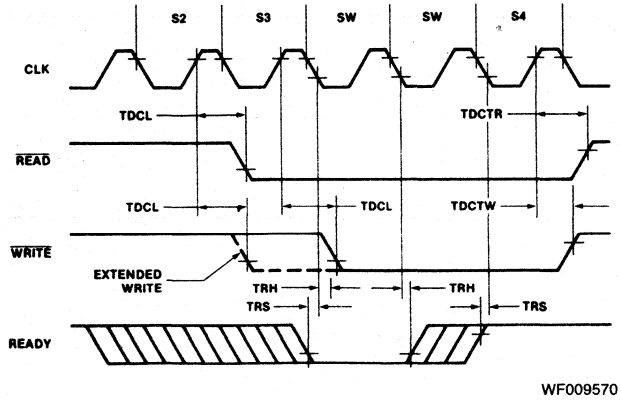


Figure 11. Ready

SWITCHING WAVEFORMS (Continued)

COMPRESSED TRANSFER TIMING

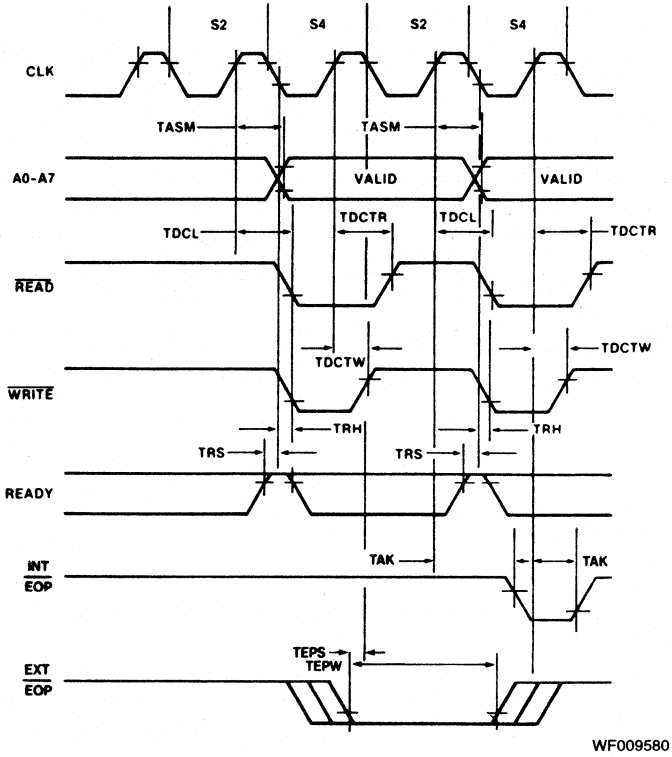


Figure 12. Compressed Transfer

RESET TIMING

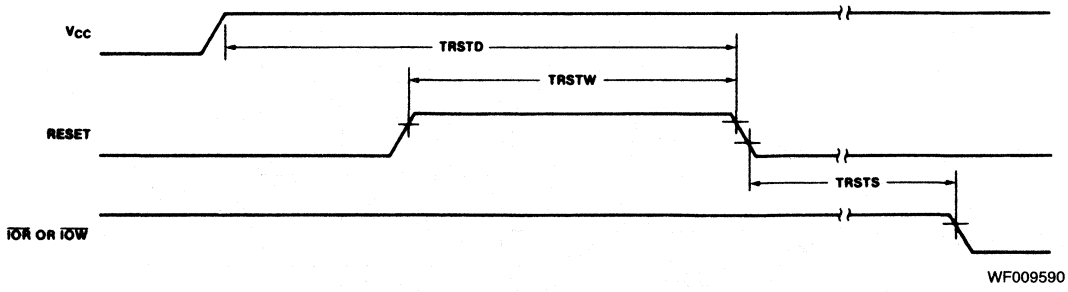


Figure 13. Reset

8251/Am9551

Programmable Communication Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Separate control and transmit register input buffers
- Synchronous or asynchronous serial data transfer
- Parity, overrun and framing errors detected
- Half or full duplex signalling
- Character length of 5, 6, 7 or 8 bits
- Internal or external synchronization
- Odd parity, even parity or no parity bit
- Modem interface controlled by processor
  - Programmable Sync pattern
  - Fully TTL compatible logic levels

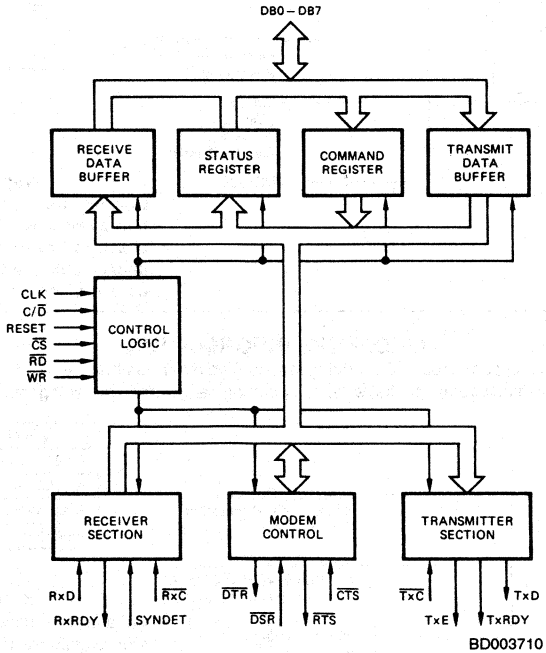
GENERAL DESCRIPTION

The 8251/Am9551 is a programmable serial data communication interface that provides a Universal Synchronous/Asynchronous Receiver/Transmitter (USART) function. It is normally used as a peripheral device for an associated processor and may be programmed by the processor to operate in a variety of standard serial communication formats.

The device accepts parallel data from the CPU, formats and serializes the information based on its current operating mode, and then transmits the data as a serial bit stream. Simultaneously, serial data can be received, converted into parallel form, deformatted, and then presented to the CPU. The USART can operate in an independent full duplex mode.

Data, Control, operation and format options are all selected by commands from an associated processor. This provides an unusual degree of flexibility and allows the 8251/Am9551 to service a wide range of communication disciplines and applications.

BLOCK DIAGRAM

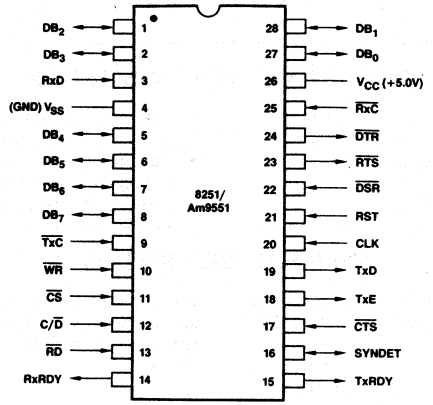


02334B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-28, P-28

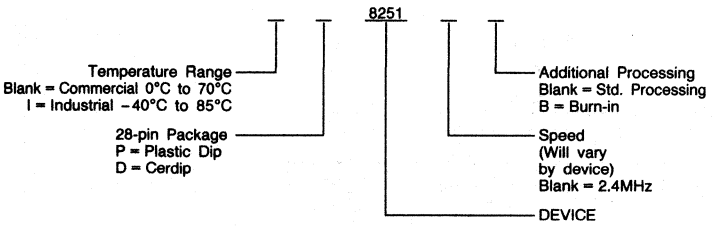


CD005481

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



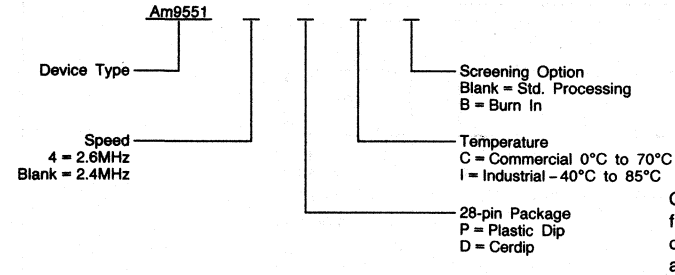
Valid Combinations	
8251	P, D, ID
8251B	P, D, ID
8251	/BXA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9551	PC, DC, DI, DIB
Am9551-4	PC, DC, DI, DIB
Am9551-4	/BXA
Am9551	/BXA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
27, 28, 1, 2, 5-8	Data Bus (DB_0-DB_7)	I/O	The Am9551 uses an 8-bit bidirectional data bus to exchange information with an associated processor. Internally, data is routed between the data bus buffers and the transmitter section or receiver section as selected by the Read ($\overline{RD}$) or Write ($\overline{WR}$) control inputs.
11	Chip Select ($\overline{CS}$)	I	The active low Chip Select input allows the Am9551 to be individually selected from other devices within its address range. When Chip Select is HIGH, reading or writing is inhibited, and the data bus output is in its high-impedance state.
21	Reset (RST)	I	The Am9551 will assume an Idle state when a high level is applied to the Reset input. When the Reset is returned LOW, the Am9551 will remain in the Idle state until it receives a new mode control instruction.
13	Read ($\overline{RD}$)	I	The active low Read input enables data to be transferred from the Am9551 to the processor.
10	Write ($\overline{WR}$)	I	The active low Write input enables data to be transferred from the processor to the Am9551.
12	Control/Data ($C\overline{D}$)	I	During a Read operation, if this input is at a high level, the status byte will be read, and if it is at a low level, the receive data will be read by the processor. When a Write operation is being performed, this input will indicate to the Am9551 that the bus information being written is a command if C/$\overline{D}$ is HIGH and data if C/$\overline{D}$ is LOW.

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	Am9551 DATA → DATA BUS
0	1	0	0	DATA BUS → Am9551 DATA
1	0	1	0	Am9551 STATUS → DATA BUS
1	1	0	0	DATA BUS → Am9551 COMMAND
X	X	X	1	DATA BUS→ THREE-STATE

20	Clock (CLK)	I	This input is used for internal timing within the Am9551. It does not control the transmit or receive rate. However, it should be at least 30 times the receive or transmit rate in the synchronous mode and 4.5 times the receive or transmit rate in the asynchronous mode. The CLK frequency is also restricted by both an upper and a lower bound. This input is often connected to a clock from the associated processor.
3	Receiver Data (RxD)	I	Serial data is received from the communication line on this input.
25	Receiver Clock ($\overline{RxC}$)	I	The serial data on input RxD is clocked into the Am9551 by the $\overline{RxC}$ clock signal. In the synchronous mode, $\overline{RxC}$ is determined by the baud rate and supplied by the modem. In the asynchronous mode, $\overline{RxC}$ is 1, 16, or 64 times the baud rate as selected in the mode control instruction. Data is sampled by the Am9551 on the rising edge or $\overline{RxC}$.
14	Receiver Ready (RxRDY)	O	The RxRDY output signal indicates to the processor that data has been shifted into the receiver buffer from the receiver section and may be read. The signal is active high and will be reset when the buffer is read by the processor. RxRDY can be activated only if the receiver enable (RxE) has been set in the command register, even though the receiver may be running. If the processor does not read the receiver buffer before the next character is shifted from the receiver section, then an overrun error will be indicated in the status buffer.
16	Sync Detect (SYNDET)	I/O	This signal is used only in the synchronous mode. It can be either an output or input depending on whether the program is set for internal or external synchronization. As an output, a high level indicates when the sync character has been detected in the received data stream after the Internal Synchronization mode has been programmed. If the Am9551 is programmed to utilize two sync characters, then SYNDET will go to a high level when the last bit of the second sync character is received. SYNDET is reset when the status buffer is read or when a Reset signal is activated. SYNDET will perform as an input when the External Synchronization mode is programmed. External logic can supply a positive-going signal to indicate to the Am9551 that synchronization has been attained. This will cause it to initialize the assembly of characters on the next falling edge of $\overline{RxC}$. To successfully achieve synchronization, the SYNDET signal should be maintained in a high condition for at least one full period of $\overline{RxC}$.
19	Transmit Data (TxD)	O	Serial data is transmitted to the communication line on this output.
9	Transmitter Clock ($\overline{TxC}$)	O	The serial data on TxD is clocked out with the $\overline{TxC}$ signal. The relationship between clock rate and baud rate is similar to that for RxC. Data is shifted out of the Am9551 on the falling edge of $\overline{TxC}$.
15	Transmitter Ready (TxRDY)	O	The TxRDY output signal goes HIGH when data in the Transmit Data Buffer has been shifted into the transmitter section allowing the Transmit Data Buffer to accept the next byte from the processor. TxRDY will be reset when information is written into the Transmit Data Buffer. Loading command register also resets TxRDY. TxRDY will be available on this output pin only when the Am9551 is enabled to transmit (CTS = 0, TxEN = 1). However, the TxRDY bit in the status Buffer will always be set when the Transmit Data Buffer is empty regardless of the state of TxEN and CTS.
18	Transmitter Empty (TxE)	O	The TxE output signal goes HIGH when the Transmitter section has transmitted its data and is empty. The signal will remain HIGH until a new data byte is shifted from the Transmit Data Buffer to the Transmitter section. In the synchronous mode if the processor does not load a new byte into the buffer in time, TxE will, independent of the status of the TxEN bit in the command register, momentarily go to a high level as SYNC characters are loaded into the Transmitter Section.
24	Data Terminal Ready ($\overline{DTR}$)	O	This signal is a general purpose output which reflects the state of bit 1 in the Command instruction. It is commonly connected to an associated modem to indicate that the Am9551 is ready.
22	Data Set Ready ($\overline{DSR}$)	I	This is a general purpose input signal and forms part of the status byte that may be read by the processor. DSR is generally used as a response to DTR, by the Modem, to indicate that it is ready. The signal acts only as a flag and does not control any internal logic.
23	Request to Send ($\overline{RTS}$)	O	This is a general purpose output, similar to $\overline{DTR}$, and reflects the state of bit 5 in Command Instruction. It is normally used to initiate a data transmission by requesting the modem to prepare to send.
17	Clear to Send ($\overline{CTS}$)	I	This is a general purpose input signal used to enable the 8251/Am9551 to transmit data if the TxEN bit in the Command byte is a one. $\overline{CTS}$ is generally used as a response to RTS by a modem to indicate that transmission may begin. Designers not using $\overline{CTS}$ in their systems should remember to tie it LOW so that 8251/Am9551 data transmission will not be disabled.

02334B

Refer to page 7-1 for Essential Information on Military Devices

PROGRAMMING INFORMATION

The microcomputer program controlling the Am9551 performs these tasks:

- Outputs control codes
- Inputs status
- Outputs data to be transmitted
- Inputs data which have been received

Control codes determine the mode in which the Am9551 will operate and are used to set or reset control signals output by the Am9551.

The Status register contents will be read by the program monitoring this device's operation to determine error conditions and when and how to read data, write data or output control codes. Program logic may be based on reading status bit levels, or control signals may be used to request interrupts.

Initializing the Am9551

The Am9551 may be initialized following a system reset or prior to starting a new serial I/O sequence. The USART must be reset following power up and subsequently may be reset at any time following completion of one activity and preceding a new set of operations. Following a reset, the Am9551 enters an Idle state in which it can neither transmit nor receive data.

The Am9551 is initialized with two, three or four control words from the processor. Figure 1 shows the sequence of control words needed to initialize the Am9551, for synchronous or for asynchronous operation. Note that in asynchronous operation a mode control is output to the device followed by a command. For synchronous operation, the mode control is followed by one or two SYNC characters and then a command.

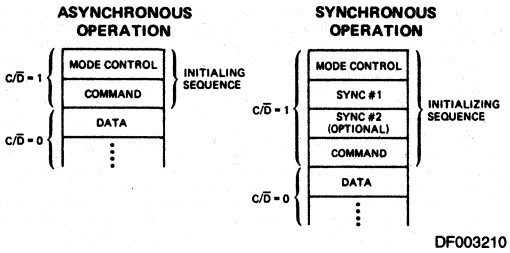


Figure 1. Control Word Sequence for Initialization

Only a single address is set aside for mode control bytes, command bytes and SYNC character bytes. For this to be possible, logic internal to the chip directs control information to its proper destination based on the sequence in which it is received. Following a reset, the first control code output is interpreted as a mode control. If the mode control specifies synchronous operation, then the next one or two bytes (as determined by the mode byte) output as control codes will be interpreted as SYNC characters. For either asynchronous or synchronous operation, the next byte output as a control code is interpreted as command. All subsequent bytes output as control codes are interpreted as commands. There are two ways in which control logic may return to anticipating a mode control input: either following an external Reset signal or following an internal Reset command.

MODE CONTROL CODES

The Am9551 interprets mode control codes as illustrated in Figures 2 and 3.

Control code bits 0 and 1 determine whether synchronous or asynchronous operation is specified. A non-zero value in bits 0 and 1 specifies asynchronous operation and defines the relationship between data transfer baud rate and receiver or transmitter clock rate. Asynchronous serial data may be received or transmitted on every clock pulse, on every 16th clock pulse, or on every 64th clock pulse. A zero in both bits 0 and 1 defines the mode of operation as synchronous.

For synchronous and asynchronous modes, control bits 2 and 3 determine the number of data bits which will be present in each data character.

For synchronous and asynchronous modes, bits 4 and 5 determine whether there will be a parity bit in each character, and if so, whether odd or even parity will be adopted. Thus in synchronous mode a character will consist of five, six, seven or eight data bits, plus an optional parity bit. In asynchronous mode, the data unit will consist of five, six, seven, or eight data bits, an optional parity bit, a preceding start bit, plus 1, 1½, or 2 trailing stop bits. Interpretation of subsequent bits differs for synchronous or asynchronous modes.

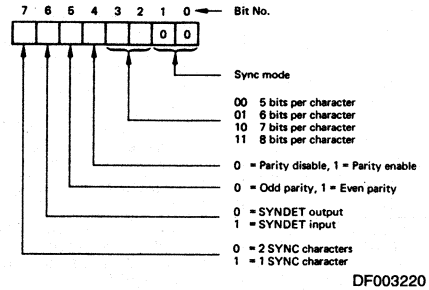


Figure 2. Synchronous Mode Control Code

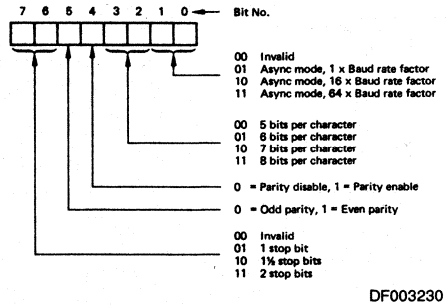


Figure 3. Asynchronous Mode Control Code

Control code bits 6 and 7 in asynchronous mode determine how many stop bits will trail each data unit. 1½ stop bits can only be specified with a 16x or 64x baud rate factor. In these two cases, the half stop bit will be equivalent to 8 or 32 clock pulses, respectively.

In synchronous mode, control bits 6 and 7 determine how character synchronization will be achieved. When SYNDET is an output, internal synchronization is specified; one or two SYNC characters, as specified by control bit 7, must be detected at the head of a data stream to establish synchronization.

Command Words

Command words are used to initiate specific functions within the Am9551, such as "reset all error flags" or "start searching for sync." Consequently, Command Words may be issued by

the micro-processor to the Am9551 at any time during the execution of a program in which specific functions are to be initiated within the communication circuit.

Figure 4 shows the format for the Command Word.

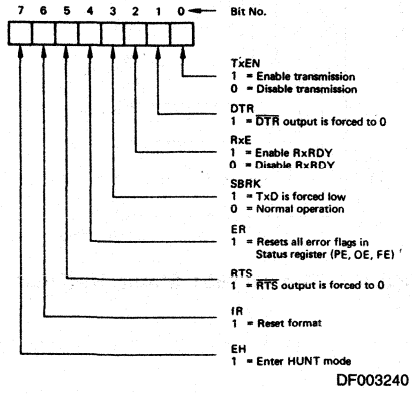


Figure 4. Am9551 Control Command

Bit 0 of the Command Word is the Transmit Enable bit (TxEN). Data transmission from the Am9551 cannot take place unless TxEN is set in the command register. Figure 5 defines the way in which TxEN, TxE and TxRDY combine to control transmitter operations.

Bit 1 is the Data Terminal Ready (DTR) bit. When the DTR command bit is set, the $\overline{DTR}$ output connection is active (low). DTR is used to advise a modem that the data terminal is prepared to accept or transmit data.

Bit 2 is the Receiver Enable Command bit (RxE). RxE is used to enable the RxRDY output signal. RxE prevents the RxRDY signal from being generated to notify the processor that a complete character is framed in the Receive Character Buffer. It does not inhibit the assembly of data characters at the input, however. Consequently, if communication circuits are active, characters will be assembled by the receiver and transferred to the Receiver Character Buffer. If RxE is disabled, the overrun error (OE) will probably be set; to insure proper operation, the overrun error is usually reset with the same command that enables RxE.

Bit 3 is the Send Break Command bit (SBRK). When SBRK is set, the transmitter output (TxD) is interrupted and a continuous binary "0" level (spacing) is applied to the TxD output signal. The break will continue until a subsequent Command Word is sent to the Am9551 to remove SBRK.

Bit 4 is the Error Reset bit (ER). When a Command Word is transmitted with the ER bit set, all three error flags in the Status Register are reset. Error Reset occurs when the Command Word is loaded into the Am9551. No latch is provided in the Command Register to save the ER command bit.

Bit 5, the Request To Send Command bit (RTS), sets a latch to reflect the RTS signal level. The output of this latch is created independently of other signals in the Am9551. As a result, data transfers may be made by the microprocessor to the Transmit Register, and data may be actively transmitted to the communication line through TxD regardless of the status of RTS.

Bit 6, the Internal Reset (IR), causes the Am9551 to return to the Idle mode. All functions within the Am9551 cease and no new operation can be resumed until the circuit is reinitialized. If the operating mode is to be altered during the execution of a microprocessor program, the Am9551 must first be reset. Either the external reset connection can be activated, or the Internal Reset Command can be sent to the Am9551. Internal Reset is a momentary function performed only when the command is issued.

Bit 7 is the Enter Hunt command bit (EH). The Enter Hunt mode command is only effective for the Am9551 when it is operating in the Synchronous mode. EH causes the receiver to stop assembling characters at the RxD input and start searching for the prescribed sync pattern. Once the "Enter Hunt" mode has been initiated, the search for the sync pattern will continue indefinitely until EH is reset when a subsequent Command Word is sent, when the IR command is sent to the Am9551, or when SYNC characters are recognized.

TxEN	Txe	TxRDY	
1	1	1	Transmit Output Register and Transmit Character Buffer empty. TxD continues to mark if Am9551 is in the asynchronous mode. TxD will send Sync pattern if Am9551 is in the Synchronous Mode. Data can be entered into Buffer.
1	0	1	Transmit Output Register is shifting a character. Transmit Character Buffer is available to receive a new byte from the processor.
1	1	0	Transmit Register has finished sending. A new character is waiting for transmission. This is a transient condition.
1	0	0	Transmit Register is currently sending and an additional character is stored in the Transmit Character Buffer for transmission.
0	0/1	0/1	Transmitter is disabled.

Figure 5. Operation of the Transmitter Section as a Function of TxE, TxRDY and TxEN

Status Register

The Status Register maintains information about the current operational status of the Am9551. Figure 6 shows the fomat of the Status Register.

TxRDY signals the processor that the Transmit Character Buffer is empty and that the Am9551 can accept a new character for transmission.

RxRDY signals the processor that a completed character is holding in the Receive Character Buffer Register for transfer to the processor.

TxE signals the processor that the Transmit Register is empty.

PE is the Parity Error signal indicating to the CPU that the character stored in the Receive Character Buffer was received with an incorrect number of binary "1" bits.

OE is the receiver Overrun Error. OE is set whenever a byte stored in the Receiver Character Register is overwritten with a new byte before being transferred to the processor.

FE is the character framing error which indicates that the asynchronous mode byte stored in the Receiver Character Buffer was received with incorrect character bit format as specified by the current mode.

SYNDET is the synchronous mode status bit associated with internal sync detection.

DSR is the status bit set by the external Data Set Ready signal to indicate that the communication Data Set is operational. All status bits are set by the functions described for them. SYNDET is reset whenever the processor reads the Status Register. OE, FE, PE are reset only by command.

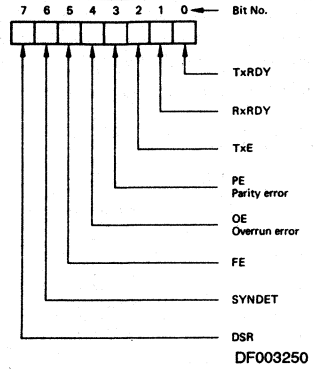


Figure 6. The Am9551 Status Register

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages
with Respect to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 1.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8251 Am9551 Am9551-4	0°C to 70°C	5V ±5%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range (Note 1)

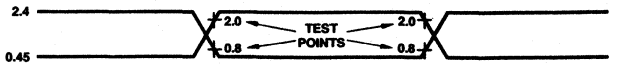
Parameters	Description	Test Conditions	8251			Am9551			Units
			Min	Typ	Max	Min	Typ	Max	
V_{OH}	Output HIGH Voltage	$I_{OH} = -200\mu A$				2.4			Volts
		$I_{OH} = -100\mu A$	2.4						
V_{OL}	Output LOW Voltage	$I_{OL} = 3.2mA$						0.45	Volts
		$I_{OL} = 1.6mA$			0.45				
V_{IH}	Input HIGH Voltage		2.2		V_{CC}	2.2		V_{CC}	Volts
V_{IL}	Input LOW Voltage		−0.5		0.8	−0.5		0.8	Volts
I_{LI}	Input Load Current	$V_{SS} \leq V_{IN} \leq V_{CC}$			10			10	μA
I_{DL}	Data Bus Leakage	$V_{OUT} = 0.45V$			−50			−50	μA
		$V_{OUT} = V_{CC}$			10			10	
I_{CC}	V_{CC} Supply Current	$T_A = +25°C$		45			45		mA
		$T_A = 0°C$			80			80	
C_O	Output Capacitance	fc = 1.0MHz, Inputs = 0V						15	pF
C_I	Input Capacitance				10			10	pF
$C_{I/O}$	I/O Capacitance				20			20	pF

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Note 2)

Parameters	Description		8251 Min	8251 Max	Am9551 Min	Am9551 Max	Am9551-4 Min	Am9551-4 Max	Units
t_{AR}	$\overline{CS}$, C/$\overline{D}$ Stable to $\overline{READ}$ Low Set-up Time		50		50		50		ns
t_{AW}	$\overline{CS}$, C/$\overline{D}$ Stable to $\overline{WRITE}$ Low Set-up Time		20		20		20		ns
t_{CR}	$\overline{DSR}$, $\overline{DTS}$ to $\overline{READ}$ Low Set-up Time			16		16		16	tCY
t_{CY}	Clock Period		.420	1.35	.380	1.35	.380	1.35	µs
t_{DF}	$\overline{READ}$ High to Data Bus Off Delay		25	200	25	200	25	200	ns
t_{DTx}	TxC Low to TxD Delay			1.0		1.0		1.0	µs
t_{DW}	Data to $\overline{WRITE}$ High Set-up Time		200		150		100		ns
t_{ES}	External SYNDET to RxC Low Set-up Time			16		16		16	tCY
t_{HRx}	Sampling Pulse to Rx Data Hold Time		2.0		2.0		2.0		µs
t_{IS}	Data Bit (Center) to Internal SYNDET Delay			25		25		25	tCY
$t_{\phi W}$	Clock Pulse Width		220	0.7tCY	175	0.7tCY	175	0.7tCY	ns
t_R, t_F	Clock Rise & Fall Time		0	50	0	50	0	50	ns
t_{RA}	$\overline{READ}$ High to $\overline{CS}$, C/D Hold Time		5.0		5.0		5.0		ns
t_{RD}	$\overline{READ}$ Low to Data Bus On Delay			350		250		180	ns
t_{RPD}	Receiver Clock High Time	1xBaud Rate	15		15		15		tCY
		16x & 64x Baud Rate	3.0		3.0		3.0		
t_{RPW}	Receiver Clock Low Time	1x Baud Rate	12		12		12		tCY
		16x & 64x Baud Rate	1.0		1.0		1.0		
t_{RR}	$\overline{READ}$ Pulse Width		430		380		250		ns
t_{RV}	Time Between $\overline{WRITE}$ Pulses During Initialization (Note 3)		6.0		6.0		6.0		tCY
t_{Rx}	Data Bit (Center) to RxRDY Delay			20		20		20	tCY
t_{SRx}	Rx Data to Sampling Pulse Set-up Time		2.0		2.0		2.0		µs
t_{TPD}	Transmitter Clock High Time	1x Baud Rate	15		15		15		tCY
		16x & 64x Baud Rate	3.0		3.0		3.0		
t_{TPW}	Transmitter Clock Low Time	1x Baud Rate	12		12		12		tCY
		16x & 64x Baud Rate	1.0		1.0		1.0		
t_{Tx}	Data Bit (Center) to TxRDY Delay			16		16		16	tCY
t_{TxE}	Data Bit (Center) to Tx EMPTY Delay			16		16		16	tCY
t_{WA}	$\overline{WRITE}$ High to $\overline{CS}$, C/D Hold Time		20		20		20		ns
t_{WC}	$\overline{WRITE}$ High to TxE, $\overline{DTR}$, $\overline{RTS}$ Delay			16		16		16	tCY
t_{WD}	$\overline{WRITE}$ High to Data Hold Time		40		40		40		ns
t_{WW}	$\overline{WRITE}$ Pulse Width		400		380		250		ns
f_{Rx}	Receiver Clock Frequency	1x Baud Rate	DC	56	DC	56	DC	56	kHz
		16x & 64x Baud Rate	DC	520	DC	520	DC	520	
f_{Tx}	Transmitter Clock Frequency	1x Baud Rate	DC	56	DC	56	DC	56	kHz
		16x & 64x Baud Rate	DC	520	DC	520	DC	520	

Notes:
1. Typical values are for $T_A = 25°C$, nominal supply voltage and nominal processing parameters.
2. Test conditions include: transition times ≤ 20ns, output loading of 1 TTL gate plus 100pF, input and output timing reference levels of 0.8V and 2.0V.
3. This time period between write pulses is specified for initialization purposes only when MODE, SYNC 1, SYNC 2, COMMAND and first DATA BYTE are written into the Am9551. Subsequent writing of both COMMAND and DATA are only allowed when TxRDY = 1.
4. Reset Pulse Width = 6tCY min.
5. Switching Characteristics parameters are listed in alphabetical order.

SWITCHING TEST INPUT/OUTPUT WAVEFORM

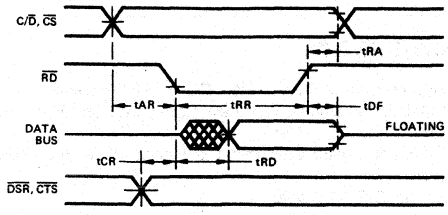


WF006490

AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

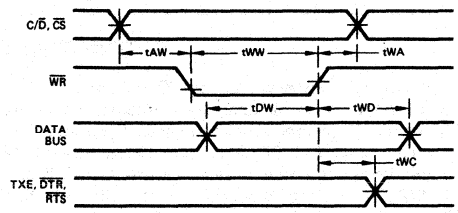
SWITCHING WAVEFORMS

READ OPERATION



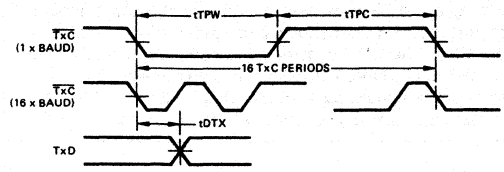
WF006500

WRITE OPERATION



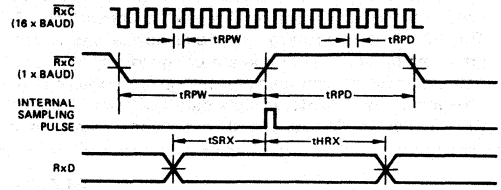
WF006510

TRANSMITTER CLOCK AND DATA



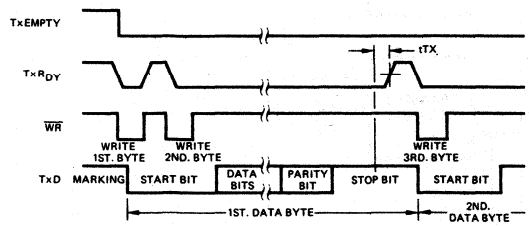
WF006520

RECEIVER CLOCK AND DATA



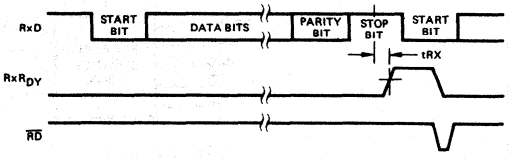
WF006530

TxRD TIMING (ASYNC MODE)



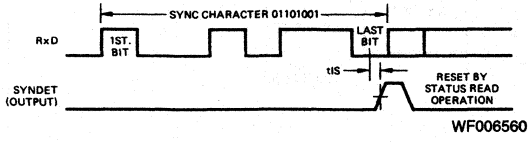
WF006540

RxRDY TIMING (ASYNC MODE)



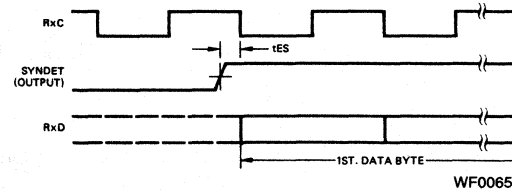
WF006550

INTERNAL SYNC DETECT (SYNC MODE ONLY)



WF006560

EXTERNAL SYNC DETECT (SYNC MODE ONLY)



WF006570

8251A

Programmable Communication Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Synchronous and Asynchronous Operation
- Synchronous 5 – 8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5 – 8 Bit Characters; Clock Rate – 1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1 1/2, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling
- Synchronous Baud Rate – DC to 64K Baud
- Asynchronous Baud Rate – DC to 19.2K Baud
- Full-Duplex, Double-Buffered Transmitter and Receiver
- Error Detection – Parity, Overrun and Framing
- Compatible with an Extended Range of Microprocessors
- 28-Pin DIP Package
- All Inputs and Outputs are TTL Compatible

GENERAL DESCRIPTION

The 8251A is the enhanced version of the industry standard, 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with microprocessor families, such as the iAPX86, 88. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals, such as SYNDET, TxEMPTY. The chip is fabricated using N-channel silicon gate technology.

BLOCK DIAGRAM

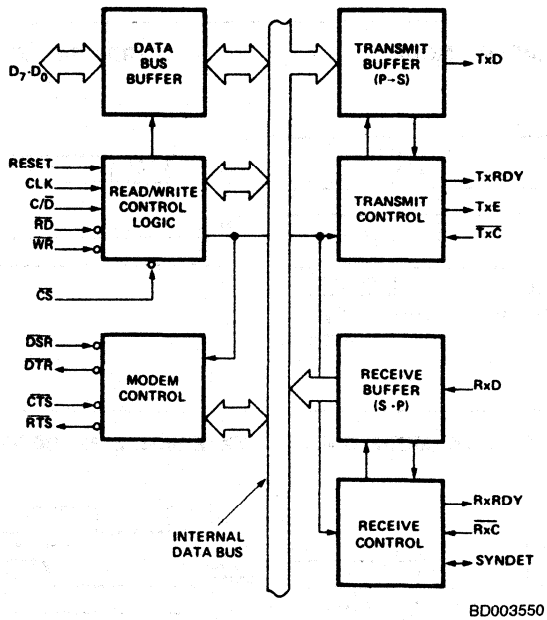


Figure 1.

04133A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

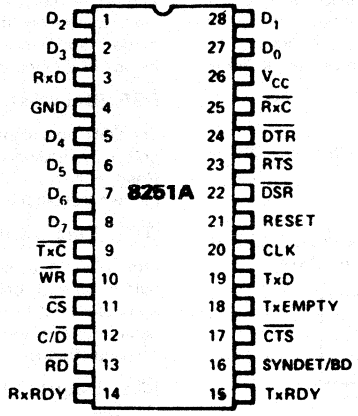
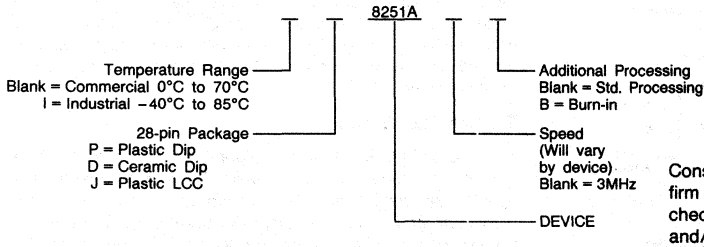


Figure 2.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8251A	P, D, ID
8251AB	P, D, ID
8251A	/BXA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION and RELATED INFORMATION

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "LOW" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "LOW" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

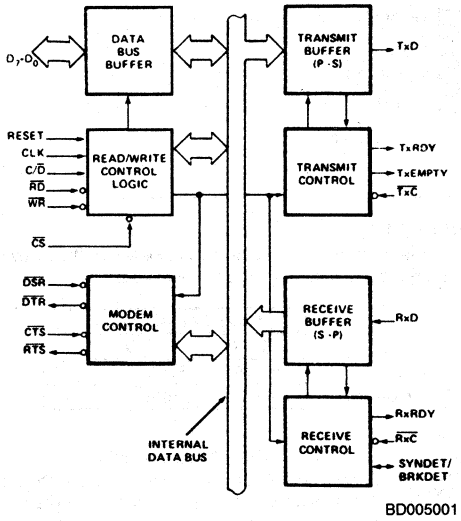


Figure 3. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

C/$\overline{D}$	$\overline{RD}$	$\overline{W}$	$\overline{CS}$	
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → CONTROL
X	1	1	0	DATA BUS → 3 STATE
X	X	X	1	DATA BUS → 3 STATE

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.

1 = CONTROL/STATUS; 0 = DATA.

$\overline{CS}$ (Chip Select)

A "LOW" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus is in the float state and $\overline{RD}$ and $\overline{WR}$ have no effect on the chip.

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is a general-purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "LOW" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "LOW" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "LOW" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable off or $\overline{CTS}$ off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down.

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{TxC}$. The transmitter will begin transmission upon being enabled if $\overline{CTS}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable or $\overline{CTS}$ is off or the transmitter is empty.

Transmitter Control

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by TxEnable; or for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is not masked by TxEnable, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to send, the TxEMPTY output will go "HIGH." It resets upon receiving a character from CPU if the transmitter is enabled. TxEMPTY remains high

when the transmitter is disabled. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplex operational mode.

In the Synchronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers." TxEMPTY does not go LOW when the SYNC characters are being shifted out.

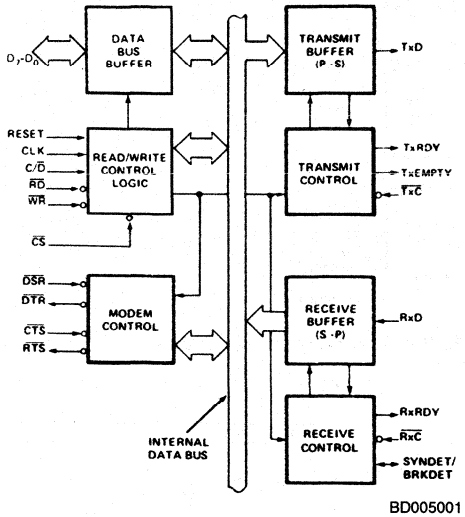


Figure 4. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

$\overline{TxC}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the $\overline{TxC}$ frequency. In Asynchronous transmission mode, the baud rate is a fraction of the actual $\overline{TxC}$ frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{TxC}$.

For Example:

If Baud Rate equals 110 Baud,
$\overline{TxC}$ equals 110 Hz in the 1x mode.
$\overline{TxC}$ equals 1.72 kHz in the 16x mode.
$\overline{TxC}$ equals 7.04 kHz in the 64x mode.

The falling edge of $\overline{TxC}$ shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin and is clocked in on the rising edge of $\overline{RxC}$.

Receiver Control

This functional block manages all receiver-related activities which consist of the following features.

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition." Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid LOW (Start bit) is enabled. This feature is only active in the asynchronous mode and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the Start bit (RxD = LOW).

Parity error detection sets the corresponding status bit.

The Framing Error status bit is set if the Stop bit is absent at the end of the data byte (asynchronous mode).

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or, for polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

RxEnable, when off, holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be enable to sense a Start Bit, and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled, and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error, and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set, and the old character will be lost.

$\overline{RxC}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{RxC}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{RxC}$ frequency. A portion of the mode instruction selects this factor: 1, 1/16 or 1/64 the $\overline{RxC}$.

For example:

Baud Rate equals 300 Baud, if
$\overline{RxC}$ equals 300 Hz in the 1x mode;
$\overline{RxC}$ equals 4800 Hz in the 16x mode;
$\overline{RxC}$ equals 19.2 kHz in the 64x mode.

Baud Rate equals 2400 Baud, if
$\overline{RxC}$ equals 2400 Hz in the 1x mode;
$\overline{RxC}$ equals 38.4 kHz in the 16x mode;
$\overline{RxC}$ equals 153.6 kHz in the 64x mode.

Data is sampled into the 8251A on the rising edge of $\overline{RxC}$.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{TxC}$ and $\overline{RxC}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

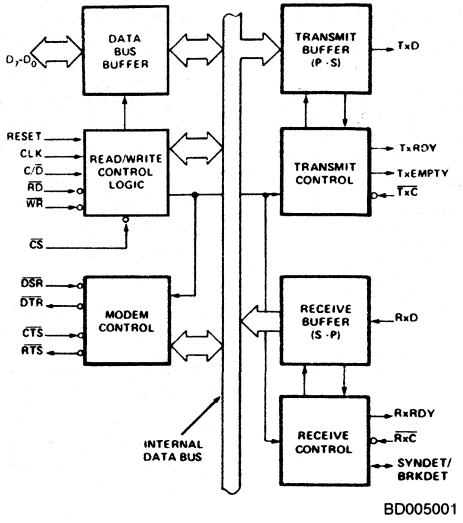


Figure 5. 8251A Block Diagram Showing Receiver Buffer and Control Functions

SYNDET (SYNC Detect/ BRKDET Break Detect)

This pin is used in Synchronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode LOW upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "HIGH" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bisync), then SYNDET will go "HIGH" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "HIGH" input signal can be removed. When External SYNC Detect is programmed, Internal SYNC Detect is disabled.

BREAK (Async Mode Only)

This output will go HIGH whenever the receiver remains LOW through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

FEATURES AND ENHANCEMENTS

The 8251A is an advanced design of the industry standard USART, the 8251. The 8251A operates with an extended range of microprocessors and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

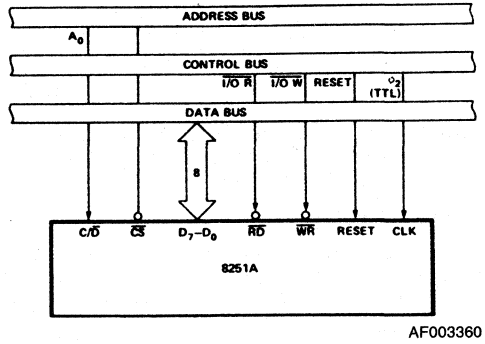


Figure 6. 8251A Interface to 8080A Standard System Bus

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.

- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that, if double character sync is programmed, the characters be contiguously detected and the Rx register cleared to all ones whenever the Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time, but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.

PRODUCT OVERVIEW

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed for a wide range of microcomputers, such as the 8080A, 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support most serial data techniques in use, including IBM "bi-sync."

In a communication environment, an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The Command Status, Data-In and Data-Out registers are separate, 8-bit registers communicating with the system bus through the Data Bus Buffer.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "HIGH" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{cy} (clock must be running).

A command reset operation also puts the device into the "Idle" state.

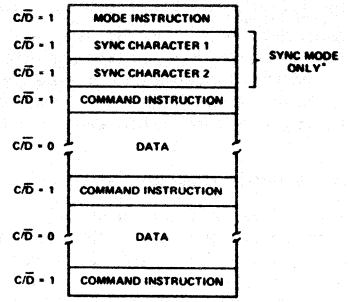
DETAILED DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "HIGH" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "HIGH" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.



DF003920

*THE SECOND SYNC CHARACTER IS SKIPPED IF MODE INSTRUCTION HAS PROGRAMMED THE 8251A TO SINGLE CHARACTER SYNC MODE. BOTH SYNC CHARACTERS ARE SKIPPED IF MODE INSTRUCTION HAS PROGRAMMED THE 8251A TO ASYNC MODE.

Figure 7. Typical Data Block

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words penetrated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

This instruction defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be written.

Command Instruction

This instruction defines a word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation (see Figure 7). The Mode Instruction must be written immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components, one Asynchronous and the other Synchronous, sharing the same package. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE: When parity is enabled, it is not considered as one of the data bits for the purpose of programming the word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU, the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A, the TxD output remains "HIGH" (marking) unless a Break (continuously LOW) has been programmed.

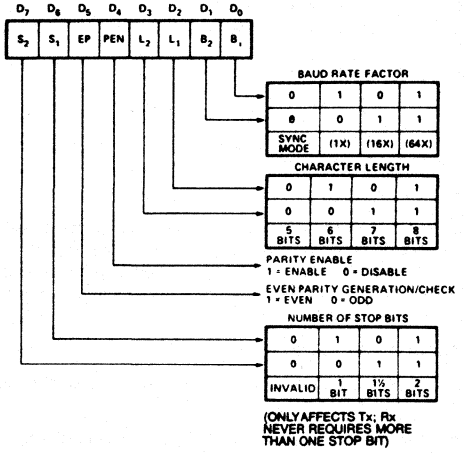
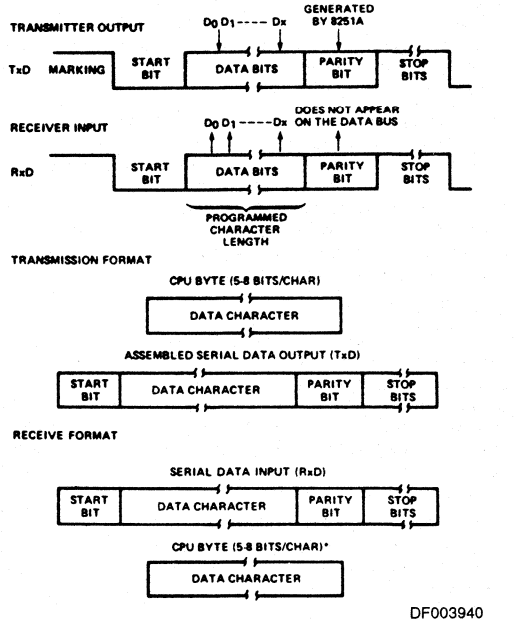


Figure 8. Mode Instruction Format, Asynchronous Mode

Asynchronous Mode (Receive)

The RxD line is normally HIGH. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a LOW is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the STOP bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the receiver requires only one STOP bit, regardless of the number of STOP bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset Instruction. The occurrence of any of these errors will not affect the operation of the 8251A.



*Note: IF CHARACTER LENGTH IS DEFINED AS 5, 6 OR 7 BITS, THE UNUSED BITS ARE SET TO "ZERO."

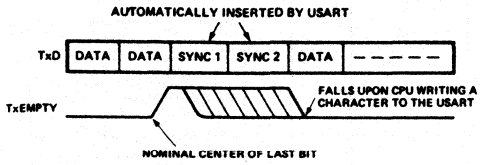
Figure 9. Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously HIGH until the CPU sends its first character to the 8251A which usually is a SYNC character. When the $\overline{CTS}$ line goes LOW, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251A with a data character before the 8251A Transmitter Buffers become empty, the SYNC characters (or character if in single SYNC character mode) will be automatically inserted in the TxD data stream. In this case, the

TxEMPTY pin is raised HIGH to signal that the 8251A is empty and SYNC characters are being sent out. TxEMPTY does not go LOW when the SYNC is being shifted out (see figure below). The TxEMPTY pin is internally reset by a data character being written into the 8251A.



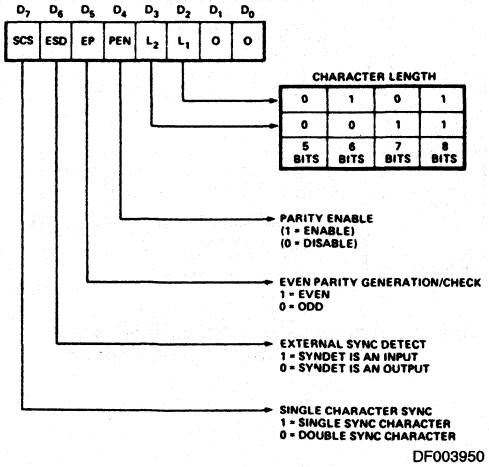
WF008860

Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the SYNC mode has been programmed, ENTER HUNT command should be included in the first comand instruction word written. Data on the RxD pin is then sampled on the rising edge of $\overline{RxC}$. The content of the Rx buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent received character is also compared. When both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set HIGH and is reset automatically by a STATUS READ. If parity is programmed, SYNDET will not be set until the middle of the parity bit instead of the middle of the last data bit.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode. The high level can be removed after one $\overline{RxC}$ cycle. An ENTER HUNT command has no effect in the asynchronous mode of operation.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode. Parity is checked when not in Hunt, regardless of whether the Receiver is enabled or not.

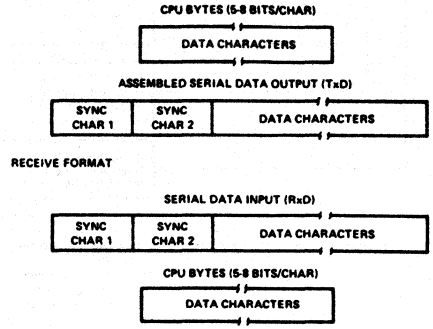


DF003950

Note: IN EXTERNAL SYNC MODE, PROGRAMMING DOUBLE CHARACTER SYNC WILL AFFECT ONLY THE T_X.

Figure 10. Mode Instruction Format, Synchronous Mode

The CPU can command the receiver to enter the HUNT mode if synchronization is lost. This will also set all the used character bits in the buffer to a "one," thus preventing a possible false SYNDET caused by data that happens to be in the Rx Buffer at ENTER HUNT time. Note that the SYNDET F/F is reset at each Status Read, regardless of whether internal or external SYNC has been programmed. This does not cause the 8251A to return to the HUNT mode. When in SYNC mode, but not in HUNT, Sync Detection is still functional, but only occurs at the "known" word boundaries. Thus, if one Status Read indicates SYNDET and a second Status Read also indicates SYNDET, then the programmed SYNDET characters have been received since the previous Status Read. (If double character sync has been programmed, then both sync characters have been contiguously received to gate a SYNDET indication.) When external SYNDET mode is selected, internal Sync Detect is disabled, and the SYNDET F/F may be set at any bit boundary.



DF003960

Figure 11. Data Format, Synchronous Mode

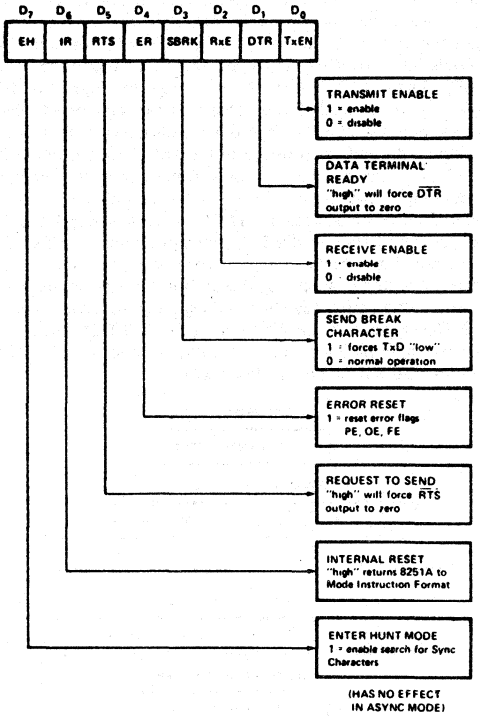
COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the sync characters are loaded (if in Sync Mode), then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions, such as Enable Transmit/Receive, Error Reset and Modem Controls, are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, if necessary, then all further "control writes" ($C/\overline{D} = 1$) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.

Note: Internal Reset on Power-up

When power is first applied, the 8251A may come up in the Mode, Sync character or Command format. To guarantee that the device is in the Command Instruction format before the Reset command is issued, it is safest to execute the worst-case initialization sequence (sync mode with two sync characters). Loading three 00Hs consecutively into the device with $C/\overline{D} = 1$ configures sync operation and writes two dummy 00H sync characters. An Internal Reset command (40H) may then be issued to return the device to the "Idle" state.



Note: Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

Figure 12. Command Instruction Format

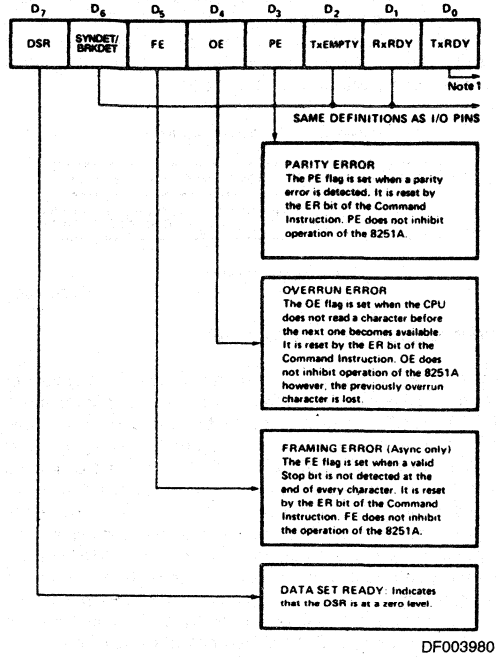
STATUS READ DEFINITION

In data communication systems, it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (Status update is inhibited during status read.)

A normal "read" command is issued by the CPU with C/$\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins, so that the 8251A can be used in a completely polled or interrupt-driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.



Note 1: TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by $\overline{CTS}$ and TxEN; the latter is conditioned by both $\overline{CTS}$ and TxEN. i.e. TxRDY status bit = DB Buffer Empty; TxRDY pin out = DB Buffer Empty • ($\overline{CTS}$ 0) • (TxEN = 1)

Figure 13. Status Read Format

APPLICATIONS INFORMATION

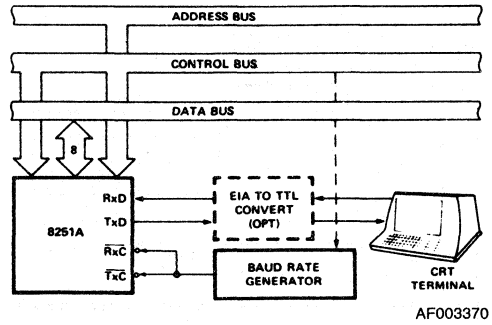


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC – 9600 Baud

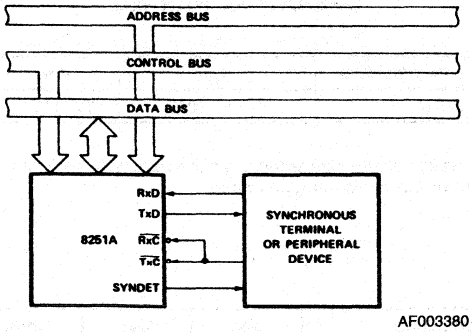


Figure 15. Synchronous Interface to Terminal or Peripheral Device

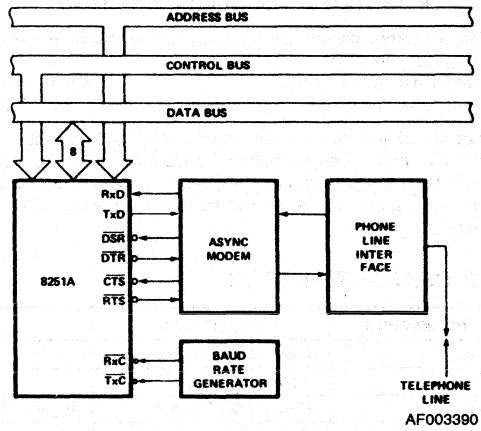


Figure 16. Asynchronous Interface to Telephone Lines

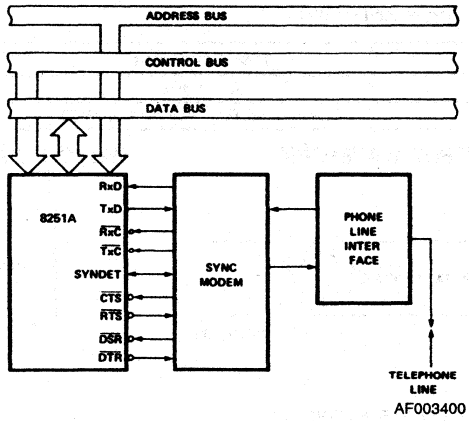


Figure 17. Synchronous Interface to Telephone Lines

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5 to +7V
Power Dissipation ..... 1 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8251A	0°C to 70°C	5V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

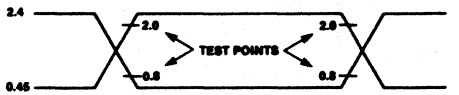
DC CHARACTERISTICS

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL} = 2.2mA		0.45	V
V_{OH}	Output High Voltage	I_{OL} = −400μA	2.4		V
I_{OFL}	Output Float Leakage	V_{OUT} = V_{CC} to 0.45V		±10	μA
I_{IL}	Input Leakage	V_{IN} = V_{CC} to 0.45V		±10	μA
I_{CC}	Power Supply Current	All Outputs = High		100	mA

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Parameters	Description	Test Conditions	Min	Max	Units
C_{IN}	Input Capacitance	fc = 1MHz		10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured Pins Returned to GND		20	pF

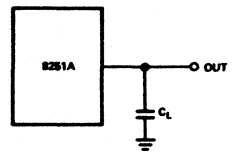
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006190

A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" and 0.8V FOR A LOGIC "0."

SWITCHING LOAD CIRCUIT



TC002030

C_L = 150pF

SWITCHING CHARACTERISTICS

Bus Parameters (Note 1)

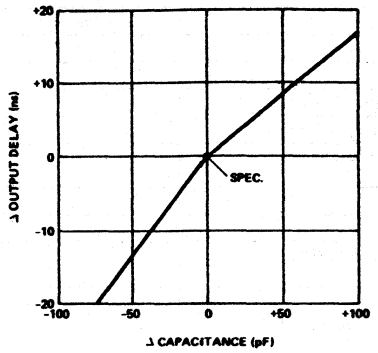
Parameters	Description	Test Conditions	Min	Max	Units
READ CYCLE					
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	Note 2	0		ns
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	Note 2	0		ns
t_{RR}	$\overline{READ}$ Pulse Width		250		ns
t_{RD}	Data Delay from $\overline{READ}$	3, C_L = 150pF		200	ns
t_{DF}	$\overline{READ}$ to Data Floating		10	100	ns
WRITE CYCLE					
t_{AW}	Address Stable Before $\overline{WRITE}$		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$		0		ns
t_{WW}	$\overline{WRITE}$ Pulse Width		250		ns
t_{DW}	Data Setup Time for $\overline{WRITE}$		150		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$		20		ns
t_{RV}	Recovery Time Between WRITES	Note 4	6		t_{CY}

SWITCHING CHARACTERISTICS (Cont.)

Parameters	Description	Test Conditions	Min	Max	Units
OTHER TIMINGS					
t_{CY}	Clock Period	Notes 5, 6	320	1350	ns
t_{Φ}	Clock High Pulse Width		120	t_{CY}–90	ns
$t_{\overline{\Phi}}$	Clock Low Pulse Width		90		ns
t_f t_r	Clock Rise and Fall Time			20	ns
t_{DTX}	TxD Delay from Falling Edge of TxC			1	μs
t_{Tx}	Transmitter Input Clock Frequency 1x Baud Rate		DC	64	kHz
	16x Baud Rate		DC	310	kHz
	64x Baud Rate		DC	615	kHz
t_{TPN}	Transmitter Input Clock Pulse Width 1x Baud Rate		12		t_{CY}
	16x and 64x Baud Rate		1		t_{CY}
t_{TPD}	Transmitter Input Clock Pulse Delay 1x Baud Rate		15		t_{CY}
	16x and 64x Baud Rate		3		t_{CY}
t_{Rx}	Receiver Input Clock Frequency 1x Baud Rate		DC	64	kHz
	16x Baud Rate		DC	310	kHz
	64x Baud Rate		DC	615	kHz
t_{RPW}	Receiver Input Clock Pulse Width 1x Baud Rate		12		t_{CY}
	16x and 64x Baud Rate		1		t_{CY}
t_{RPD}	Receiver Input Clock Pulse Delay 1x Baud Rate		15		t_{CY}
	16x and 64x Baud Rate		3		t_{CY}
t_{TxRDY}	TxRDY Pin Delay from Center of Last Bit	Note 7		14	t_{CY}
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{WR}$	Note 7		400	ns
t_{RxRDY}	RxRDY Pin Delay from Center of Last Bit	Note 7		26	t_{CY}
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{RD}$	Note 7		400	t_{CY}
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$	Note 7		26	t_{CY}
t_{ES}	External SYNDET Set-up Time After Rising Edge of $\overline{RxC}$	Note 7	18		t_{CY}
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit	Note 7		20	t_{CY}
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)	Note 7		8	t_{CY}
t_{CR}	Control to READ Set-up Time ($\overline{DSR}$, $\overline{CTS}$)	Note 7	20		t_{CY}

Notes:
1. AC timings measured V_{OH} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.
2. Chip Select ($\overline{CS}$) and Command/Data (C/$\overline{D}$) are considered as Addresses.
3. Assumes that Address is valid before $R_D\downarrow$.
4. This recovery time is for after a Mode Instruction only. Write Data is allowed only when TxRDY = 1. Recovery time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.
5. The TxC and RxC frequencies have the following limitations with respect to CLK: for 1x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30\ t_{CY})$: for 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5\ t_{CY})$.
6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

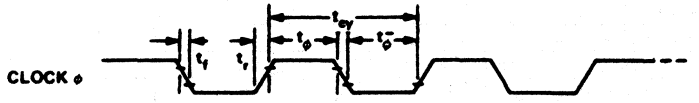
TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE (pF)



OP006180

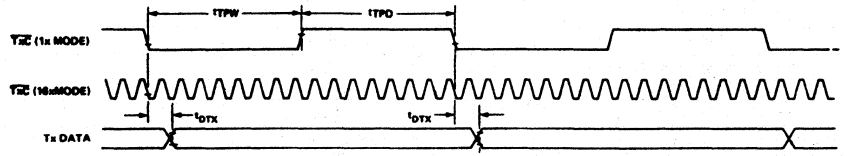
SWITCHING WAVEFORMS

SYSTEM CLOCK INPUT



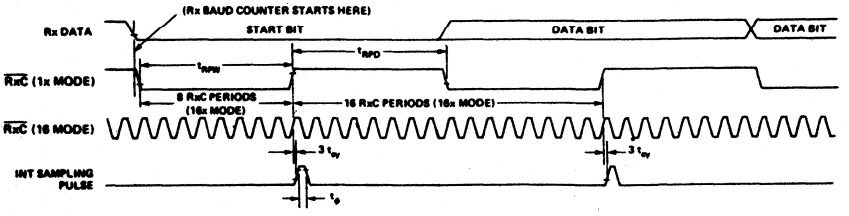
WF008150

TRANSMITTER CLOCK AND DATA



WF006200

RECEIVER CLOCK AND DATA



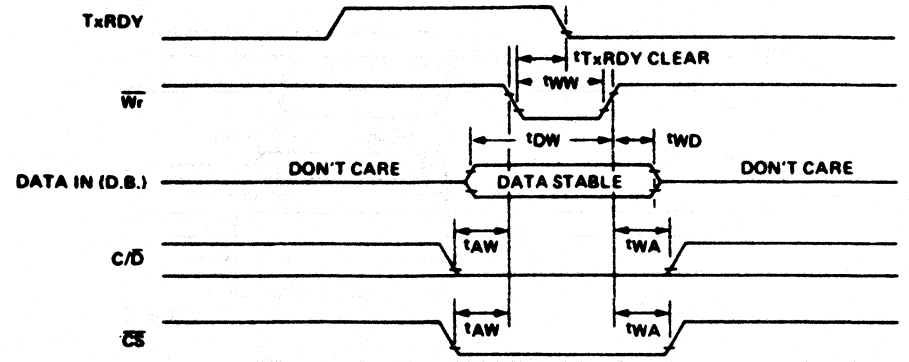
WF008140

04133A

Refer to page 7-1 for Essential Information on Military Devices

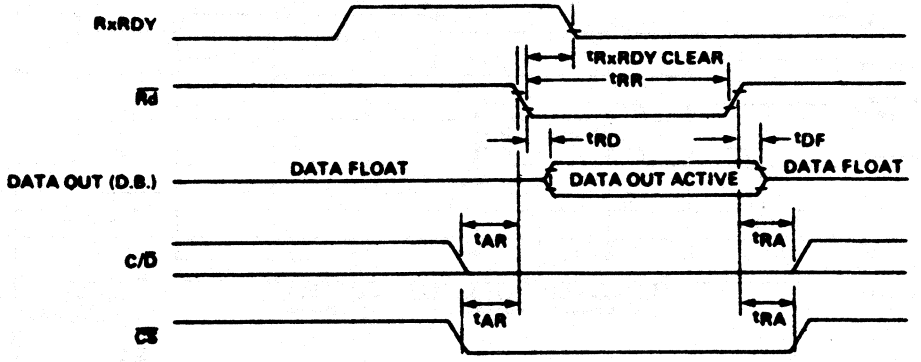
SWITCHING WAVEFORMS (Continued)

WRITE DATA CYCLE (CPU → USART)



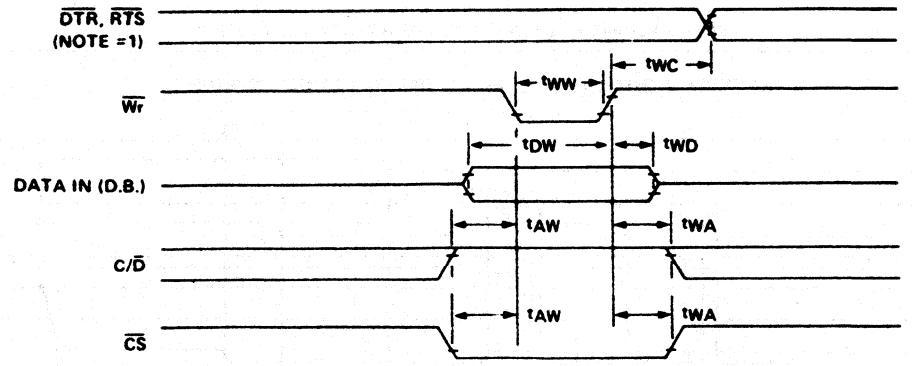
WF006140

READ DATA CYCLE CPU ← USART)



WF006150

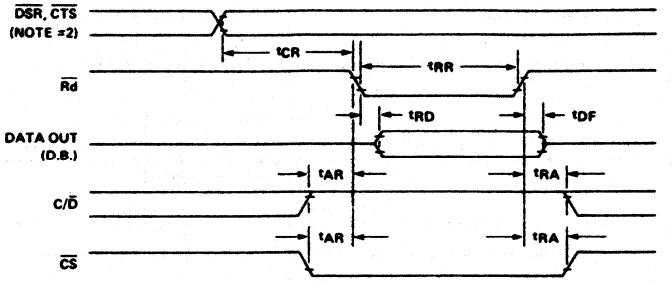
WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → USART)



WF006160

SWITCHING WAVEFORMS (Continued)

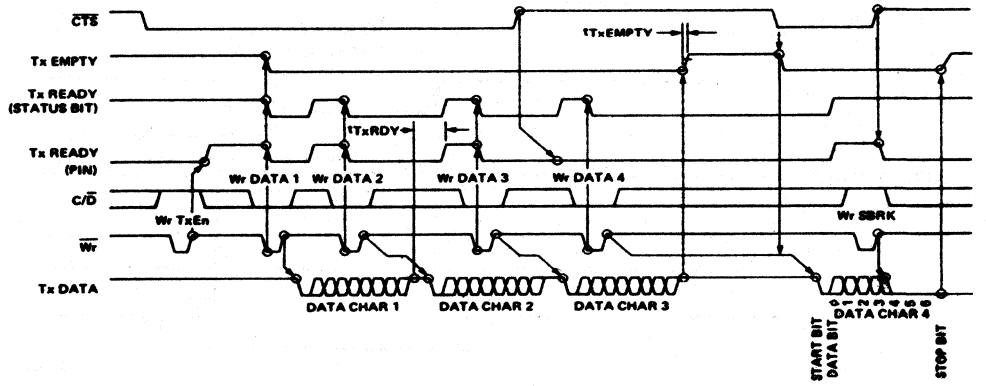
READ CONTROL OR INPUT PORT (CPU ← USART)



WF006170

Notes: 1. T_{WC} INCLUDES THE RESPONSE TIMING OF A CONTROL BYTE.
2. T_{CR} INCLUDES THE EFFECT OF CTS ON THE TxENBL CIRCUITY.

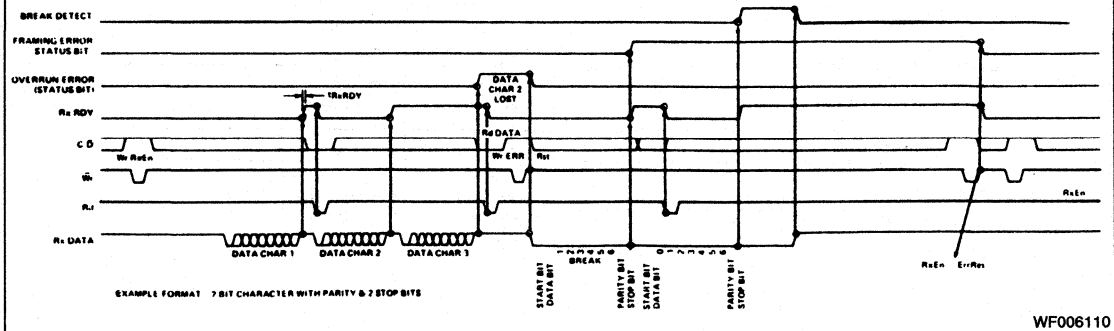
TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)



WF006180

EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY & 2 STOP BITS.

RECEIVER CONTROL AND FLAG TIMING (ASYNC MODE)



WF006110

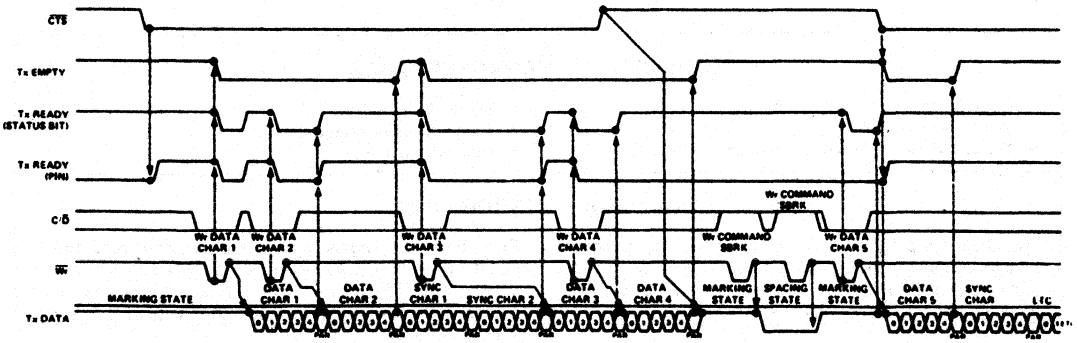
EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY 2 & STOP BITS.

04133A

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING WAVEFORMS (Continued)

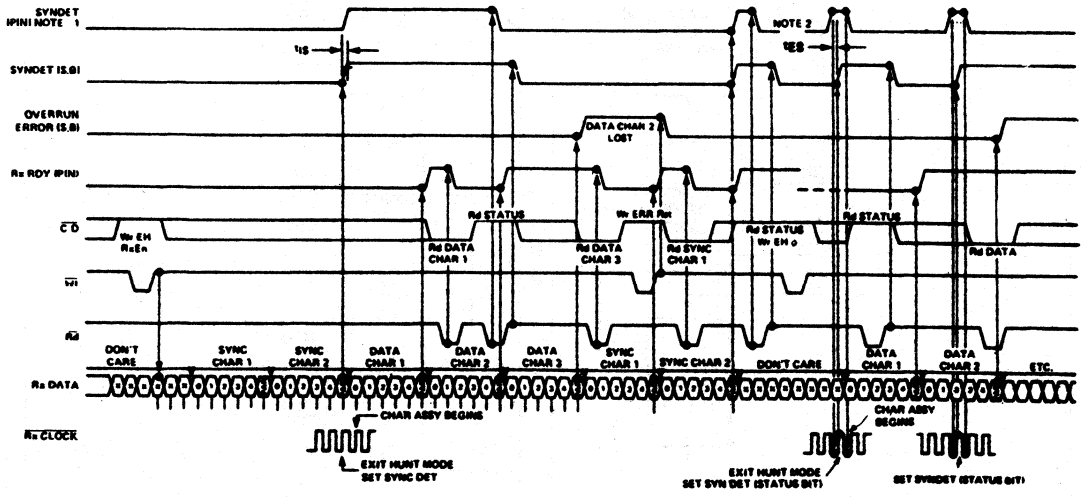
TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



WF006120

EXAMPLE FORMAT = 5 BIT CHARACTER WITH PARITY 2 SYNC CHARACTERS.

RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)



WF006130

Notes: 1. INTERNAL SYNC. 2 SYNC CHARACTERS. 5 BITS WITH PARITY
2. EXTERNAL SYNC. 5 BITS WITH PARITY

8253

Programmable Interval Timer
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

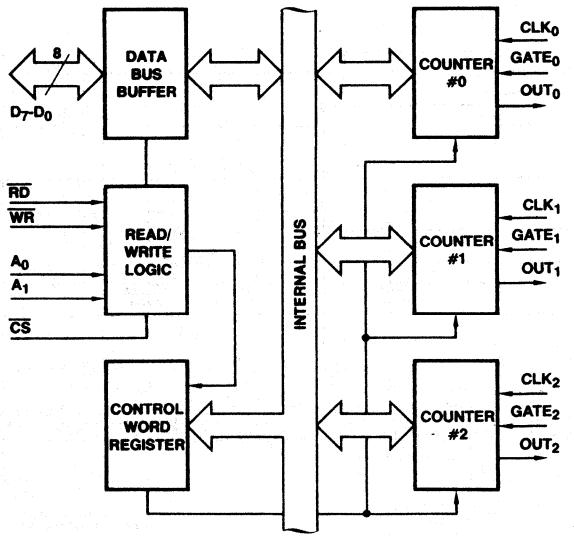
- Both Binary and BCD counting
- Single +5V supply
- Three independent 16-bit counters
- DC to 5MHz
- Programmable counter modes
- Bus oriented I/O

GENERAL DESCRIPTION

The 8253 is a programmable counter/timer chip designed for use with 8080A/8085A microprocessors. It uses NMOS technology with a single +5V supply and is a direct replacement for Intel's 8253/8253-5.

Each device is organized as three independent 16-bit counters, each counter having a rate of up to 5MHz. All modes of operation are software programmable. For improved performance devices see the Am9513A System Timing Controller.

BLOCK DIAGRAM

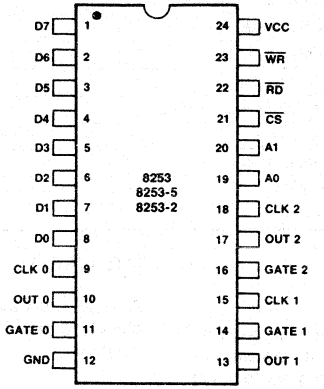


BD003760

041264

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-24-1, P-24-1



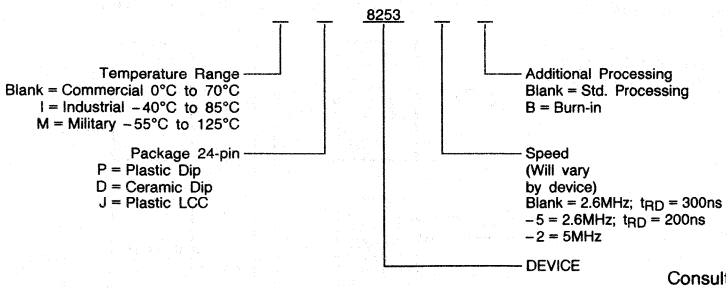
CD005670

Note: Pin 1 is marked for orientation

Also available in PLCC. See section 7 for pinout details.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8253 8253B	P, D, ID
8253-5 8253-5B	P, D
8253-2 8253-2B	P, D
8253	/BJA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
1-8	D7-D0	I/O	Data bus (8-bit).
9, 15, 18	CLK N	I	Counter clock inputs.
11, 14, 16	GATE N	I	Counter gate inputs.
10, 13, 17	OUT N	O	Counter outputs.
22	$\overline{RD}$	I	Read counter.
23	$\overline{WR}$	I	Write command or data.
21	$\overline{CS}$	I	Chip select.
19, 20	A0-A1	I	Counter select.
24	VCC		+5 Volts.
12	GND		Ground.

DETAILED DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with 8080A Microcomputer systems. Its function is that of a general-purpose, multitiming element that can be treated as an array of I/O ports in the system's software.

The 8253 solves one of the most common problems in any microcomputer system: the generation of accurate time delays under software control. Instead of setting up timing loops in the system's software, the programmer configures the 8253 to match his requirements and initializes one of the counters of the 8253 with a desired quantity. Then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its task. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real-Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions:

1. Programming the MODES of the 8253,
2. Loading the count registers, and
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no Read or Write operation can occur unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "LOW" on this input informs the 8253 that the CPU is inputting data in the form of a counter's value.

$\overline{WR}$ (Write)

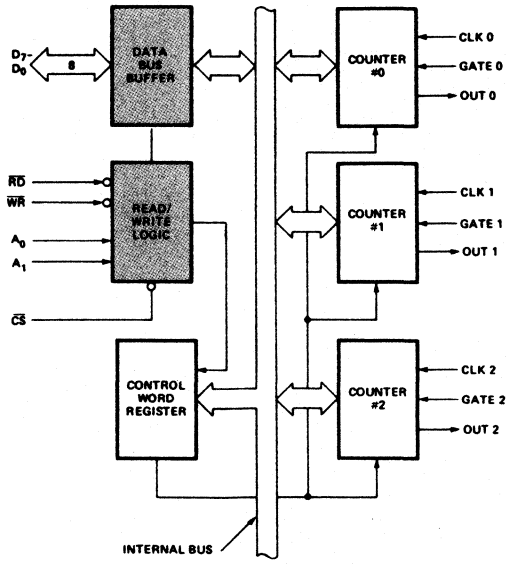
A "LOW" on this input informs the 8253 that the CPU is outputting data in the form of MODE information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for MODE selection.

$\overline{CS}$ (Chip Select)

A "LOW" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.



BD005101

Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, the selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

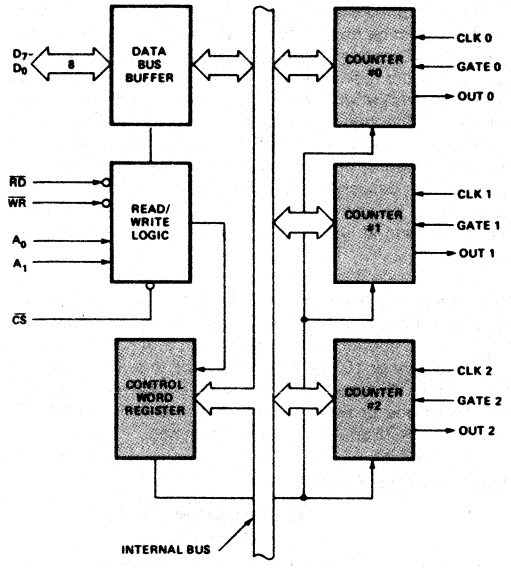
Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described. Each Counter consists of a single, 16-bit, presettable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have a different MODE configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

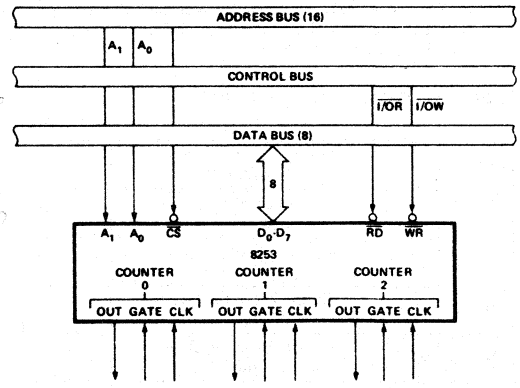
The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications, and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write MODE Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State



BD005101

Figure 2. Block Diagram Showing Control Word Register and Counter Functions



BD005120

Figure 3. 8253 System Interface

8253 SYSTEM INTERFACE

The 8253 is a component of the iAPX Family and interfaces in the same manner as all other peripherals of the family. It is treated by the system's software as an array of peripheral I/O ports; three are counters, and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method, or it can be connected to the output of a decoder, such as an AMD Am25LS2548 or Am25LS2538 for larger systems.

8253 READ/WRITE PROCEDURE

Write Operations

The system's software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection; e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent (SC0, SC1).

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded, it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 a new count will not start until the load has been completed. The count register will accept one or two bytes depending on how the MODE control words (RL0, RL1) are programmed.

Programming Format

MODE Control Word
Counter n

LSB	Count Register byte Counter n
MSB	Count Register byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Alternate Programming Formats

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253, the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1-11). The only requirement with this method is that to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input. The contents of the counter selected will be available as follows:

first I/O Read contains the least significant byte (LSB).

second I/O Read contains the most significant byte (MSB).

Due to the internal logic of the 8253, it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read, then two bytes must be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

For the programmer to read the contents of any counter without effecting or disturbing the counting operation, the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly," he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter, and the contents of the latched register are available.

MODE Register for Latching Count

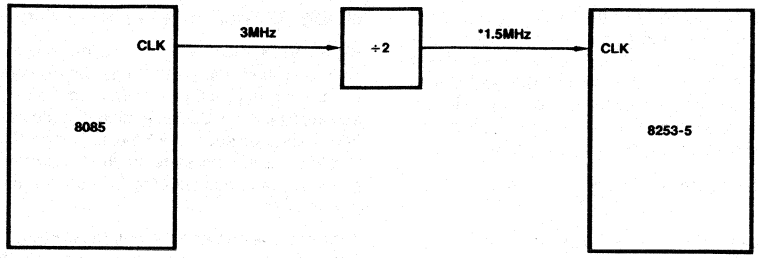
A0, A1 = 11

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SCI,SCO – specify counter to be latched.
D5,D4 – 00 designates counter latching operation.
X – don't care

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.

Clock Interface*



AF002870

*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2MHz or less.

PROGRAMMING INFORMATION

General

The complete functional definition of the 8253 is programmed by the system's software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. These control words program the MODE, loading sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned.

The actual counting operation of each counter is completely independent, and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the system's software by simple I/O operations.

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register (A0, A1 = 11).

Control Word Format

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

SC – Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL – Read/Load:

RL1	RL0	
0	0	Counter Latching operation.
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

M – MODE:

M2	M1	M0	
0	0	0	MODE 0
0	0	1	MODE 1
X	1	0	MODE 2
X	1	1	MODE 3
1	0	0	MODE 4
1	0	1	MODE 5

BCD:

0	Binary Counter 16-bits
1	Binary Code Decimal (BCD) Counter (4 Decades)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE DEFINITION

MODE 0: Interrupt on Terminal Count

The output will be initially LOW after the mode set operation. After the count is loaded into the selected count register, the output will remain LOW and the counter will count. When terminal count is reached, the output will go HIGH and remain HIGH until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

1. Write 1st byte stops the current counting.
2. Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot

The output will go LOW on the count following the rising edge of the gate input.

The output will go HIGH on the terminal count. If a new count value is loaded while the output is LOW, it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain LOW for the full count after any rising edge of the gate input.

MODE 2: Rate Generator

Divide by N counter. The output will be LOW for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses, the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when LOW, will force the output HIGH. When the gate input goes HIGH, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain HIGH until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator

Similar to MODE 2 except that the output will remain HIGH until one half the count has been completed (for even numbers) and go LOW for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is HIGH, the first clock pulse (after the count is loaded) decrements the count by one. Subsequent clock pulses decrement the clock by two. After timeout, the output goes LOW and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by three. Subsequent clock pulses decrement the count by two until time-out. Then the whole process is repeated. In this way, if the count is odd, the output will be HIGH for $(N + 1)/2$ counts and LOW for $(N - 1)/2$ counts.

MODE 4: Software-Triggered Strobe

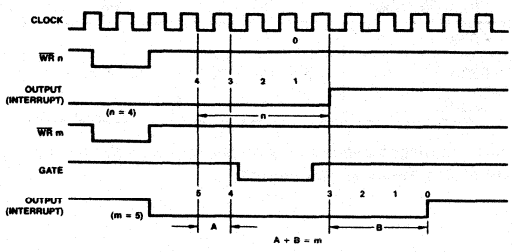
After the mode is set, the output will be HIGH. When the count is loaded, the counter will begin counting. On terminal count, the output will go LOW for one input clock period, then will go HIGH again.

If the count register is reloaded between output pulses, the present period will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is LOW. Reloading the counter register will restart counting beginning with the new number.

MODE 5: Hardware-Triggered Strobe

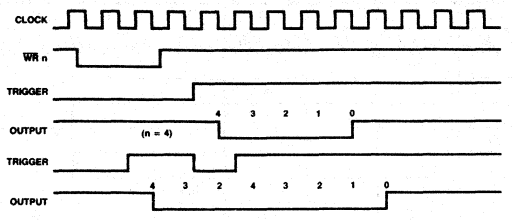
The counter will start counting after the rising edge of the trigger input and will go LOW for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go LOW until the full count after the rising edge of any trigger.

MODE 0. Interrupt on Terminal Count.



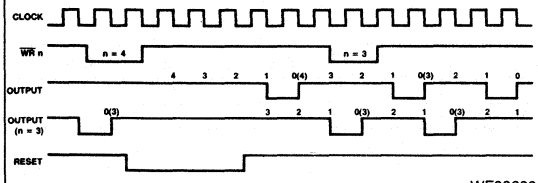
WF006860

MODE 1. Programmable One-Shot.



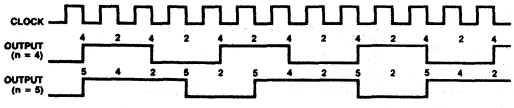
WF006870

MODE 2. Rate Generator.



WF006880

MODE 3. Square Wave Generator.

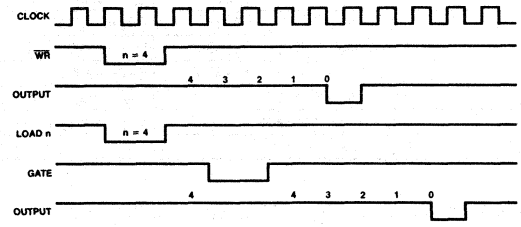


WF006890

Gate Pin Operations Summary

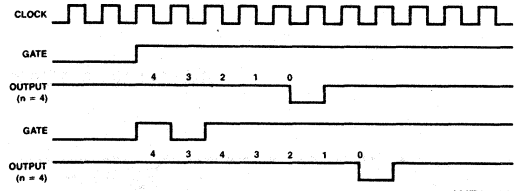
Modes	Signal Status		
	Low Or Going Low	Rising	High
0	Disables counting	–	Enables counting
1	–	1) Initiates counting 2) Resets output after next clock	–
2	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	–	Enables counting
5	–	Initiates counting	–

MODE 4. Software-Triggered Strobe.



WF006900

MODE 5. Hardware-Triggered Strobe.



WF006910

8253 Timing Diagrams

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage On Any Pin
with Respect to Ground ..... −0.5 to +7.0V
Power Dissipation 1W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8253	0°C to 70°C	5V ±10%
8253-5		
8253-2		

Operating ranges define those limits over which the functionality of the device is guaranteed.

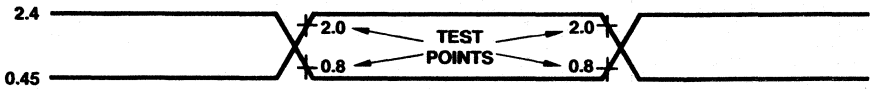
DC CHARACTERISTICS (over Operating Ranges)

Parameters	Description		Test Conditions	8253 Min	8253 Max	8253-5 Min	8253-5 Max	8253-2 Min	8253-2 Max	Units
V_{IL}	Input Low Voltage			−.5	.8	−.5	.8	−.5	.8	V
V_{IH}	Input High Voltage			2.2	V_{CC} + .5V	2.2	V_{CC} + .5V	2.2	V_{CC} + .5V	V
V_{OL}	Output Low Voltage	8253	I_{OL} = 1.6mA		.45					V
		8253-5	I_{OL} = 2.2mA				.45			
		8253-2	I_{OL} = 2.2mA						.45	
V_{OH}	Output High Voltage	8253	I_{OH} = −150μA	2.4						V
		8253-5	I_{OH} = −400μA			2.4				
		8253-2	I_{OH} = −400μA					2.4		
I_{IL}	Input Load Current		V_{IN} = V_{CC} to 0V		±10		±10		±10	μA
I_{OFL}	Output Float Leakage		V_{OUT} = V_{CC} to 0V		±10		±10		±10	μA
I_{CC}	V_{CC} Supply Current				140		140		140	mA

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Parameters	Description	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	f_c = 1MHz			10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured pins returned to V_{SS}			20	pF

SWITCHING TEST INPUT WAVEFORM



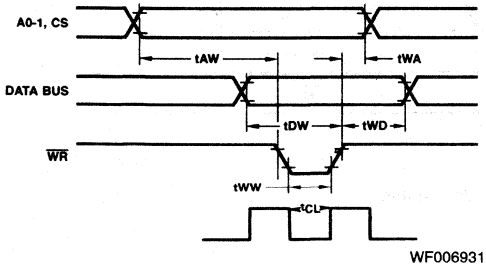
WF006950

SWITCHING CHARACTERISTICS (over Operating Ranges)

Parameters	Description	8253 Min	8253 Max	8253-5 Min	8253-5 Max	8253-2 Min	8253-2 Max	Units
Read Cycle								
t_{AR}	Address Stable Before $\overline{READ}$	50		30		25		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		5		5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		300		150		ns
t_{RD}	Data Delay from $\overline{READ}$ (Note 2)		300		200		120	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	25	100	25	100	ns
t_{RV}	Recovery Time Between $\overline{READ}$ and Any Other Control Signal	1		1		500		μs
Write Cycle								
t_{AW}	Address Stable Before $\overline{WRITE}$	50		30		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		30		0		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		300		150		ns
t_{DW}	Data Set-up Time for $\overline{WRITE}$	300		250		100		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		30		0		ns
t_{RV}	Recovery Time Between $\overline{WRITE}$ and Any Other Control Signal (Note 3)	1		1		500		μs

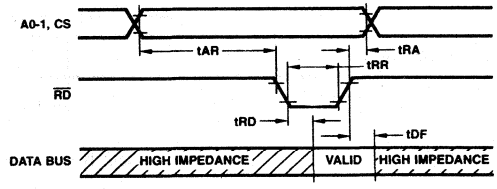
Notes: 1. AC timings measured at V_{OH} = 2.2, V_{OL} = 0.8.
2. Test Conditions: 8253, C_L = 100pF; 8253-5, 8253-2, C_L = 150pF.
3. t_{RV} applies for any READ or WRITE that occurs regardless of the state of the CS input.
4. If the clock occurs less than 100ns after the rising edge of READ or WRITE, the counter selected during the READ or WRITE could be affected.

Write Timing



WF006931

Read Timing



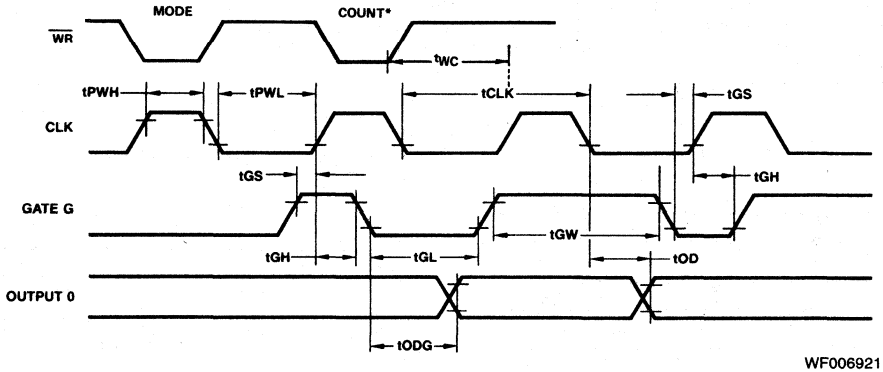
WF006940

CLOCK AND GATE TIMING

Parameters	Description	8253		8253-5		8253-2		Units
		Min	Max	Min	Max	Min	Max	
t_{CLK}	Clock Period	380	DC	380	DC	200	DC	ns
t_{PWH}	High Pulse Width	230		230		90		ns
t_{PWL}	Low Pulse Width	150		150		90		ns
t_{GW}	Gate Width High	150		150		120		ns
t_{GL}	Gate Width Low	100		100		80		ns
t_{GS}	Gate Set-up Time to CLK↑	100		100		60		ns
t_{GH}	Gate Hold Time After CLK↑	50		50		50		ns
t_{OD}	Output Delay from CLK↓ (Note 1)		400		400		250	ns
t_{ODG}	Output Delay from Gate↓ (Note 1)		300		300		150	ns
t_{WC}	Write to CLK Set-up	450		350		350		ns

Note: Test Conditions: 8253, C_L = 100pF; 8253-5, 8253-2, C_L = 150pF.

CLOCK AND GATE TIMING



WF006921

*last byte of count being written

8255A

Programmable Peripheral Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Direct bit set/reset capability easing control application interface
- Reduces system package count
- Improved DC driving capability
- 24 programmable I/O pins
- Completely TTL compatible
- Fully compatible with the iAPX86 microprocessor family
- Improved timing characteristics

GENERAL DESCRIPTION

The 8255A is a general purpose programmable I/O device designed for use with iAPX Family microprocessors. It has 24 I/O pins which may be individually programmed in two groups of twelve and used in three major modes of operation. In the first mode, each group of twelve I/O pins may be programmed in sets of 4 and 8 to be input or output. In Mode 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining four pins, three are used for handshaking and interrupt control signals. The third mode of operation (Mode 2) is a bidirectional bus mode which uses eight lines for a bidirectional bus, and five lines, borrowing one from the other group, for handshaking.

BLOCK DIAGRAM

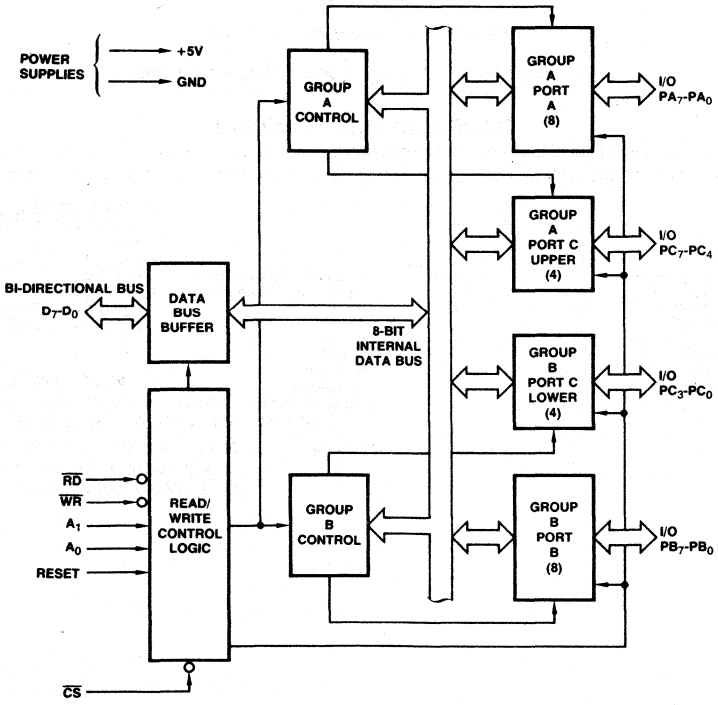


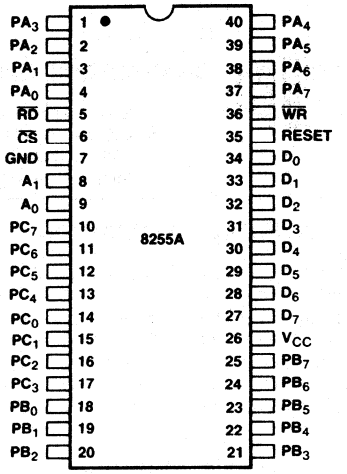
Figure 1.

04122

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40-1, P-40-1



CD005400

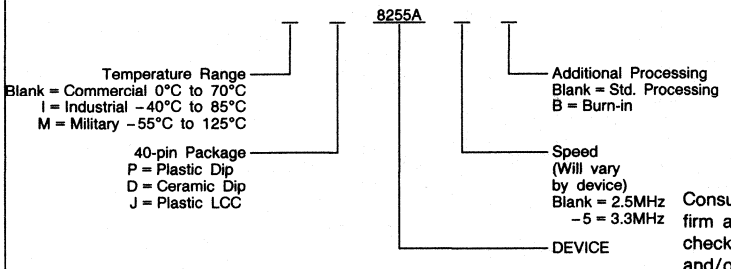
Figure 2.

Note: Pin 1 is marked for orientation

Also available in PLCC. See Section 7 for pinout details.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8255A 8255AB 8255A-5 8255A-5B	P, D, ID
8255A	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Pin Description
27-34	D_7-D_0	I/O	Data Bus (Bidirectional).
35	Reset	I	Reset Input.
6	$\overline{CS}$	I	Chip Select.
5	$\overline{RD}$	I	Read Input.
36	$\overline{WR}$	I	Write Input.
9, 8	A_0, A_1	I	Port Address.
37-40, 1-4	PA_7-PA_0	I/O	Port A (Bit).
25-18	PB_7-PB_0	I/O	Port B (Bit).
10-13, 17-14	PC_7-PC_0	I/O	Port C (Bit).
26	V_{CC}		+5 Volts.
7	GND		0 Volts.

DETAILED DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and, in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A→DATA BUS
0	1	0	1	0	PORT B→DATA BUS
1	0	0	1	0	PORT C→DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS→PORT A
0	1	1	0	0	DATA BUS→PORT B
1	0	1	0	0	DATA BUS→PORT C
1	1	1	0	0	DATA BUS→CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS→3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS→3-STATE

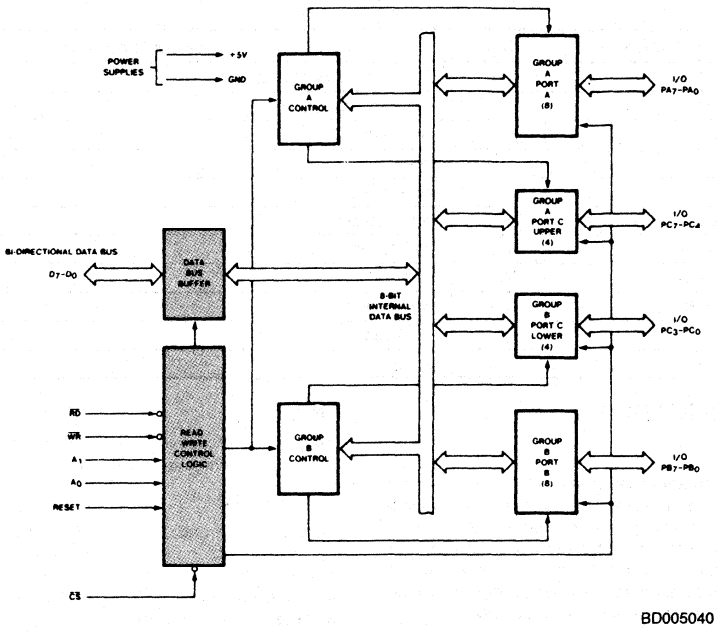


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8225A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7 – C4)
Control Group B – Port B and Port C lower (C3 – C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

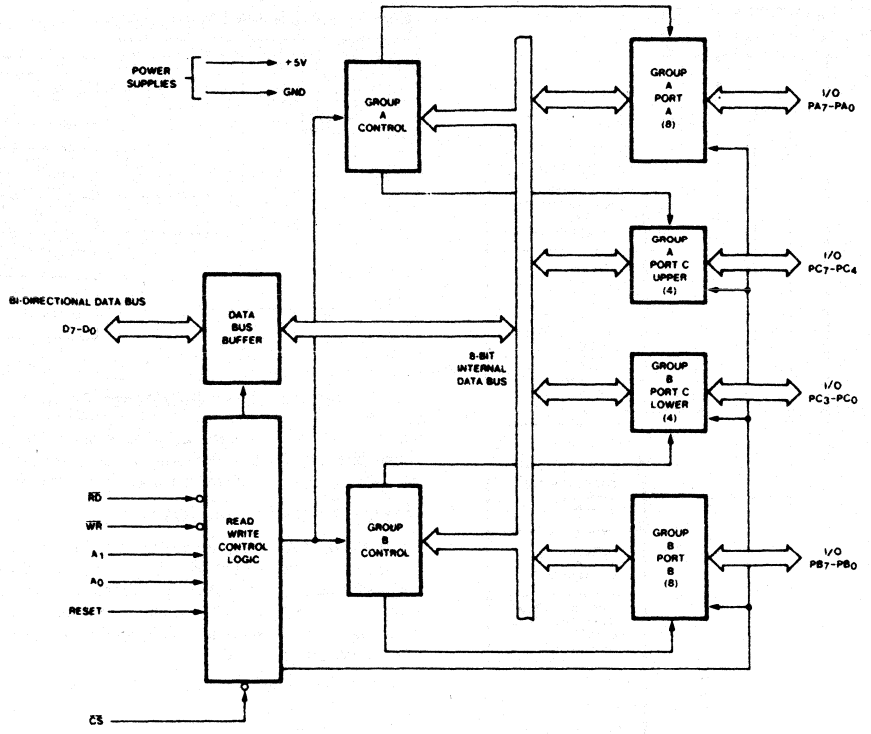


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

PROGRAMMING INFORMATION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance, Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, and Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

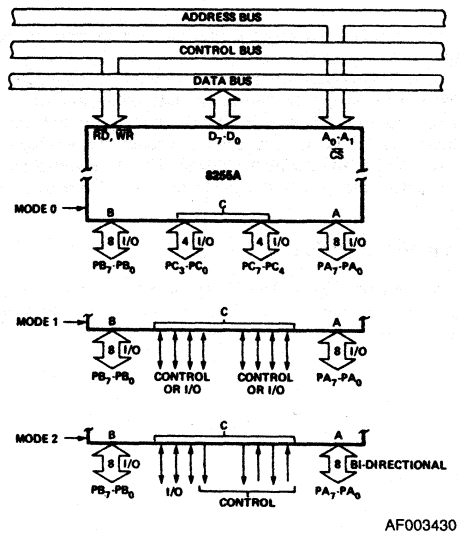


Figure 5. Basic Mode Definitions and Bus Interface

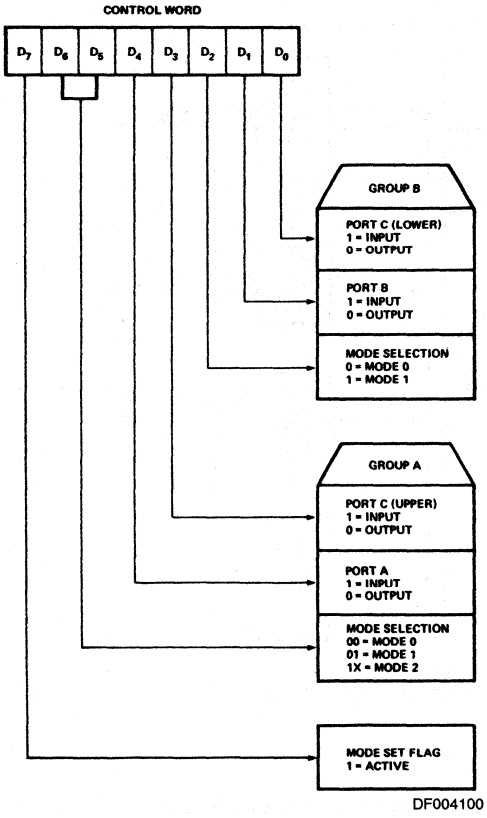


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first, but after a cursory review of the compete device operation, a simple, logical I/O approach will surface. The design of the 8255A has taken into account things, such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

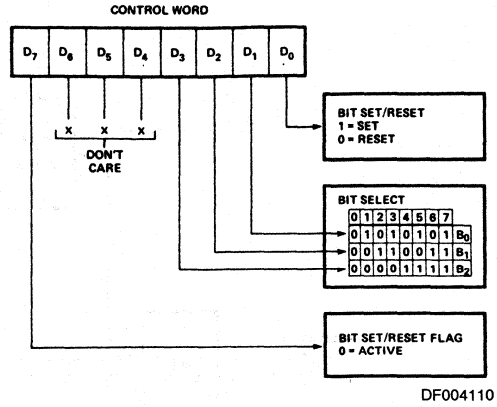


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) – INTE is SET – Interrupt enable
(BIT-RESET) – INTE is RESET – Interrupt disable

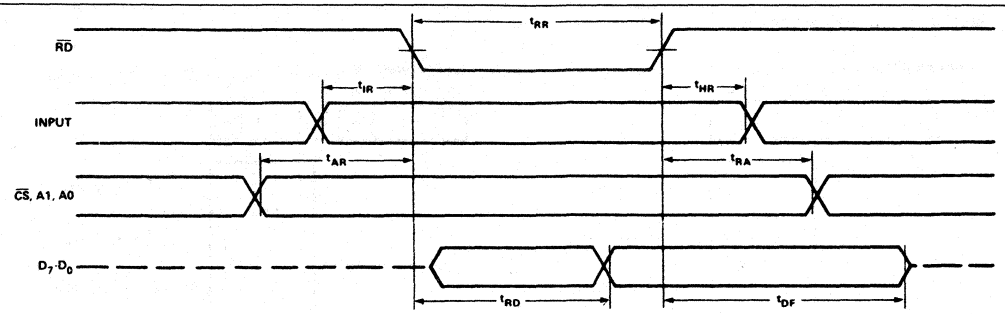
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

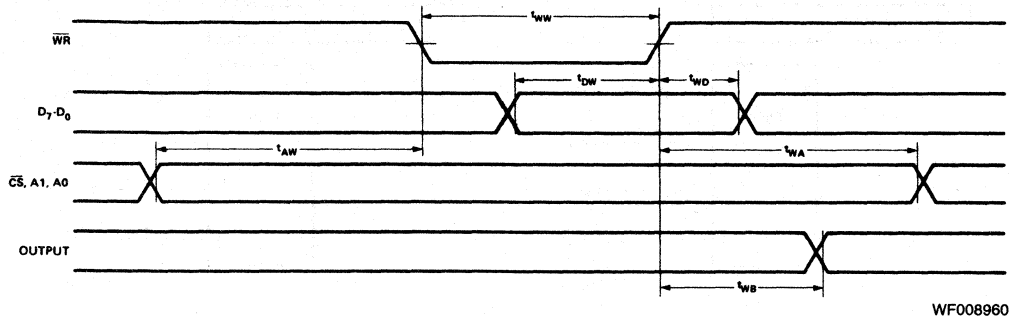
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



Mode 0 (Basic Input)



MODE 0 (Basic Output)

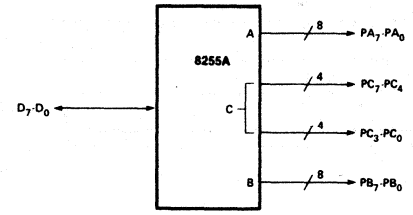
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

CONTROL WORD =0

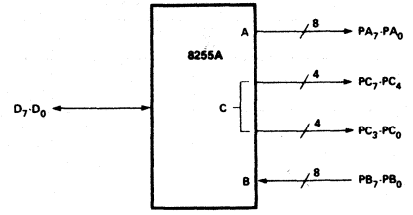
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



LS001460

CONTROL WORD =2

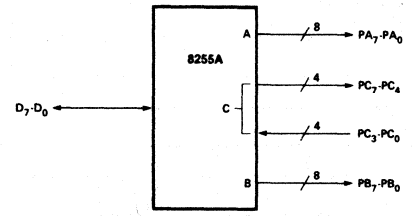
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



LS001470

CONTROL WORD #1

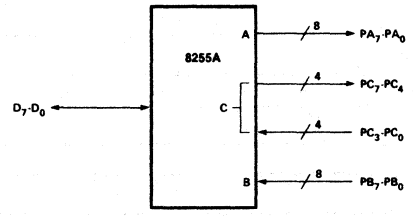
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



LS001480

CONTROL WORD #3

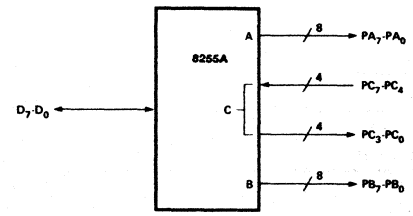
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



LS001490

CONTROL WORD #4

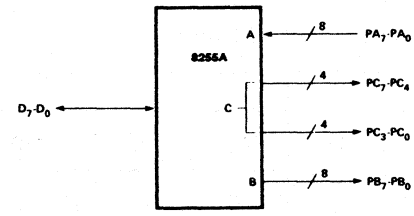
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



LS001500

CONTROL WORD =8

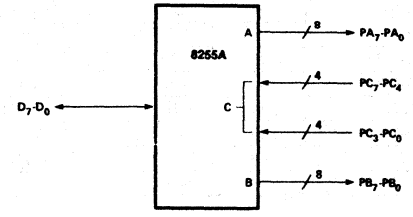
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



LS001510

CONTROL WORD #5

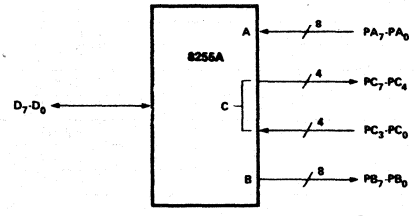
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



LS001520

CONTROL WORD #9

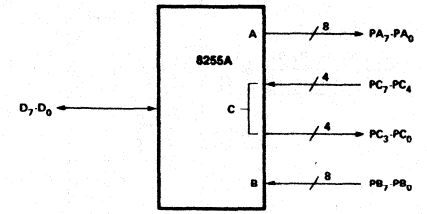
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



LS001530

CONTROL WORD =6

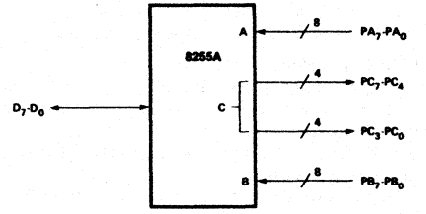
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



LS001540

CONTROL WORD =10

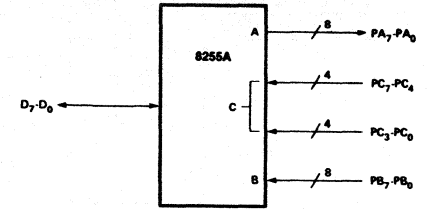
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



LS001550

CONTROL WORD =7

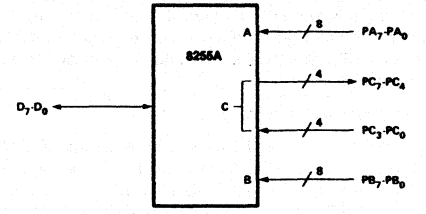
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



LS001560

CONTROL WORD =11

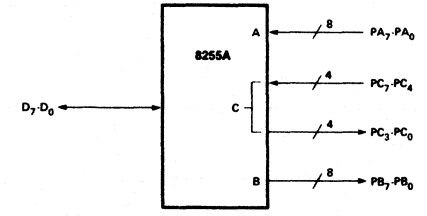
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



LS001570

CONTROL WORD =12

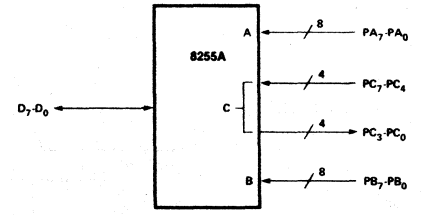
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0



LS001580

CONTROL WORD =14

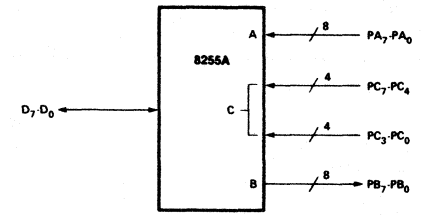
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	0



LS001590

CONTROL WORD =13

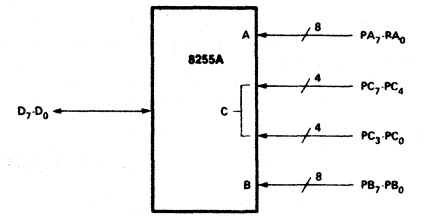
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	1



LS001600

CONTROL WORD =15

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	1



LS001610

Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch: in essence, an acknowledgment. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set when $\overline{STB}$ is a "one," IBF is a "one" and INTE is a "one." It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A
Controlled by bit set/reset of PC_4.

INTE B
Controlled by bit set/reset of PC_2.

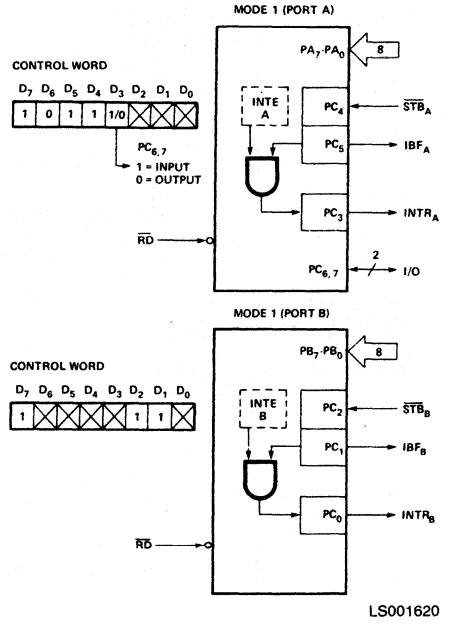


Figure 8. MODE 1 Input

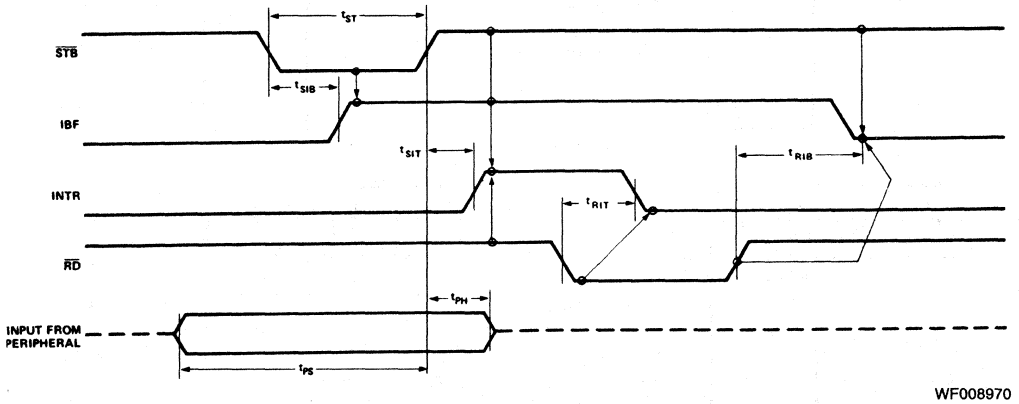


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by ACK Input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a "one," OBF is a "one" and INTE is a "one." It is reset by the falling edge of WR.

INTE A
Controlled by bit set/reset of PC_6.
INTE B
Controlled by bit set/reset of PC_2.

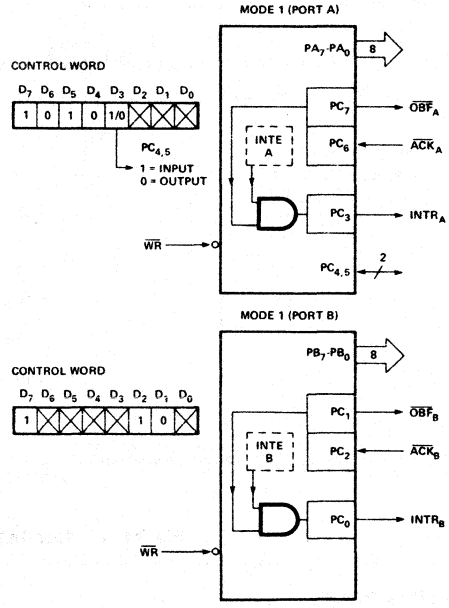


Figure 10. MODE 1 Output

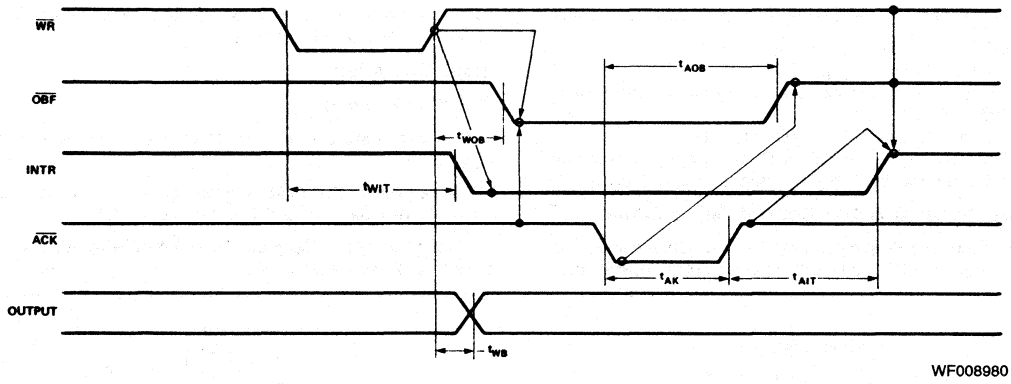
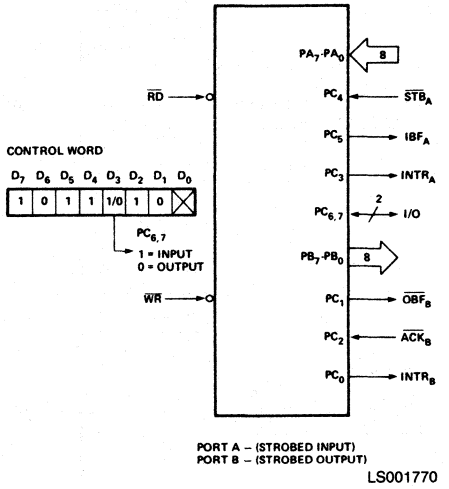


Figure 11. Mode 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.



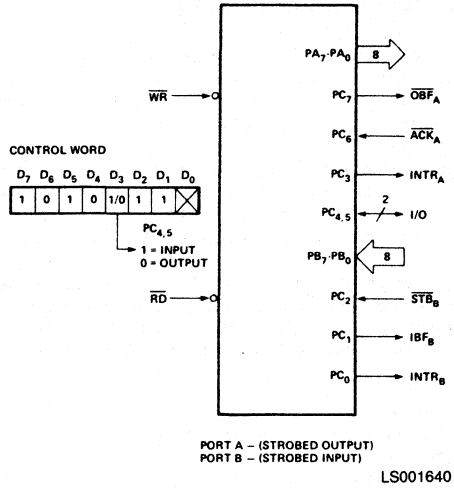


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The OBF output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input)

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

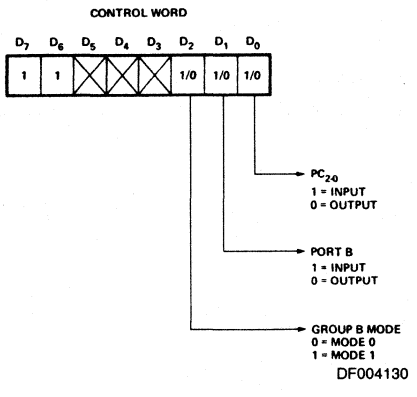


Figure 13. MODE Control Word

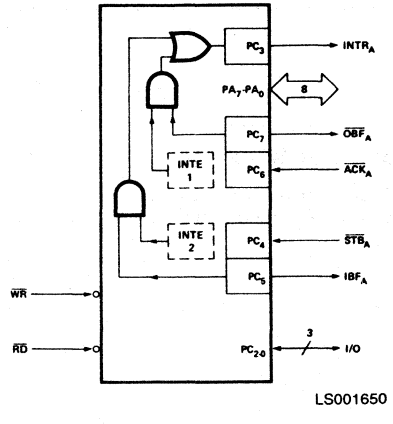


Figure 14. MODE 2

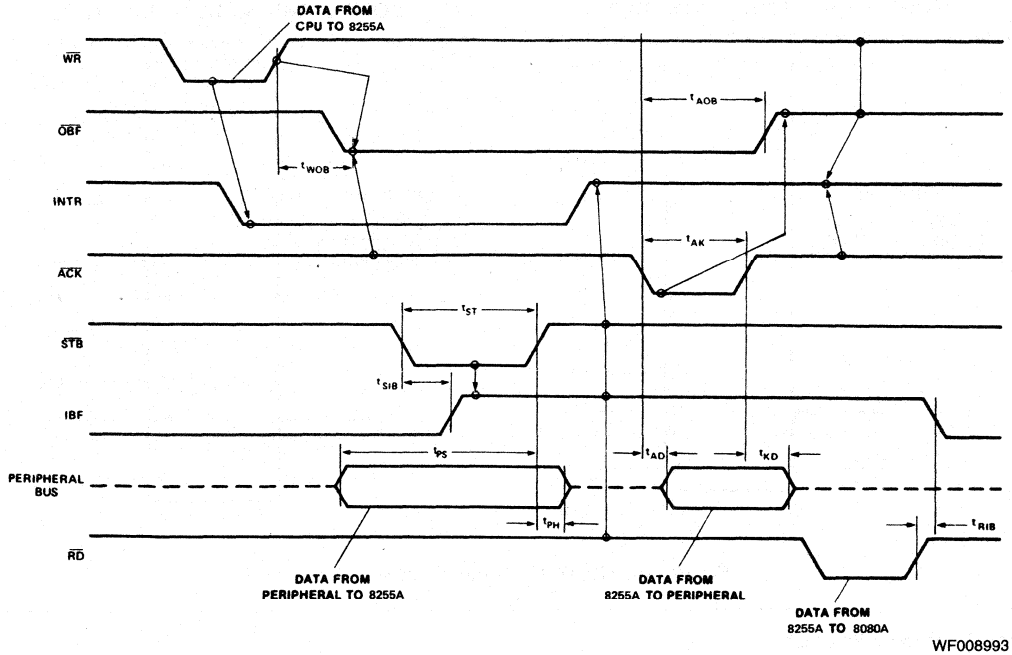
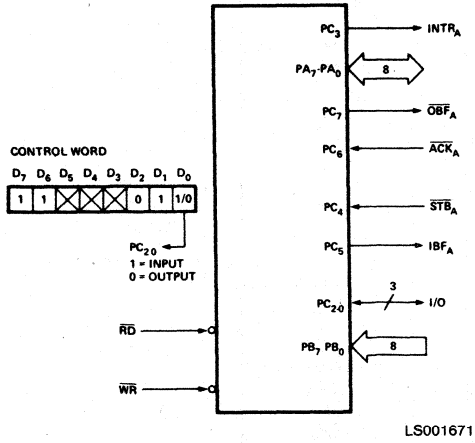


Figure 15. MODE 2 (Bidirectional)

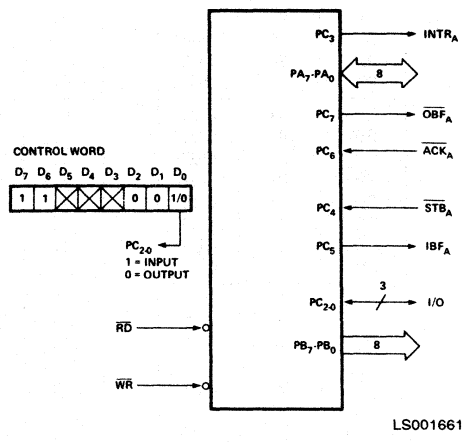
Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR}$)

MODE 2 AND MODE 0 (INPUT)



LS001671

MODE 2 AND MODE 0 (OUTPUT)



LS001661

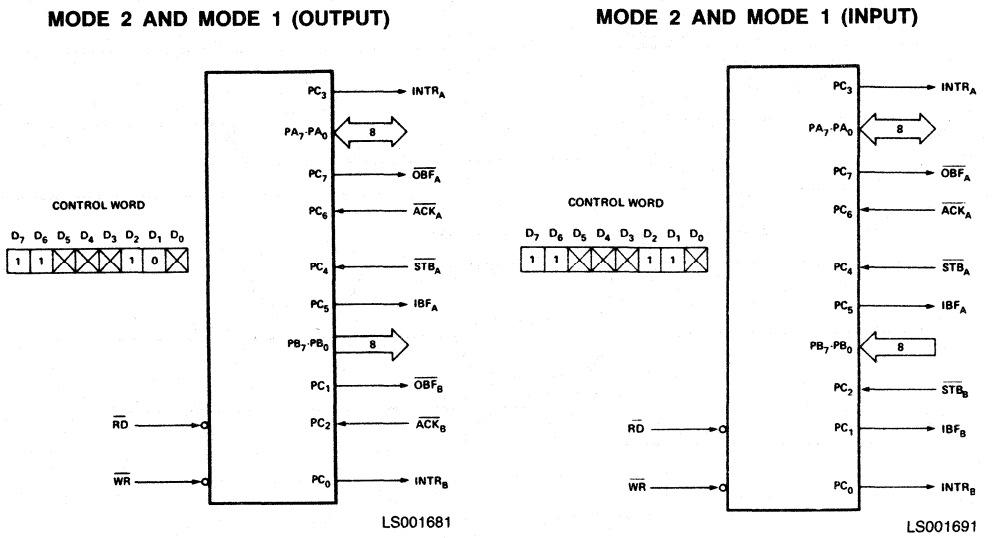


Figure 16. MODE ¼ Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$\overline{INTR_B}$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF_B}$	I/O	
PC_2	IN	OUT	$\overline{STB_B}$	$\overline{ACK_B}$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB_A}$	I/O	$\overline{STB_A}$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK_A}$	$\overline{ACK_A}$	
PC_7	IN	OUT	I/O	$\overline{OBF_A}$	$\overline{OBF_A}$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs –
All input lines can be accessed during a normal Port C read.

If Programmed as Outputs –
Bits in C upper ($PC_7 - PC_4$) must be individually accessed using the bit set/reset function.

Bits in C lower ($PC_3 - PC_0$) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255A to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255A is programmed to function in Modes 1 or 2, Port C generates or accepts "handshaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

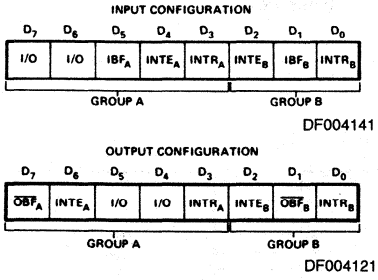


Figure 17. MODE 1 Status Word Format

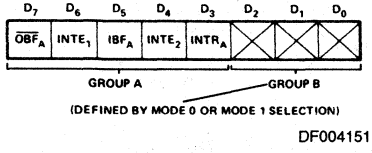


Figure 18. MODE 2 Status Word Format

APPLICATIONS INFORMATION

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 present a few examples of typical applications of the 8255A.

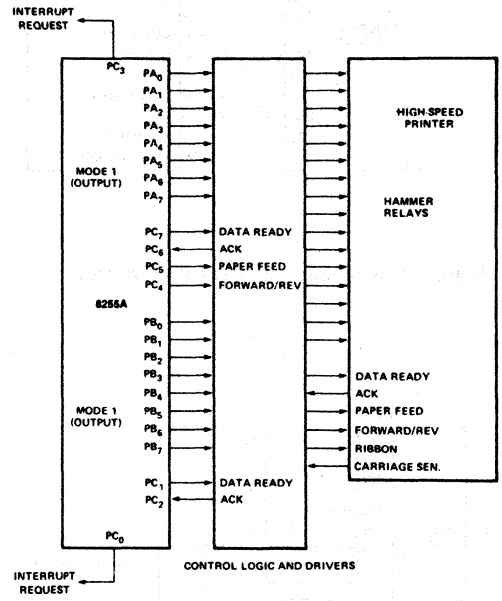
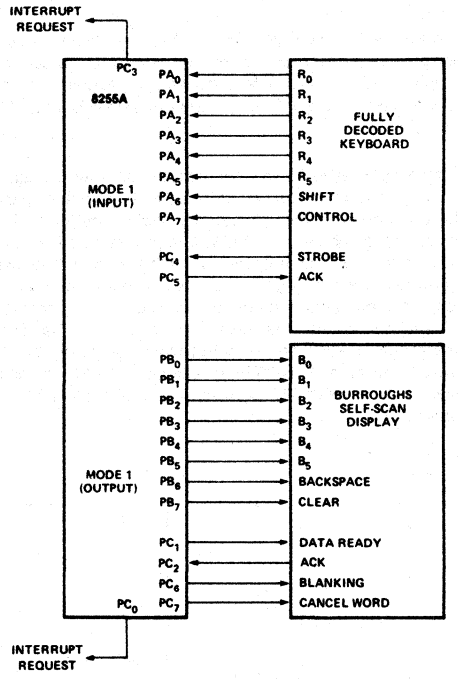
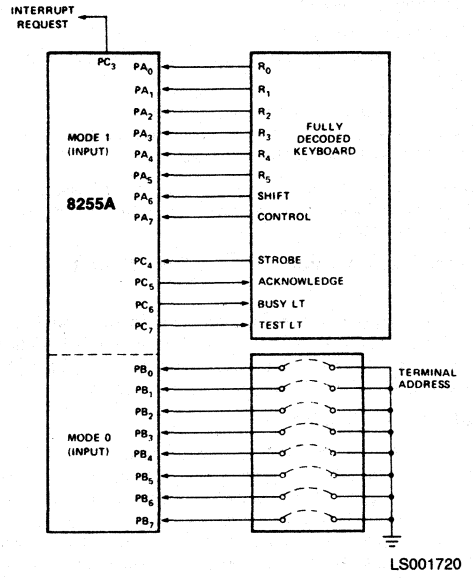


Figure 19. Printer Interface



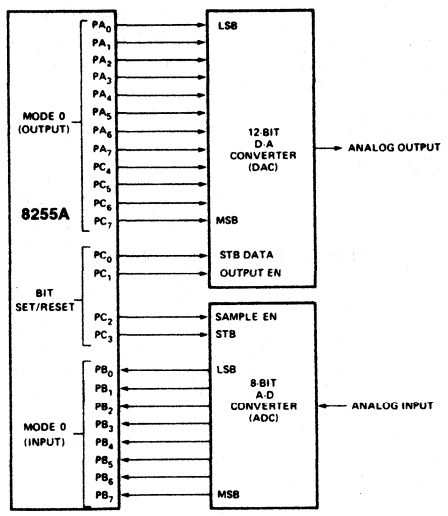
LS001700

Figure 20. Keyboard and Display Interface



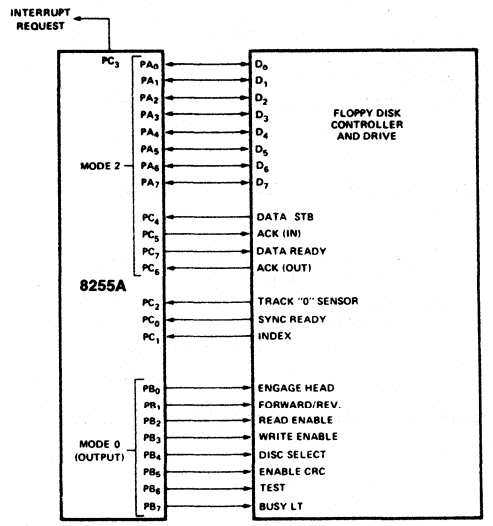
LS001720

Figure 21. Keyboard and Terminal Address Interface



LS001730

Figure 22. Digital to Analog, Analog to Digital



LS001740

Figure 23. Basic Floppy Disk Interface

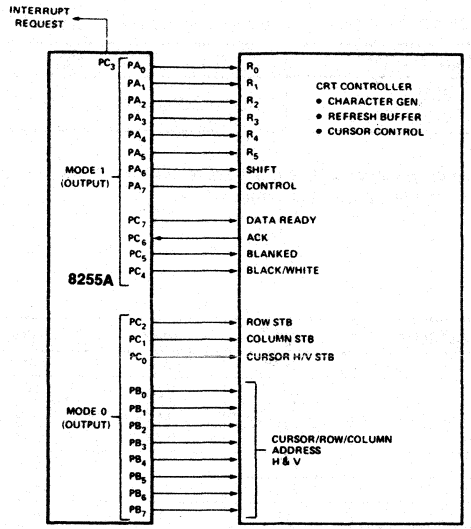


Figure 24. Basic CRT Controller Interface

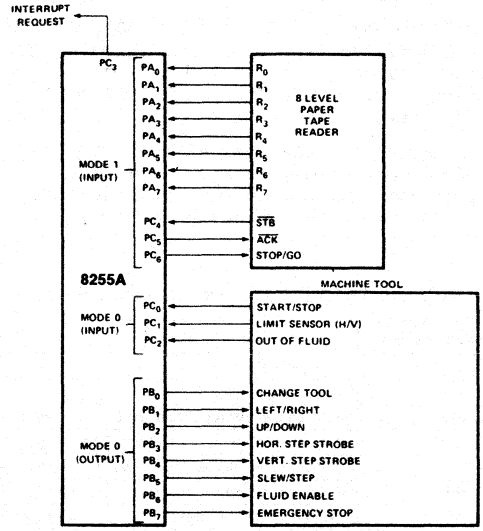


Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to 7.0V
All Signal Voltages
with Respect to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8255A 8255A-5	0°C to 70°C	5V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over Operating Ranges)

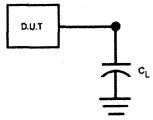
Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	Volts
V_{IH}	Input High Voltage		2.0	V_{CC}	Volts
V_{OL}(DB)	Output Low Voltage (Data Bus)	I_{OL} = 2.5mA		0.45	Volts
V_{OL}(PER)	Output Low Voltage (Peripheral Port)	I_{OL} = 1.7mA		0.45	Volts
V_{OH}(DB)	Output High Voltage (Data Bus)	I_{OH} = −400μA	2.4		Volts
V_{OH}(PER)	Output High Voltage (Peripheral Port)	I_{OH} = −200μA	2.4		Volts
I_{DAR} (Note 1)	Darlington Drive Current	R_{EXT} = 750Ω; V_{EXT} = 1.5V	−1.0	−4.0	mA
I_{CC}	Power Supply Current			120	mA
I_{IL}	Input Load Current	V_{IN} = V_{CC} to 0V		±10	μA
I_{OFL}	Output Float Leakage	V_{OUT} = V_{CC} to 0V		±10	μA

Note 1: Available on any 8 pins from Port B and C.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Parameters	Description	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	fc = 1MHz			10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured pins returned to GND			20	pF

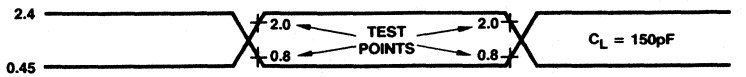
SWITCHING TEST LOAD CIRCUIT



TC002141

*V_{EXT} is set at various voltages during testing to guarantee the specification.
C_L includes jig capacitance.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF006350

AC testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0."
Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

BUS PARAMETERS

READ

Parameters	Description	8255A		8255A-5		Units
		Min	Max	Min	Min	
t_{AR}	Address Stable Before READ	0		0		ns
t_{RA}	Address Stable After READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ (Note 1)		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

WRITE

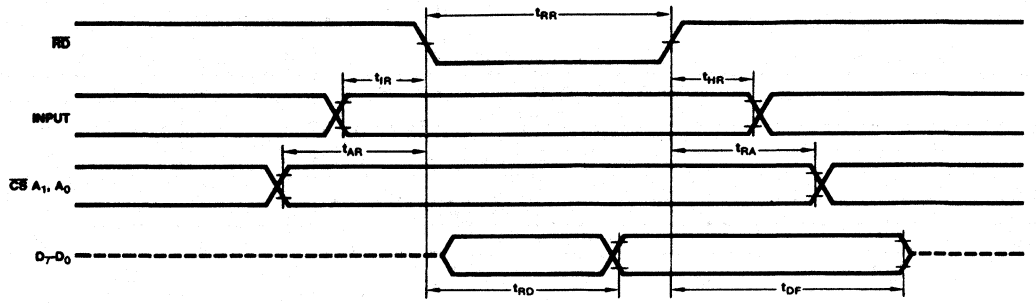
Parameters	Description	8255A		8255A-5		Units
		Min	Max	Min	Max	
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

OTHER TIMINGS

Parameters	Description	8255A		8255A-5		Units
		Min	Max	Min	Max	
t_{WB}	WR = 1 to Output (Note 1)		350		350	ns
t_{IR}	Peripheral Data Before RD	0		0		ns
t_{HR}	Peripheral Data After RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output (Note 1)		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0 (Note 1)		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1 (Note 1)		350		350	ns
t_{SIB}	STB = 0 to IBF = 1 (Note 1)		300		300	ns
t_{RIB}	RD = 1 to IBF = 0 (Note 1)		300		300	ns
t_{RIT}	RD = 0 to INTR = 0 (Note 1)		400		400	ns
t_{SIT}	STB = 1 to INTR = 1 (Note 1)		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1 (Note 1)		350		350	ns
t_{WIT}	WR = 0 to INTR = 0 (Notes 1, 3)		450		450	ns

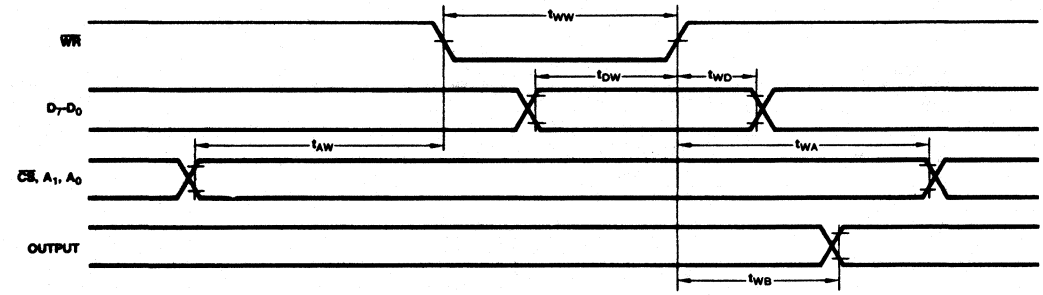
Notes: 1. Test Conditions: 8255A: C_L = 100pF; 8255A-5: C_L = 100pF.
2. Period of Reset pulse must be at least 50μ during or after power on. Subsequent Reset pulse can be 500ns min.
3. INTR↑ may occur as early as WR↓.

SWITCHING WAVEFORMS



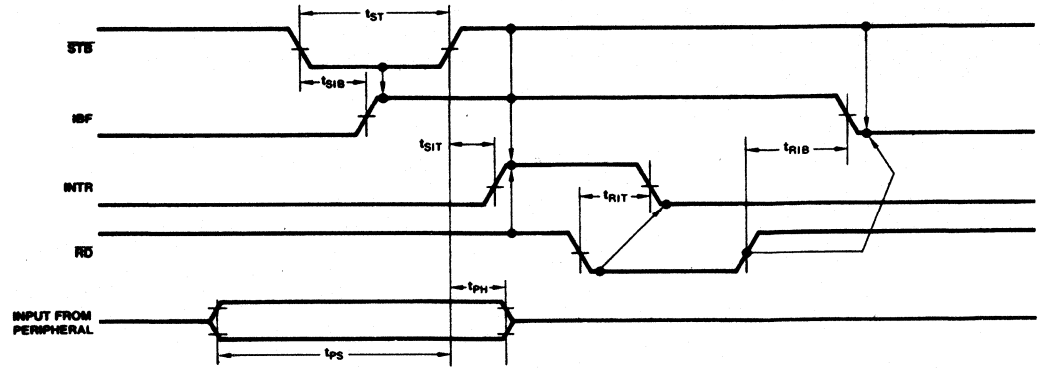
WF006320

Mode 0 (Basic Input)



WF006330

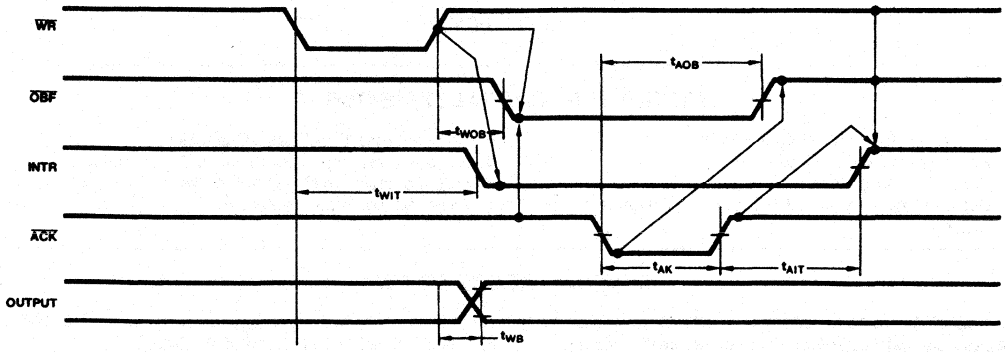
Mode 0 (Basic Output)



WF006340

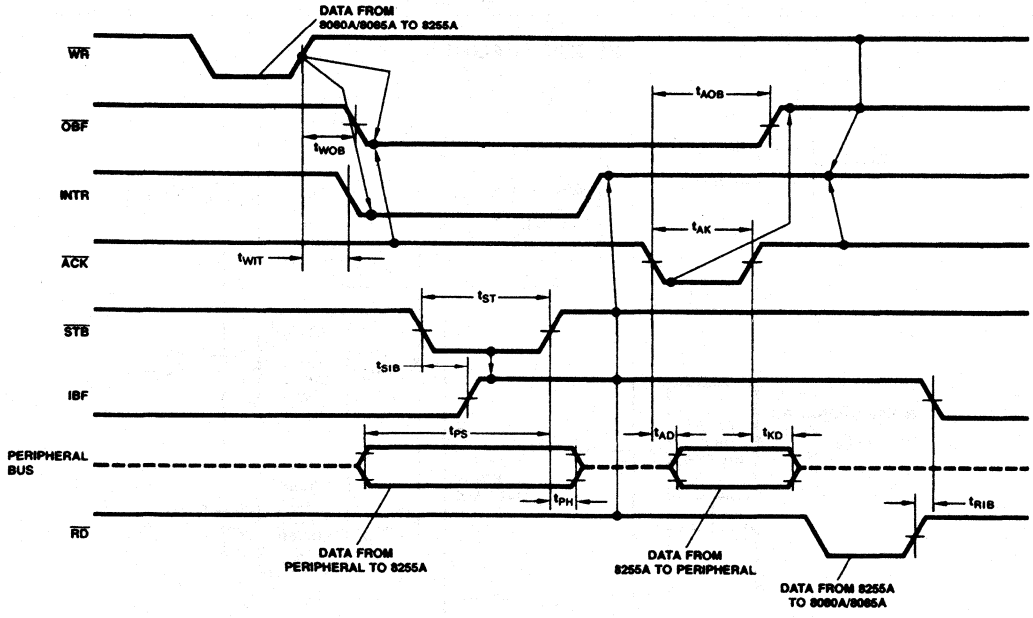
Mode 1 (Strobed Input)

SWITCHING WAVEFORMS (Cont.)



WF006361

Mode 1 (Strobed Output)



WF006370

Mode 2 (Bidirectional)

Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occur before $\overline{RD}$ is permissible (INTR = IBF · $\overline{MASK}$ · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · $\overline{MASK}$ · $\overline{ACK}$ · $\overline{WR}$).

82C55A

CMOS Programmable Peripheral Interface
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Pin compatible with NMOS 8255A
- 24 programmable I/O pins
- Fully TTL compatible
- Bus hold circuitry on all I/O ports – eliminates pull-up resistors
- Control Word Read-Back Capability
- 2.5 mA drive capability on all I/O port outputs
- Low standby power – ICC = 10 μA
- Direct bit set/reset capability

GENERAL DESCRIPTION

The 82C55A is a high performance CMOS version of the industry standard 8255A and is manufactured using a self-aligned silicon gate CMOS process. It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The high performance and industry standard configuration of the 82C55A make it compatible with microprocessors, such as the 80286, 80186, 8086, 8088, 8080A, and 8085AH.

Static CMOS circuit design insures low operating power. TTL compatibility of VIH = 2.0 volts over the industrial temperature range and bus hold circuitry eliminate the need for pull-up resistors. AMD's advanced CMOS process results in performance equal to or greater than existing equivalent products at a fraction of the power.

BLOCK DIAGRAM

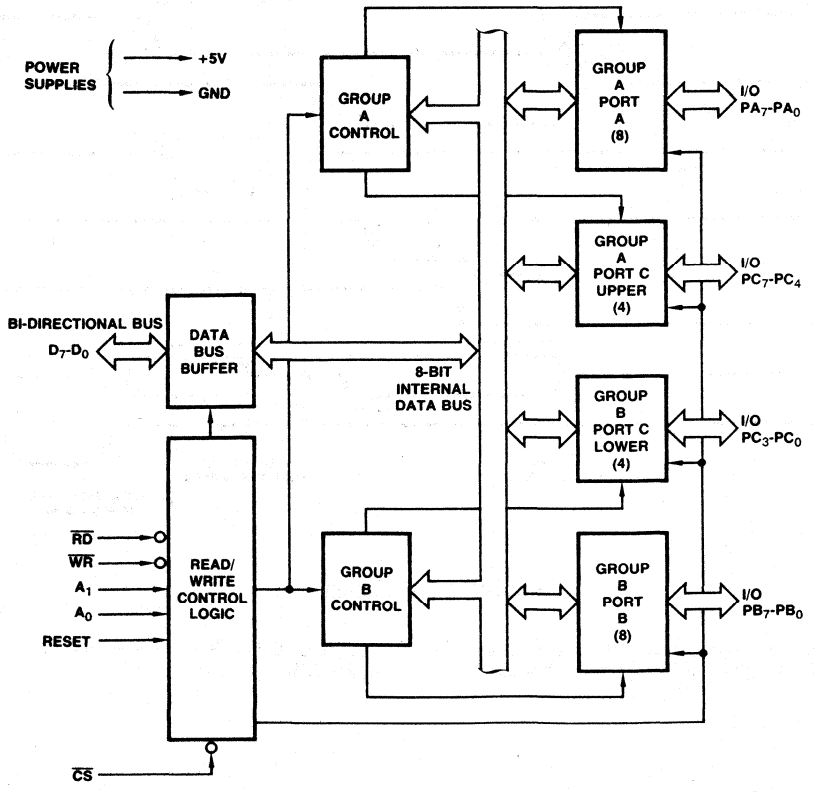


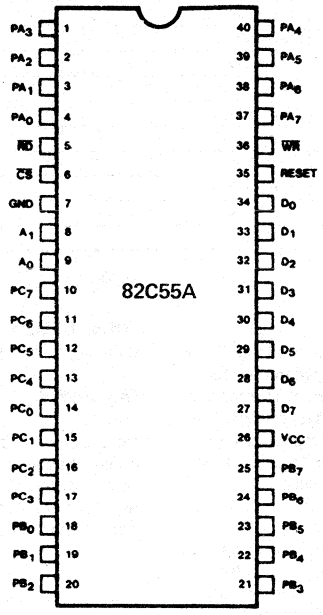
Figure 1.

06101A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-40-1, P-40-1

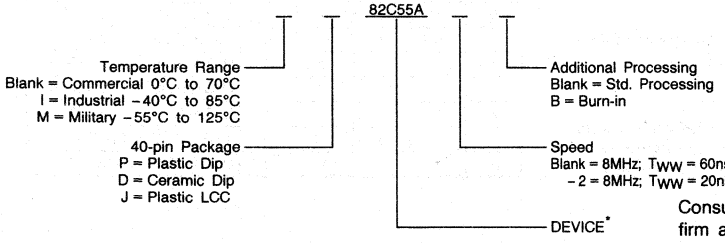


CD005702

Figure 2.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations	
82C55A	P, D, ID
82C55A-2	P, D, ID

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Pin Description
27-34	D_7-D_0	I/O	Data Bus (Bidirectional).
35	RESET	I	Reset Input.
6	$\overline{CS}$	I	Chip Select.
5	$\overline{RD}$	I	Read Input.
36	$\overline{WR}$	I	Write Input.
9, 8	A_0, A_1	I	Port Address.
37-40, 1-4	PA_7-PA_0	I/O	Port A (Bit).
25-18	PB_7-PB_0	I/O	Port B (Bit).
10-13, 17-14	PC_7-PC_0	I/O	Port C (Bit).
26	V_{CC}		+5 Volts.
7	GND		0 Volts.

DETAILED DESCRIPTION

General

The 82C55A is a programmable peripheral interface (PPI) device designed for use in microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 82C55A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and, in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "LOW" on this input pin enables the communication between the 82C55A and the CPU.

($\overline{RD}$)

Read. A "LOW" on this input pin enables the 82C55A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 82C55A.

($\overline{WR}$)

Write. A "LOW" on this input pin enables the CPU to write data or control words into the 82C55A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

82C55A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A → DATA BUS
0	1	0	1	0	PORT B → DATA BUS
1	0	0	1	0	PORT C → DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS → PORT A
0	1	1	0	0	DATA BUS → PORT B
1	0	1	0	0	DATA BUS → PORT C
1	1	1	0	0	DATA BUS → CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS → 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS → 3-STATE

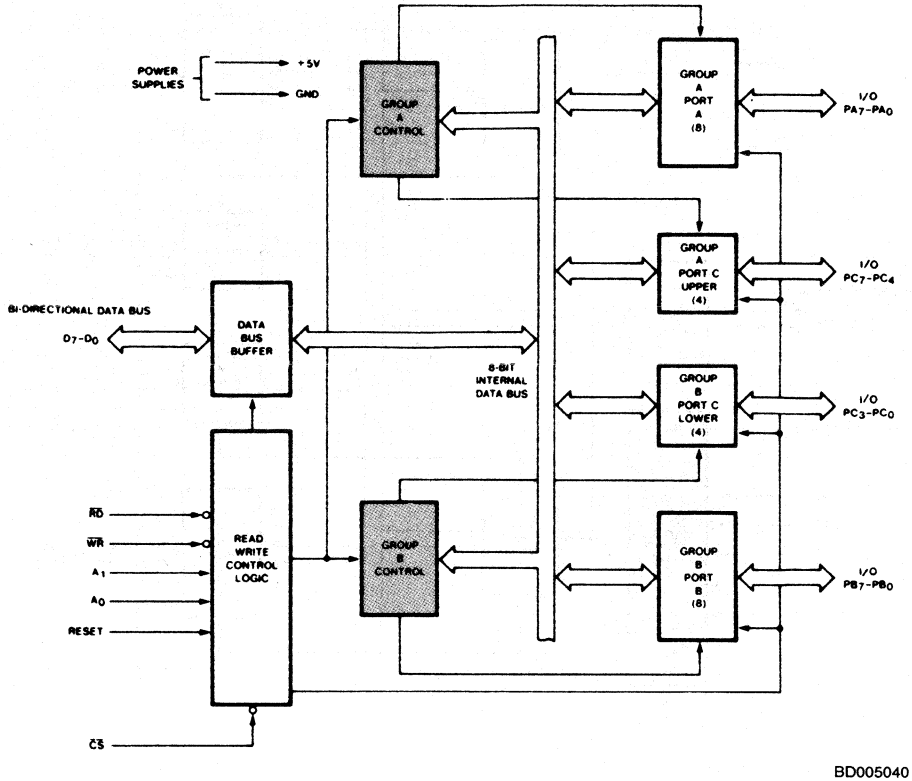


Figure 3. 82C55A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(RESET)

Reset. A "HIGH" on this input clears the control register, and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the system's software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information, such as "mode," "bit set," "bit reset," etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A — Port A and Port C upper (C7 – C4)

Control Group B — Port B and Port C lower (C3 – C0)

The Control Word Register can be both written and read as shown in the address decode table in the pin descriptions. Figure 6 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1," as this implies control word mode information.

Ports A, B, and C

The 82C55A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system's software, but each has its own special features or "personality" to further enhance the power and flexibility of the 82C55A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch. Both "pull-up" and "pull-down" bus hold devices are present on Port A.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer. Only "pull-up" bus hold devices are present on Port B.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B. Only "pull-up" bus hold devices are present on Port C.

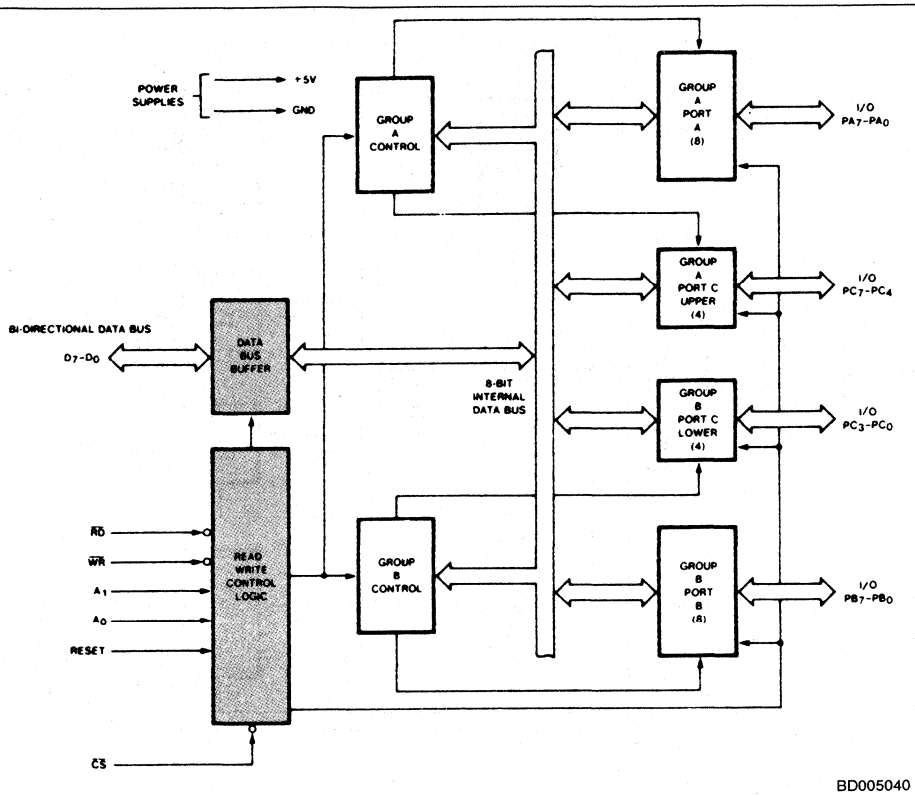


Figure 4. 82C55A Block Diagram Showing Group A and Group B Control Functions

OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 — Basic Input/Output
Mode 1 — Strobed Input/Output
Mode 2 — Bi-Directional Bus

When the reset input goes "HIGH" all ports will be set to the input mode (i.e., all 24 lines will be in the high-impedance state). After the reset is removed, the 82C55A can remain in the input mode with no additional initialization required. This eliminates the need for pull-up or pull-down resistors in "all CMOS" designs. During the execution of the system program, any of the other modes may be selected using a single output instruction. This allows a single 82C55A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance, Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results; Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

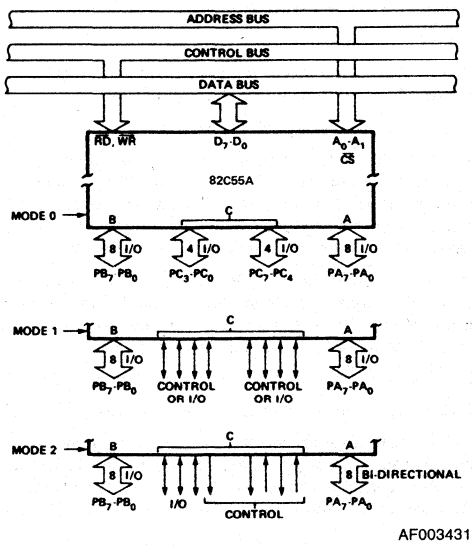


Figure 5. Basic Mode Definitions and Bus Interface

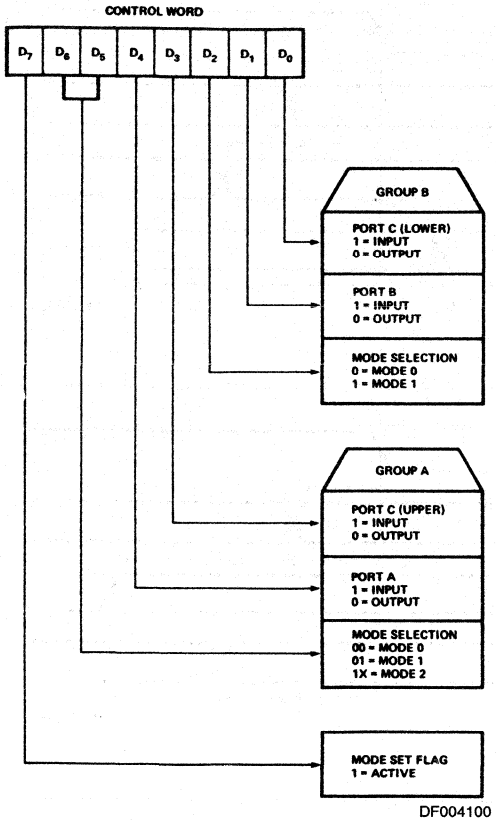


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first, but after a cursory review of the complete device operation, a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things, such as efficient PC board layout, control signal definition vs PC layout, and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

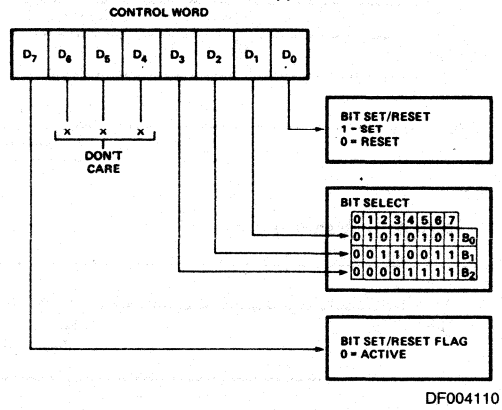


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 82C55A is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of Port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET — Interrupt enable
(BIT-RESET) — INTE is RESET — Interrupt disable

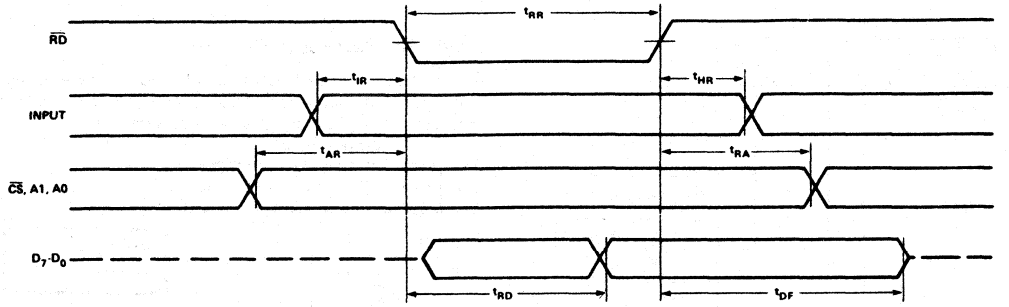
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

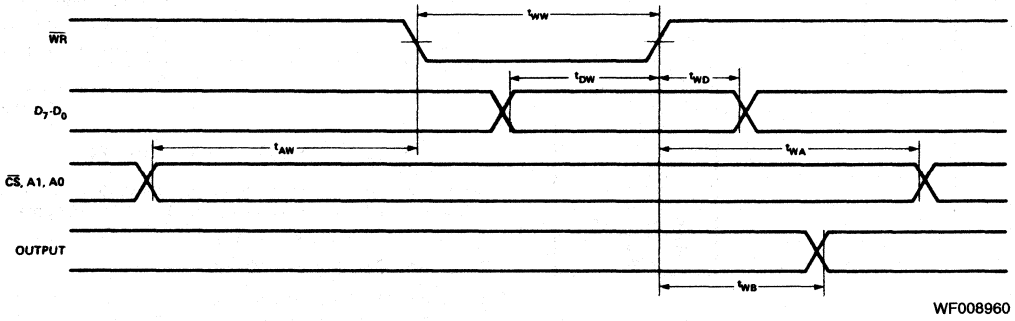
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required; data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



MODE 0 (Basic Input)



MODE 0 (Basic Output)

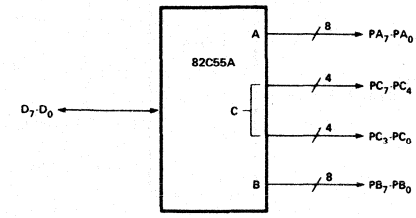
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

CONTROL WORD #0

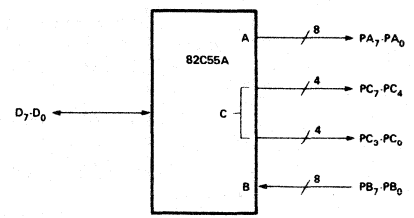
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



LS001461

CONTROL WORD #2

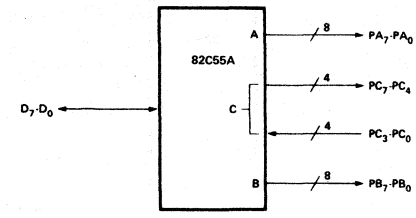
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



LS001471

CONTROL WORD #1

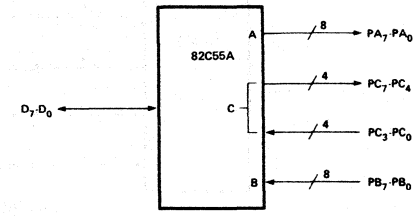
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



LS001481

CONTROL WORD #3

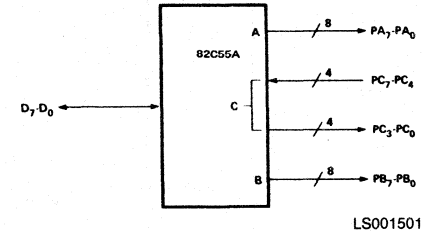
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



LS001491

CONTROL WORD #4

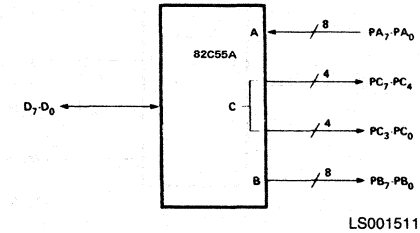
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



LS001501

CONTROL WORD #8

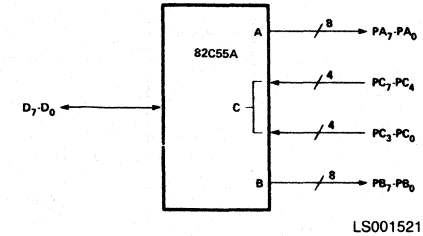
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



LS001511

CONTROL WORD #5

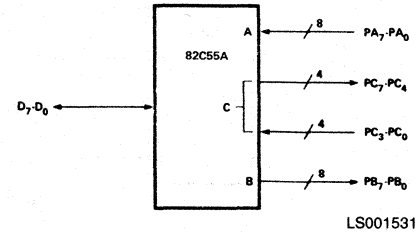
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



LS001521

CONTROL WORD #9

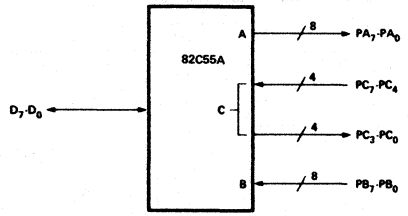
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



LS001531

CONTROL WORD #6

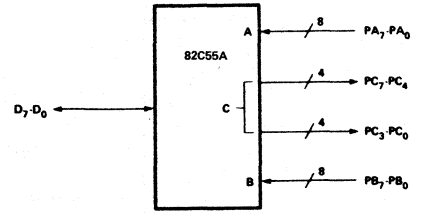
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



LS001541

CONTROL WORD #10

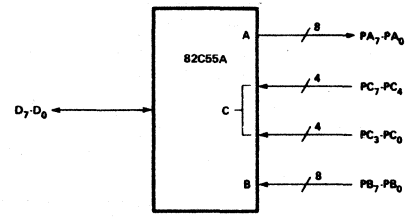
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



LS001551

CONTROL WORD #7

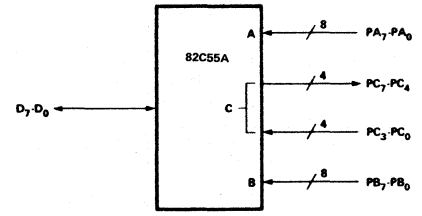
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



LS001561

CONTROL WORD #11

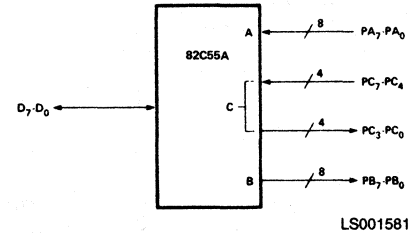
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



LS001571

CONTROL WORD #12

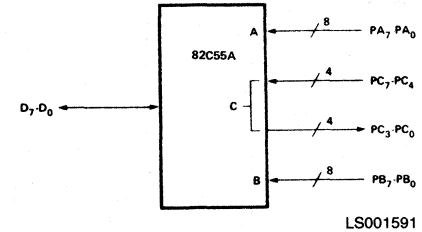
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0



LS001581

CONTROL WORD #14

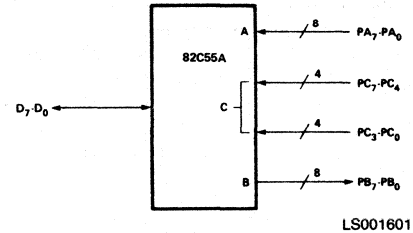
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	0



LS001591

CONTROL WORD #13

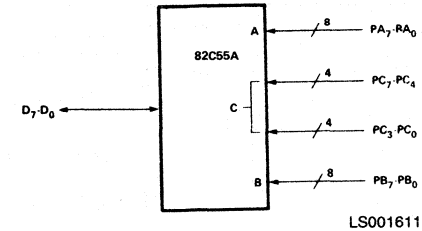
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	1



LS001601

CONTROL WORD #15

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	1



LS001611

06101A

Refer to page 7-1 for Essential Information on Military Devices

Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

STB (Strobe Input). A "LOW" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "HIGH" on this output indicates that the data has been loaded into the input latch — in essence, an acknowledgement. IBF is set by $\overline{STB}$ input being LOW and is reset by the rising edge of the $\overline{RD}$ input.

INTR (Interrupt Request)

A "HIGH" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set when the $\overline{STB}$ is a "one," IBF is a "one," and INTE is a "one." It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A
Controlled by bit set/reset of PC_4.

INTE B
Controlled by bit set/reset of PC_2.

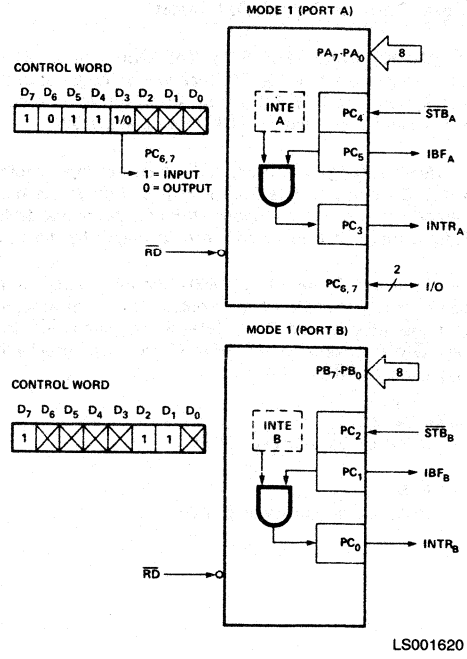


Figure 8. MODE 1 Input

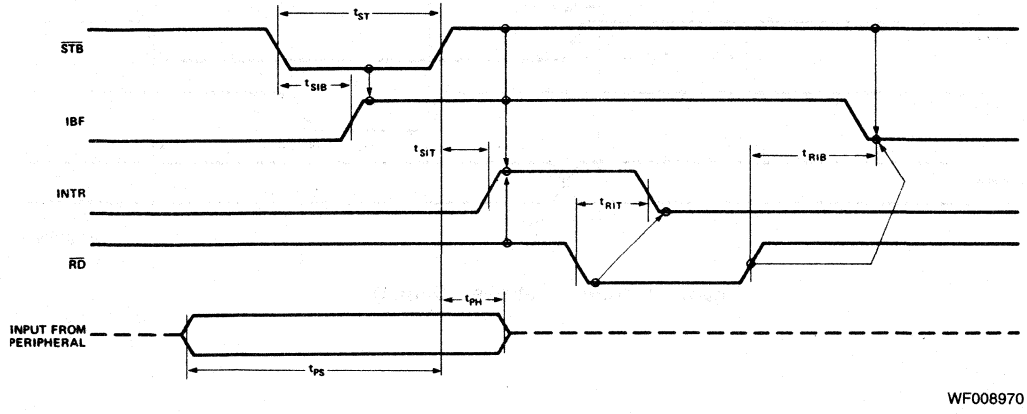


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "LOW" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ Input being LOW.

$\overline{ACK}$ (Acknowledge Input). A "LOW" on this input informs the 82C55A that the data from Port A or Port B has been accepted – in essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "HIGH" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one," $\overline{OBF}$ is a "one," and INTE is a "one." It is reset by the falling edge of $\overline{WR}$.

INTE A
Controlled by bit set/reset of PC_6.

INTE B
Controlled by bit set/reset of PC_2.

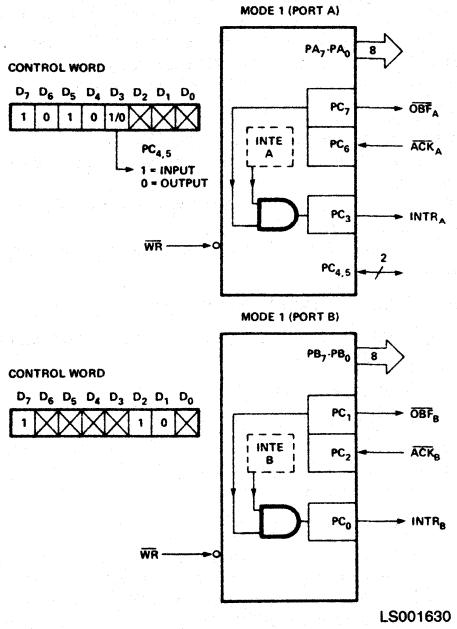


Figure 10. MODE 1 Output

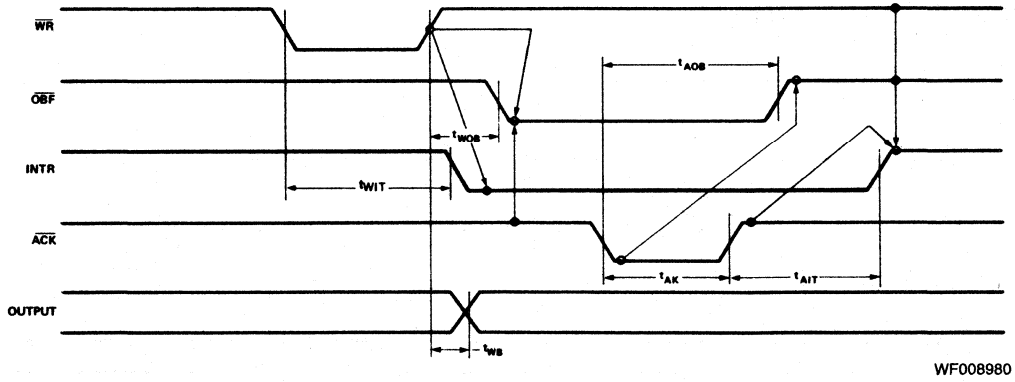


Figure 11. Mode 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

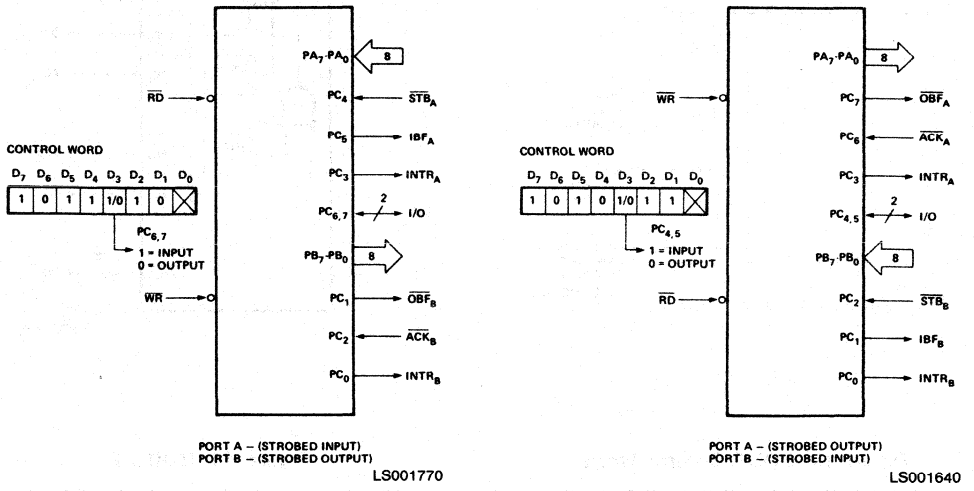


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A HIGH on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Ful). The $\overline{OBF}$ output will go "LOW" to indicate that the CPU has written data out to Port A.

$\overline{ACK}$ (Acknowledge). A "LOW" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high-impedance state.

INTE 1 (The INTE Flip-Flop Associated with $\overline{OBF}$). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "LOW" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "HIGH" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

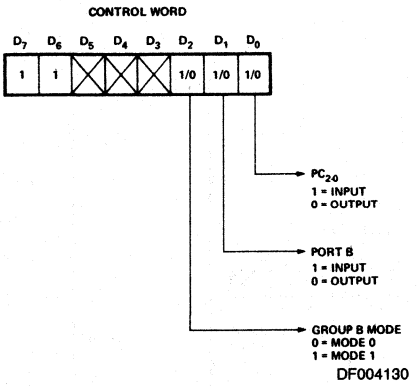


Figure 13. MODE Control Word

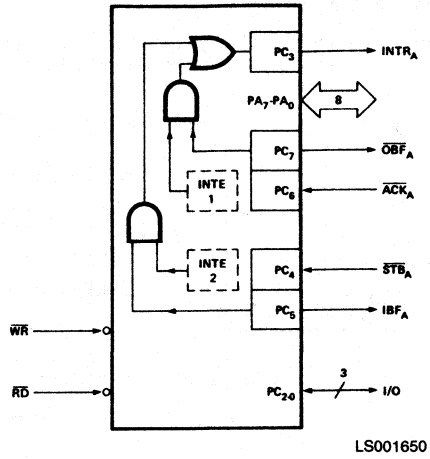


Figure 14. MODE 2

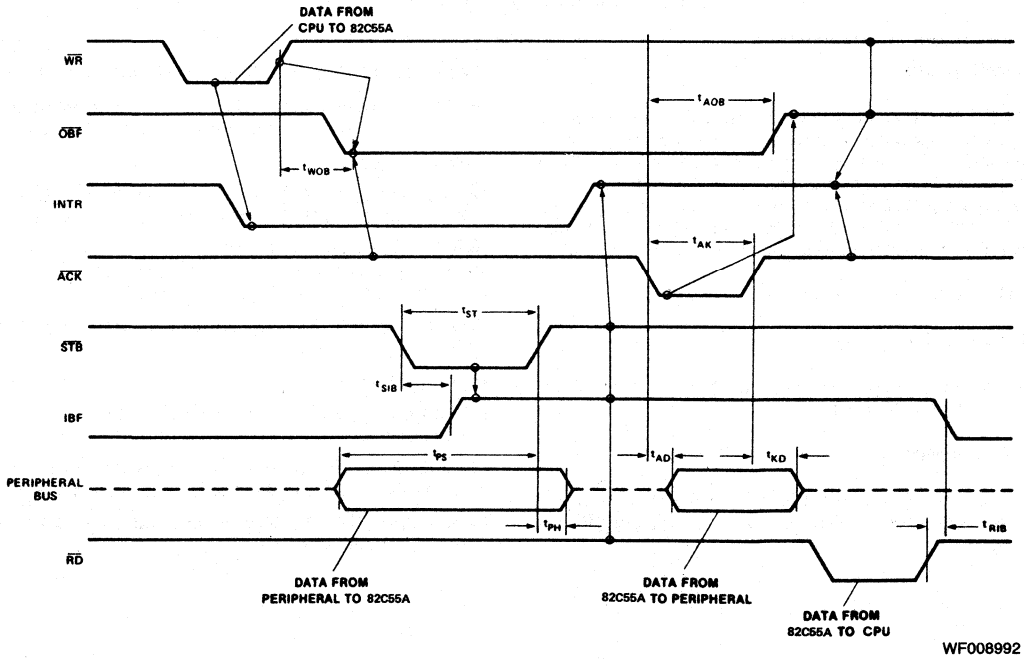
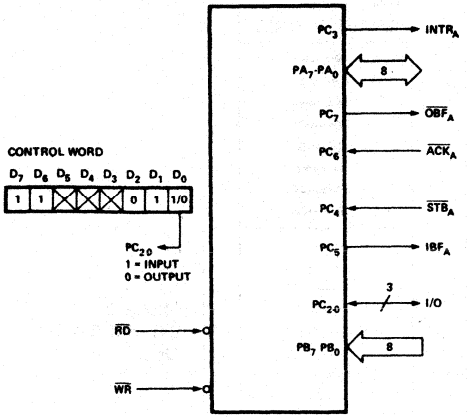


Figure 15. MODE 2 (Bidirectional)

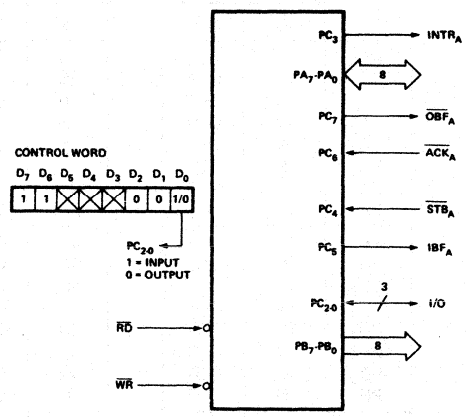
Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible (INTR = IBF · $\overline{MASK}$ · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · $\overline{MASK}$ · $\overline{ACK}$ · $\overline{WR}$).

MODE 2 AND MODE 0 (INPUT)



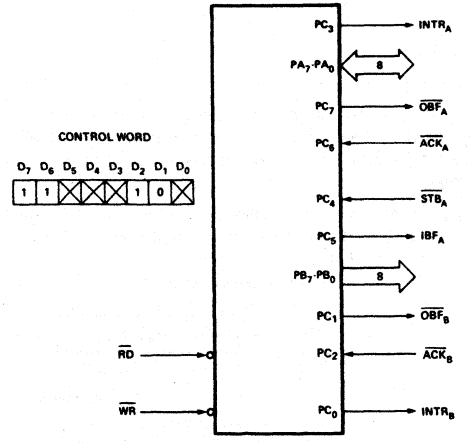
LS001671

MODE 2 AND MODE 0 (OUTPUT)



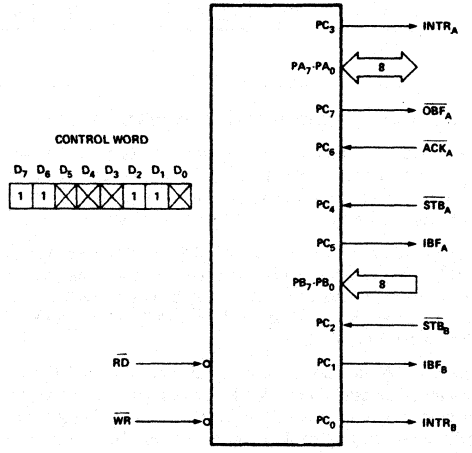
LS001661

MODE 2 AND MODE 1 (OUTPUT)



LS001681

MODE 2 AND MODE 1 (INPUT)



LS001691

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines, will be placed on the data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 18.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including INTR, IBF and $\overline{OBF}$) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including $\overline{ACK}$ and $\overline{STB}$ lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flag, as illustrated in Figure 18.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts "handshaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

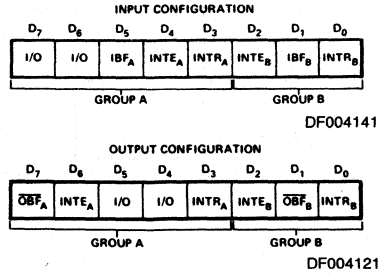


Figure 17. MODE 1 Status Word Format

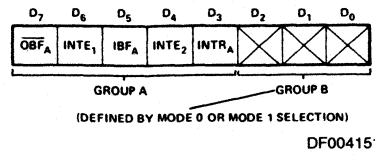


Figure 17a. MODE 2 Status Word Format

Interrupt Enable Flag	Position	Alternate Port C Pin Signal (Mode)
INTE B	PC2	$\overline{ACK}_B$ (Output Mode 1) or $\overline{STB}_B$ (Input Mode 1)
INTE A2	PC4	$\overline{STB}_A$ (Input Mode 1 or Mode 2)
INTE A1	PC6	$\overline{ACK}_A$ (Output Mode 1 or Mode 2)

Figure 18. Interrupt Enable Flags in Modes 1 and 2

APPLICATIONS INFORMATION

The 82C55A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 82C55A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 82C55A to exactly "fit" the application. Figures 19 through 25 present a few examples of typical applications of the 82C55A.

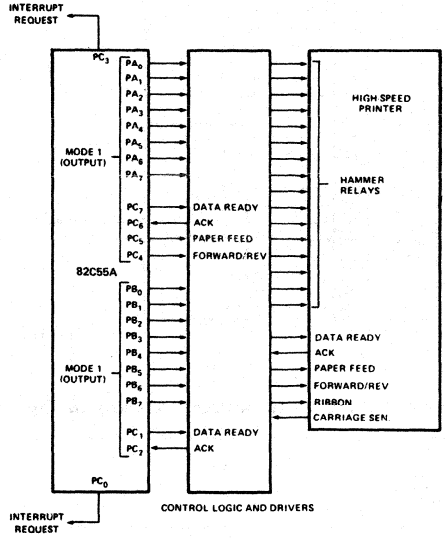
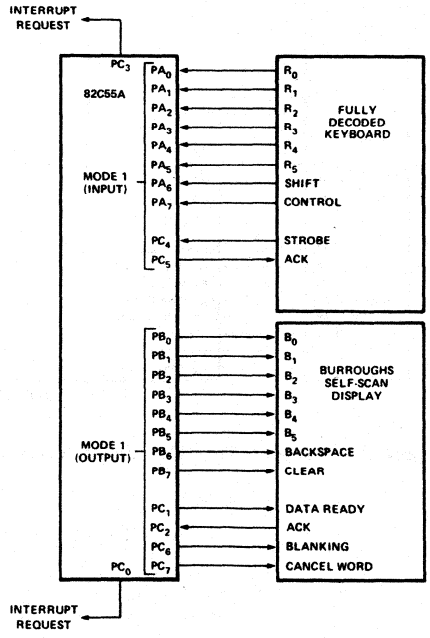
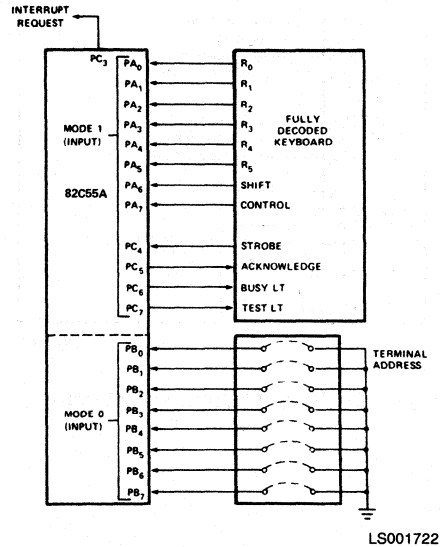


Figure 19. Printer Interface LS001712



LS001702

Figure 20. Keyboard and Display Interface



LS001722

Figure 21. Keyboard and Terminal Address Interface

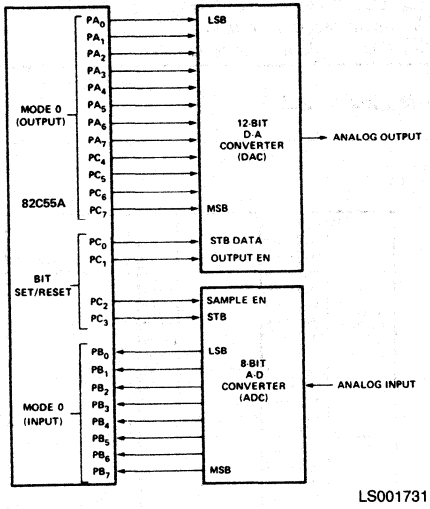


Figure 22. Digital to Analog, Analog to Digital

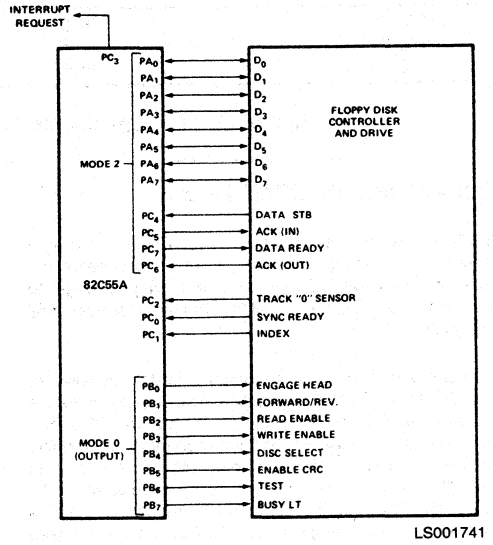


Figure 23. Basic CRT Controller Interface

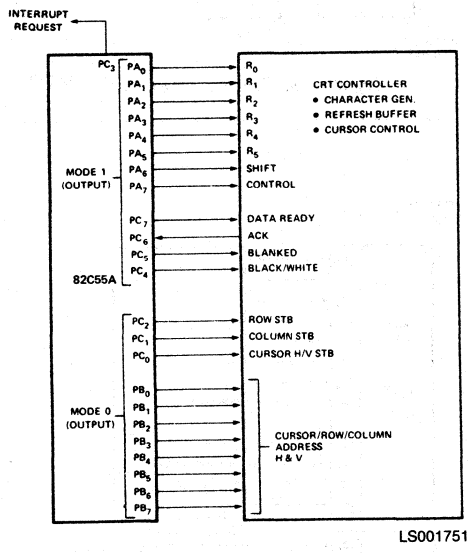


Figure 24. Basic Floppy Disc Interface

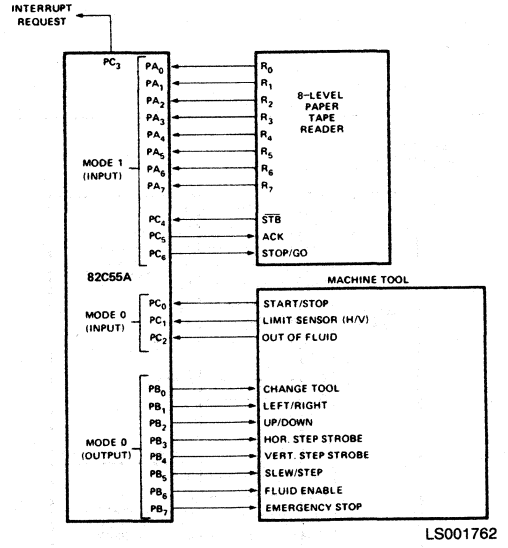


Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages with
Respect to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 1.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
82C55A-2, 82C55A	0°C to 70°C	5V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

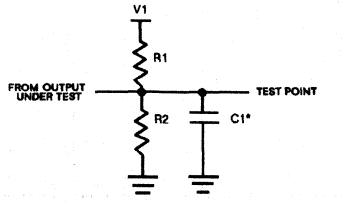
Parameters	Description	Test Conditions	Min	Max	Units
VIH	Logical One Input Voltage		2.0	V_{CC}	V
VIL	Logical Zero Input Voltage		−.5	0.8	V
VOH	Logical One Output Voltage	IOH = −2.5mA IOH = −100μA	3.0 VCC − 0.4		V V
VOL	Logical Zero Output Voltage	IOL = +2.5mA		0.4	V
IIL	Input Leakage Current	OV ≤ VIN ≤ VCC	−1.0	±10	μA
IO	I/O Pin Leakage Current	OV ≤ VO ≤ VCC	−10.0	10.0	μA
IBHH	Bus Hold High Leakage Current	VO = 3.0V Ports A, B, C	−50	−300	μA
IBHL	Bus Hold Low Leakage Current	VO = 1.0V Port A Only	+50	+300	μA
IDAR	Darlington Drive Current	Ports A, B, C Test Condition 3	−2.0		mA
ICC	Power Supply Current	VCC = 5.5V VIN = VCC or GND Outputs Open		10	μA

CAPACITANCE T_A = 25°C; VCC = GND = 0V; VIN = +5V or GND

Parameters	Description	Test Conditions	Min	Max	Units
CIN*	Input Capacitance	FREQ = 1 MHz Unmeasured pins returned to GND		10	pf
CI/O*	I/O Pin Capacitance			20	pf

* Guaranteed and sampled, but not 100% tested

SWITCHING TEST CIRCUIT

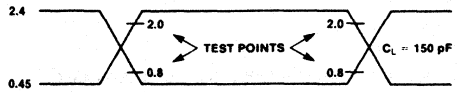


TC002170

TEST CONDITION DEFINITION TABLE

TEST CONDITION	V1	R1	R2	C1
1	1.7V	523Ω	OPEN	150 pf
2	5.0V	2KΩ	1.7KΩ	50 pf
3	1.5V	750Ω	OPEN	OPEN

SWITCHING TEST INPUT WAVEFORM



WF009000

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Bus Parameters

READ

Parameters	Description	Test Conditions	Min	Max	Units
tAR	Address Stable Before READ		0		ns
tRA	Address Stable After READ		0		ns
tRR	READ Pulse Width		150		ns
tRD	Data Valid From READ	1		100	ns
tDF	Data Float After READ	2	10	75	ns
tRV	Time Between READs and/or WRITE		300		ns

WRITE

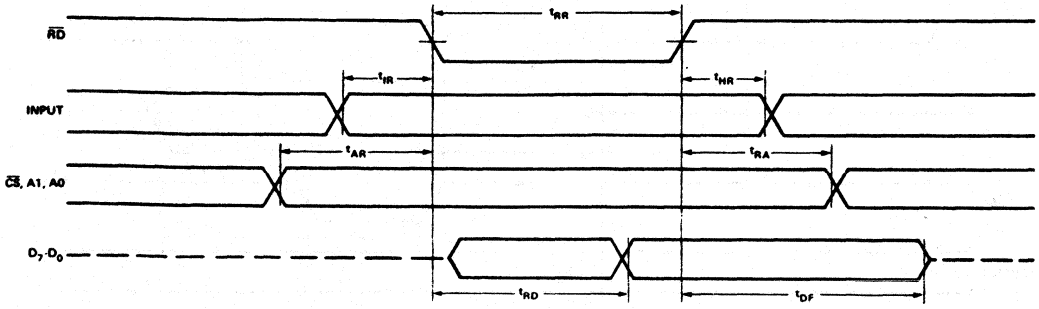
Parameters	Description	Test Conditions	Min	Max	Units
tAW	Address Stable Before WRITE		0		ns
tWA	Address Stable After WRITE	82C55A-2: Ports A, B, C	20		ns
		82C55A: Ports A, B	20		ns
		82C55A: Port C	60		ns
tWW	WRITE Pulse Width		100		ns
tDW	Data Valid to WRITE High		100		ns
tWD	Data Valid After WRITE High	82C55A-2: Ports A, B, C	30		ns
		82C55A: Ports A, B	30		ns
		82C55A: Port C	60		ns

OTHER TIMINGS

Parameters	Description	Test Conditions	Min	Max	Units
tWB	WR = 1 to Output	1		350	ns
tIR	Peripheral Data Before RD		0		ns
tHR	Peripheral Data After RD		0		ns
tAK	ACK Pulse Width		100		ns
tST	STB Pulse Width		100		ns
tPS	Per. Data Before STB High		20		ns
tPH	Per. Data After STB High		50		ns
tAD	ACK = 0 to Output	1		175	ns
tKD	ACK = 1 to Output Float	2	20	250	ns
tWOB	WR = 1 to OBF = 0	1		150	ns
tAOB	ACK = 0 to OBF = 1	1		150	ns
tSIB	STB = 0 to IBF = 1	1		150	ns
tRIB	RD = 1 to IBF = 0	1		150	ns
tRIT	RD = 0 to INTR = 0	1		200	ns
tSIT	STB = 1 to INTR = 1	1		150	ns
tAIT	ACK = 1 to INTR = 1	1		150	ns
tWIT	WR = 0 to INTR = 0	1		200	ns
tRES	Reset Pulse Width	see Note 1	500		ns

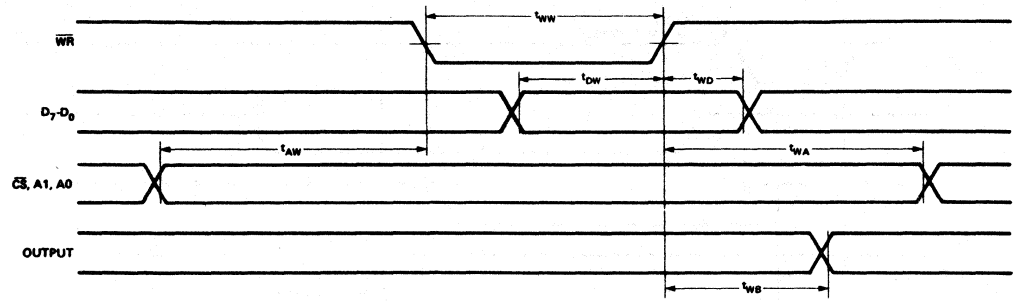
Note 1: Period of initial Reset pulse after power-on at least 50usec. Subsequent Reset pulses may be 500ns minimum.

SWITCHING WAVEFORMS



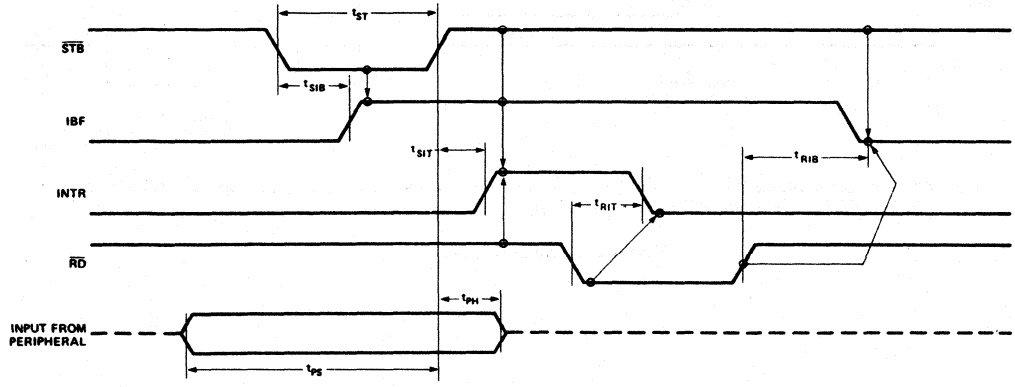
WF009010

Mode 0 (Basic Input)



WF009020

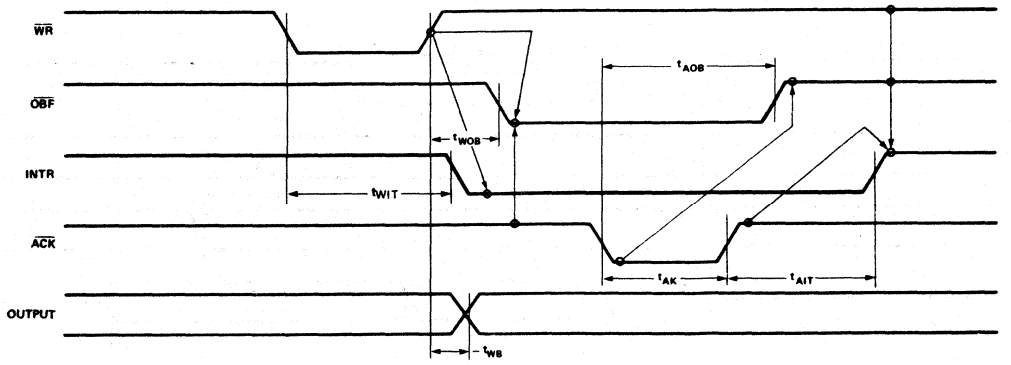
Mode 0 (Basic Output)



WF009030

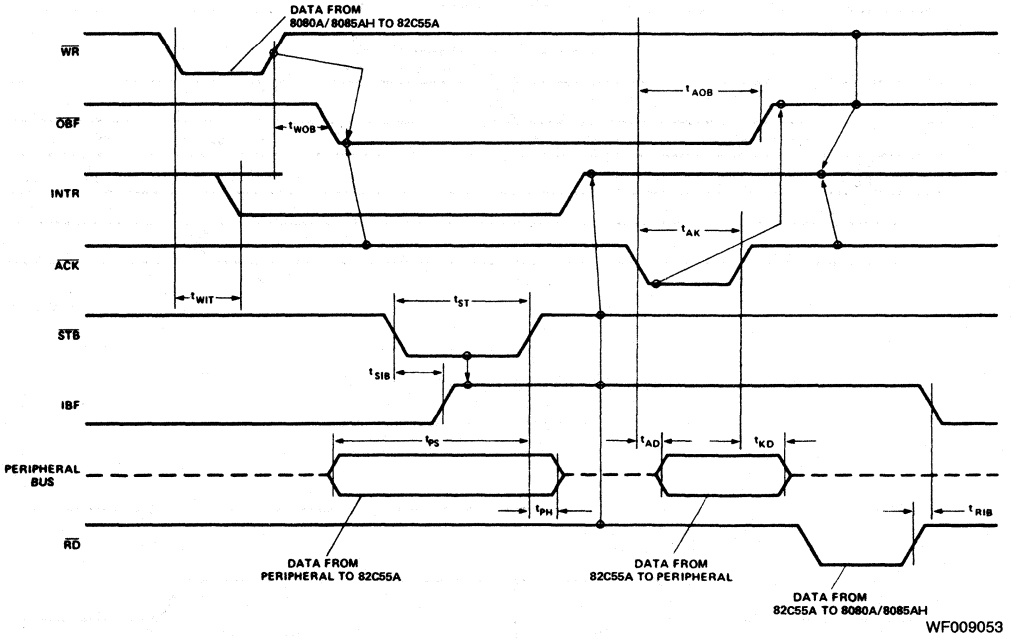
Mode 1 (Strobed Input)

SWITCHING WAVEFORMS (Cont.)



WF009040

Mode 1 (Strobed Output)



WF009053

Note: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible (INTR = IBF · $\overline{MASK}$ · $\overline{STB}$ · $\overline{RD}$ + $\overline{OBF}$ · $\overline{MASK}$ · $\overline{ACK}$ · $\overline{WR}$).

Mode 2 (Bidirectional)

8259A

Programmable Interrupt Controller
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Eight-Level Priority Controller
- Expandable to 64 Levels
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Single +5V Supply (No Clocks)
- 28-Pin Dual-In-Line Package

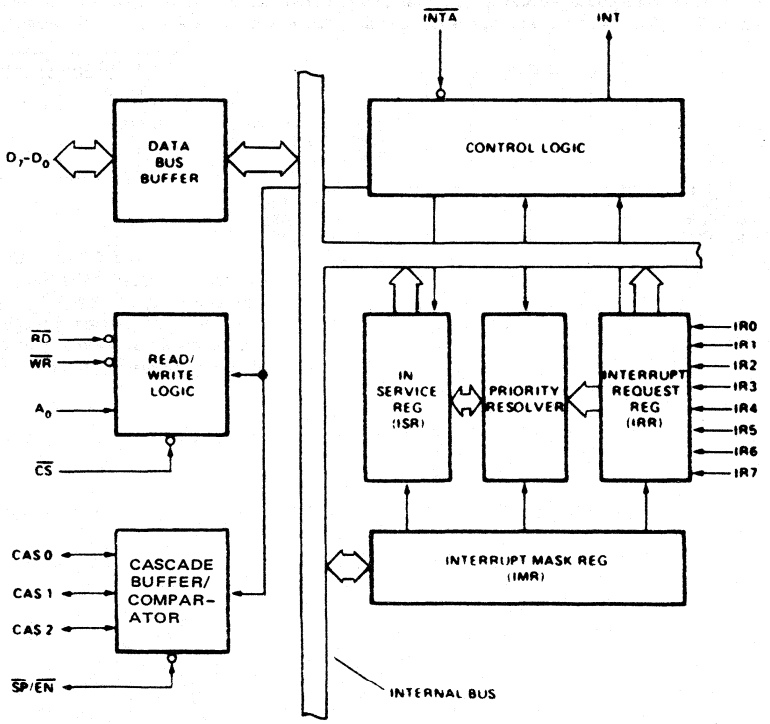
GENERAL DESCRIPTION

The 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes.

BLOCK DIAGRAM



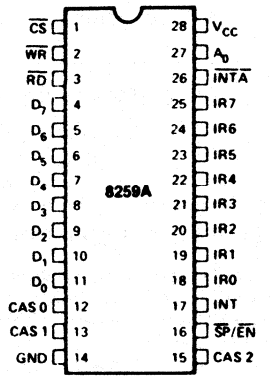
BD003540

Figure 1.

06102A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

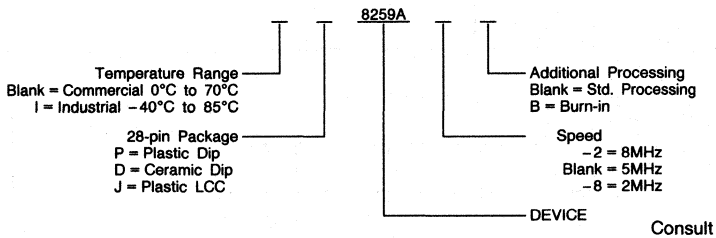


CD005640

Figure 2.
Also available in PLCC. See section 7 for pinout details.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8259A-2 8259A 8259A-8	P, D, ID
8259A-2 8259A	/BXA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}	I	**Supply:** +5V Supply.
14	GND	I	**Ground.**
1	$\overline{CS}$	I	**Chip Select:** A low on this pin enables $\overline{AD}$ and $\overline{WR}$ communication between the CPU and the 8259A. INTA functions are independent of $\overline{CS}$.
2	$\overline{WR}$	I	**Write:** A low on this pin when $\overline{CS}$ is low enables the 8259A to accept command words from the CPU.
3	$\overline{RD}$	I	**Read:** A low on this pin when $\overline{CS}$ is low enables the 8259A to release status onto the data bus for the CPU.
4–11	D_7–D_0	I/O	**Bidirectional Data Bus:** Control, status and interrupt-vector information are transferred via the bus.
12, 13, 15	CAS_0–CAS_2	I/O	**Cascade Lines:** The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A abd inputs for a slave 8259A.
16	$\overline{SP}/\overline{EN}$	I/O	**Slave Program/Enable Buffer:** This is a dual function pin. When in the Buffered Mode, it can be used as an output to control buffer transceivers (EN). When not in the buffered mode, it is used as an input to designate a master (SP = 1) or slave (SP = 0).
17	INT	O	**Interrupt:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
18–25	IR_0–IR_7	I	**Interrupt Requests:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
26	$\overline{INTA}$	I	**Interrupt Acknowledge:** This pin is used to enable 8259A Interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
27	A_0	I	**AO Address Line:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for iAPX 86, 88).

DETAILED DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I/O devices, such as keyboards, displays, sensors and other components, receive servicing in an efficient manner, so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

The 8259A

The 8259A is a device specifically designed for use in real time, interrupt-driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

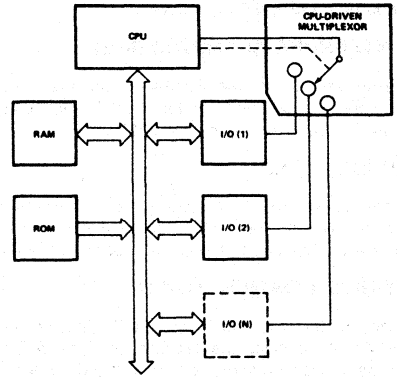
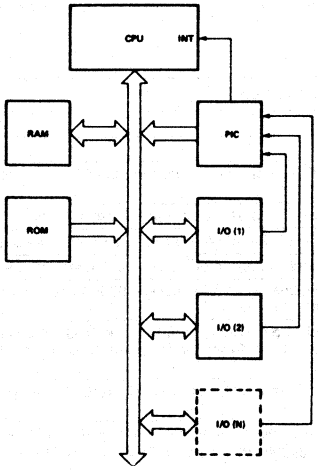


Figure 3a. Polled Method



AF003330

Figure 3b. Interrupt Method

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085AH and 8086 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

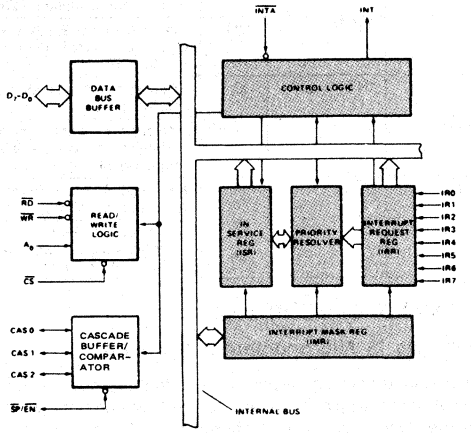
A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

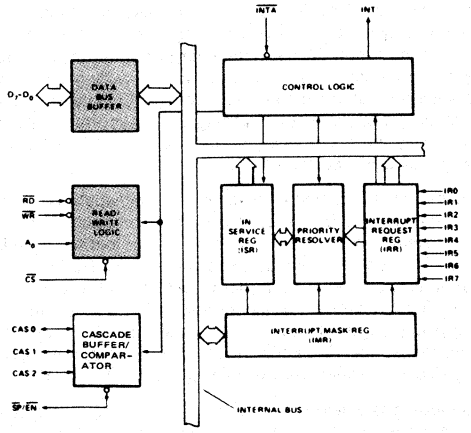
$\overline{RD}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.



BD003540

Figure 4a. 8259A Block Diagram



BD003540

Figure 4b. 8259A Block Diagram

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins

(CAS0 – 2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0 – 2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 8259A.")

Interrupt Sequence

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an 8080A/85AH system:

1. One or more of the INTERRUPT REQUEST lines (IR7 – 0) are raised high, setting the corresponding IRR bit(s).

2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.

3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7 – 0 pins.

5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 8259A from the CPU group.

6. These two $\overline{INTA}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.

7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occurring in an 8086 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.

5. The 8086 will initiate a second $\overline{INTA}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.

6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

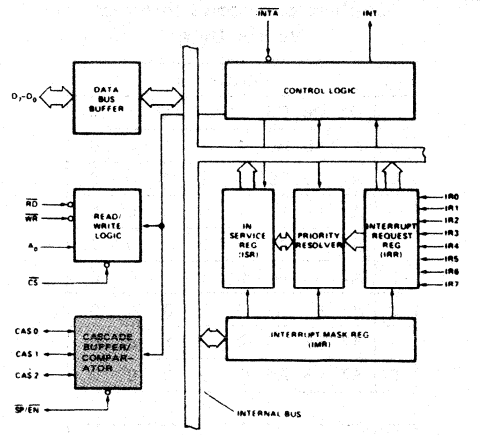


Figure 4c. 8259A Block Diagram

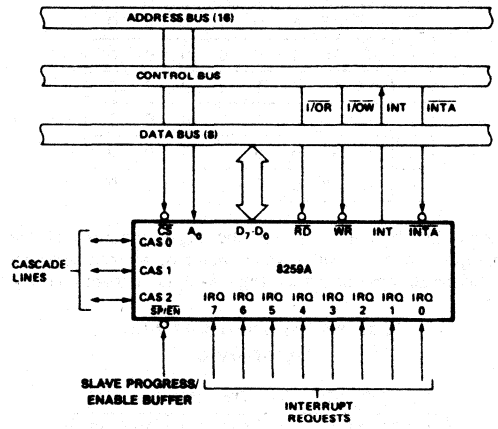


Figure 5. 8259A Interface to Standard System Bus

Interrupt Sequence Outputs

8080A/85AH

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse, the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4, bits $A_5 - A_7$ are programmed, while $A_0 - A_4$ are automatically inserted by the 8259A. When Interval = 8, only A_6 and A_7 are programmed, while $A_0 - A_5$ are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	S5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third INTA pulse, the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence ($A_8 - A_{15}$), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

8086, 8088

8086 mode is similar to 8080A mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of 8080A/85AH systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the INTA pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 8086 mode, the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and $A_5 - A_{11}$ are unused in 8086 mode):

Content of Interrupt Vector Byte for 8086 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING INFORMATION

The 8259A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 8259A in the system must be brought to a starting point – by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses.
2. *Operation Command Words (OCWs):* These are the command words which command the 8259A to operate in various interrupt modes. These modes are:

   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259A anytime after initialization.

Initialization Command Words (ICWS)

GENERAL

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
b. The Interrupt Mask Register is cleared.
c. IR7 input is assigned priority 7.
d. The slave mode address is set to 7.
e. Special Mask Mode is cleared and Status Read is set to IRR.
f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, 8080A/85AH system).

***Note: Master/Slave in ICW4 is only used in the buffered mode.**

Initialization Command Words 1 and 2 (ICW1, ICW2)

$A_5 - A_{15}$: *Page starting address of service routines.* In an 8080A/85AH system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long ($A_0 - A_{15}$). When the routine interval is 4, $A_0 - A_4$ are automatically inserted by the 8259A, while $A_5 - A_{15}$ are programmed externally. When the routine interval is 8, $A_0 - A_5$ are automatically inserted by the 8259A, while $A_6 - A_{15}$ are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an 8086 system $A_{15} - A_{11}$ are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. $A_{10} - A_5$ are ignored and ADI (Address interval) has no effect.

LTIM: If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set – ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

Initialization Command Word 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for 8080A/85AH system) and will enable the corresponding slave to release bytes 2 and 3 (for 8086 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP} = 0$, or if BUF = 1 and M/S = 0 in ICW4) bits 2 – 0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 8086 are released by it on the Data Bus.

Initialization Command Word 4 (ICW4)

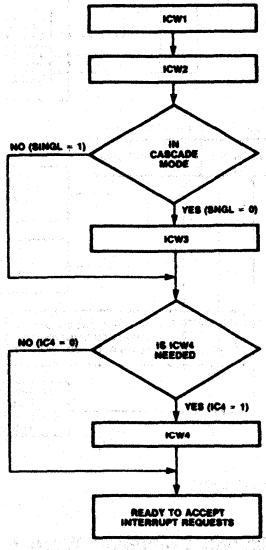
SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

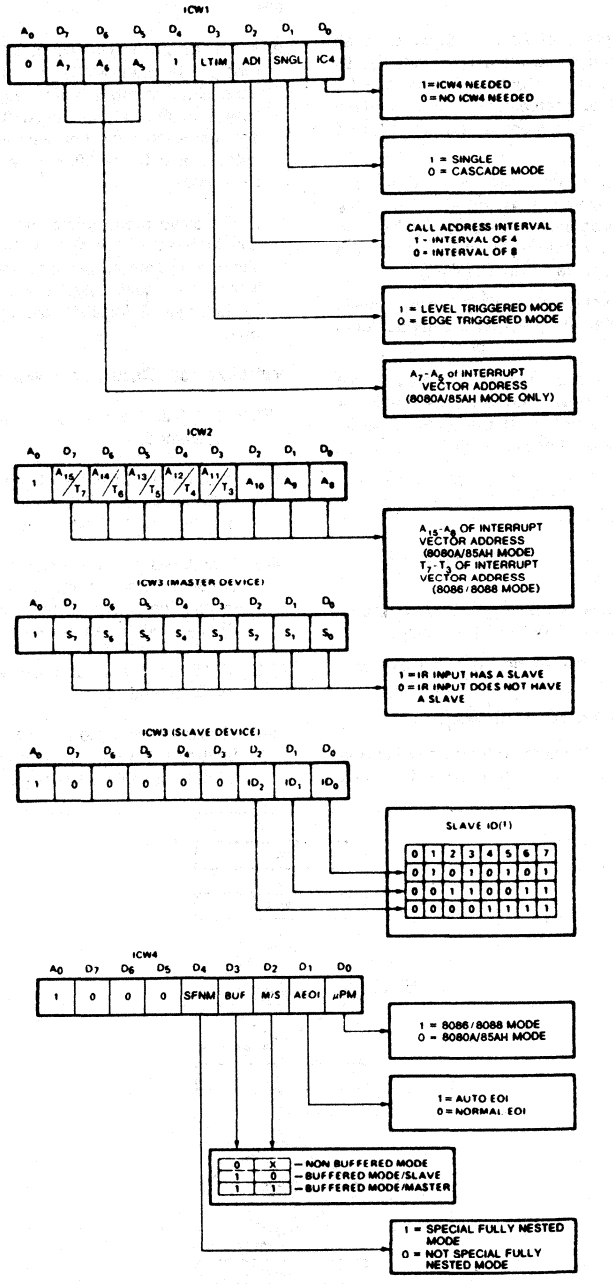
AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for 8080A/85AH system operation, μPM = 1 sets the 8259A for 8086 system operation.



PF001310

Figure 6. Initialization Sequence



DF003911

Note 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.

Figure 7. Initialization Command Word Format

Operation Command Words (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

Operation Control Words (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SL	EO1	0	0	L2	L1	L0

OCW3

0	0	ESMM	SMM	0	1	P	RR	RIS

Operation Control Word 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). $M_7 - M_0$ represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

Operation Control Word 2 (OCW2)

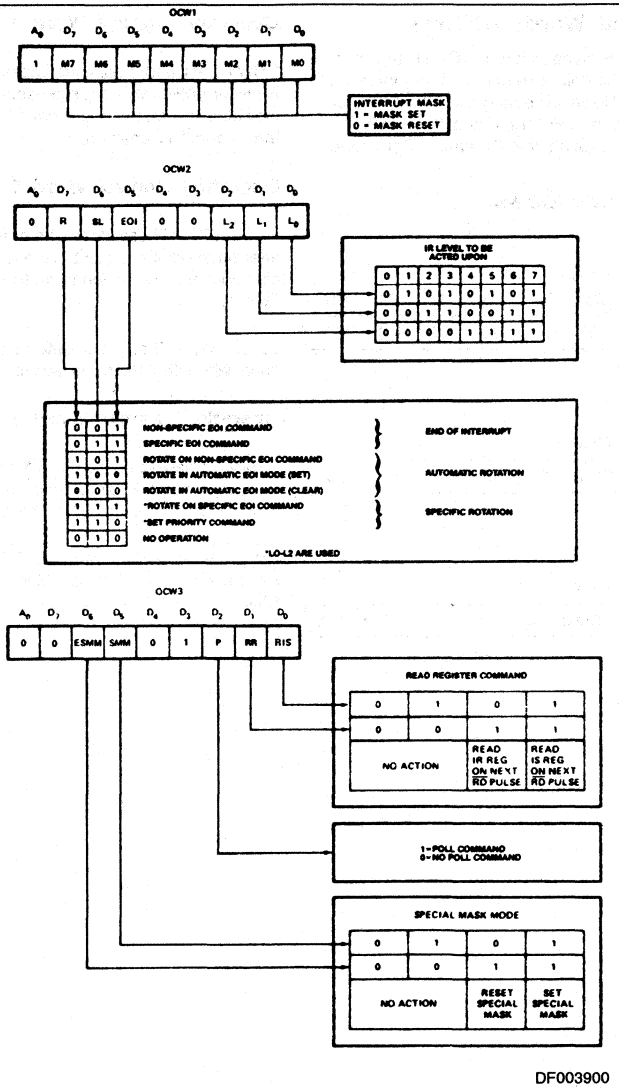
R, SL, EOI – These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0 – These bits determine the interrupt level acted upon when the SL bit is active.

Operation Control Word 3 (OCW3)

ESMM – Enable Special Mask Mode. When this bit is set to 1, it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a "don't care."

SMM – Special Mask Mode. If ESMM = 1 and SMM = 1, the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0, the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.



DF003900

Figure 8. Operation Command Word Format

Fully Nested Mode

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

End Of Interrupt (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{\text{INTA}}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is

issued the 8259A will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and LO – L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

Automatic End Of Interrupt (AEOI) Mode

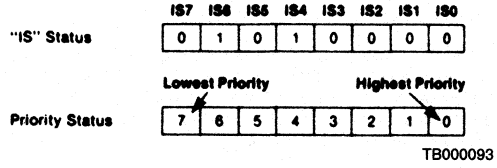
If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in 8080A/85AH, second in 8086). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

The AEOI mode can only be used in a master 8259A and not a slave.

AUTOMATIC ROTATION (Equal Priority Devices)

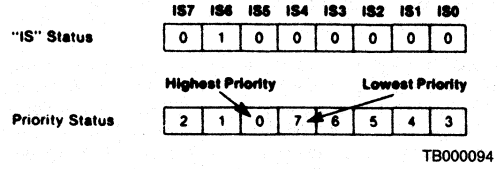
In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)



TB000093

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



TB000094

There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

SPECIFIC ROTATION (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1; LO – L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO – L2 = IR level to receive bottom priority).

Interrupt Masks

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

Poll Command

In this mode the INT output is not used or the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

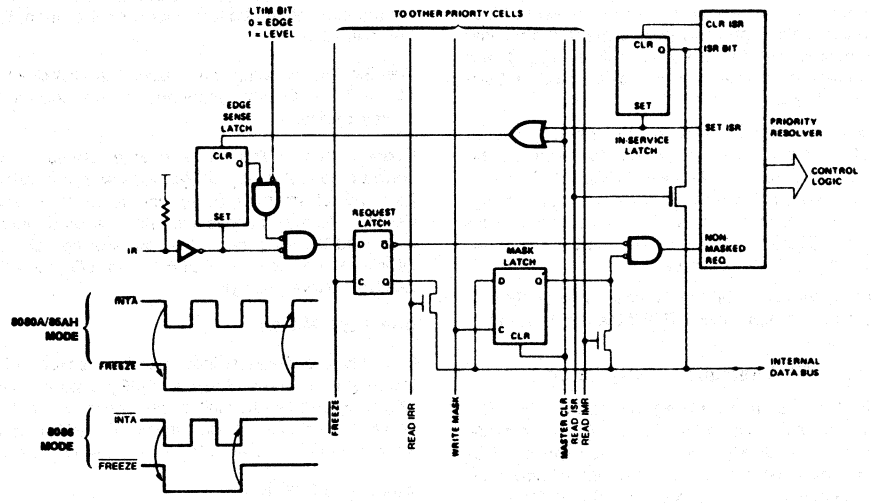
The Poll command is issued by setting P = "1" in OCW3. The 8259A treats the next $\overline{RD}$ pulse to the 8259A (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	–	–	–	–	W2	W1	W0

W0 – W2: Binary code of the highest priority level requesting service.
I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.



LS001361

NOTES1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/ IS ACTIVE DURING $\overline{\text{INTA}}$/ AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	D_i	Q_i	FOLLOW
0	X	Qn-1	HOLD

Figure 9. Priority Cell – Simplified Logic Diagram

Reading The 8259A Status

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization, the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{\text{RD}}$ is active and AO = 1 (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

Edge and Level Triggered Modes

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by low to high transition on an IR input. The IR input can remain high without generating another interrupt.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes, the IR inputs must remain high until after the falling edge of the first INTA. If the IR input goes low before this time, a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature, the IR7 routine is used for "clean up" – simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes, a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs, it is a default.

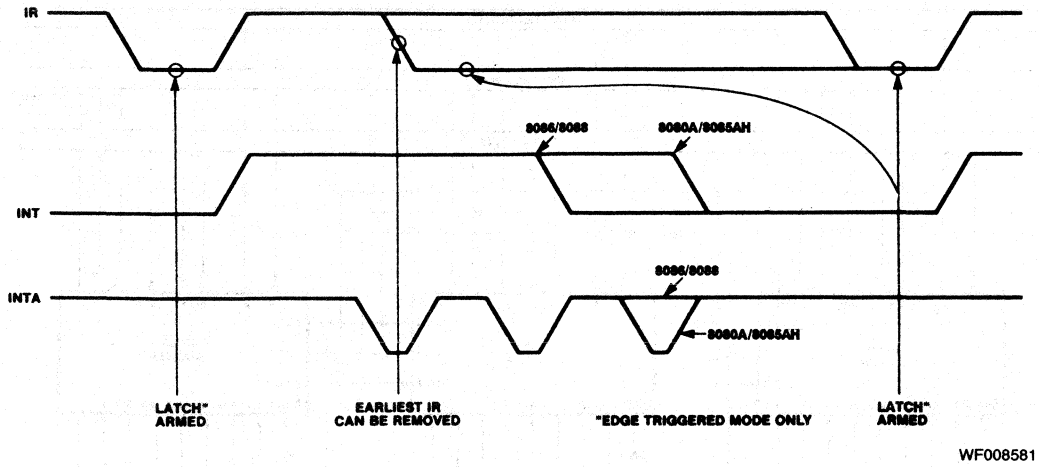


Figure 10. IR Triggering Timing Requirements

The Special Fully Nested Mode

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode, a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

Buffered Mode

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

Cascade Mode

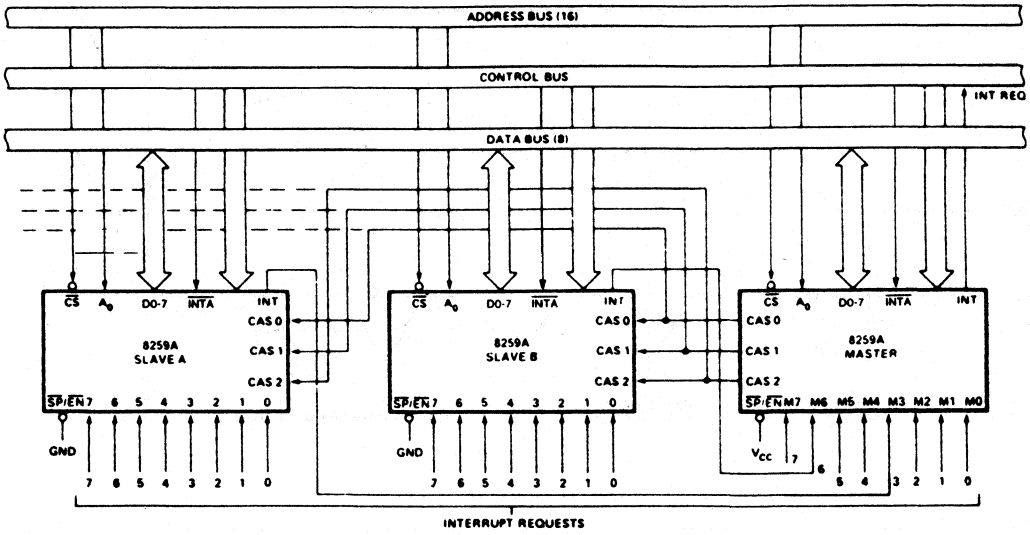
The 8259A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{INTA}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of INTA. (Byte 2 only for 8086/8088).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first INTA pulse to the trailing edge of the third pulse. Each 8259A in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each 8259A.

The cascade lines of the Master 8259A are activated only for slave inputs, non slave inputs leave the cascade line inactive (low).



AF003310

Figure 11. Cascading the 8259A

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1 Watt

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
8259A-2 82A59A	0°C to 70°C	5V ±10%
8259A-8	0°C to 70°C	5V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over Operating Ranges)

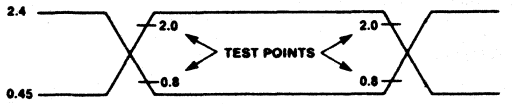
Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	0.8	V
V_{IH}	Input High Voltage		2.0*	V_{CC} + 0.5V	V
V_{OL}	Output Low Voltage	I_{OL} = 2.2mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
$V_{OH(INT)}$	Interrupt Output High Voltage	I_{OH} = −100μA	3.5		V
		I_{OH} = −400μA	2.4		V
I_{LI}	Input Load Current	$0V \leq V_{IN} \leq V_{CC}$	−10	+10	μA
I_{LOL}	Output Leakage Current	$0.45V \leq V_{OUT} \leq V_{CC}$	−10	+10	μA
I_{CC}	V_{CC} Supply Current			85	mA
I_{LIR}	IR Input Load Current	V_{IN} = 0		−300	μA
		$V_{IN} = V_{CC}$		10	μA

*Note: For extended Temperature EXPRESS V_{IH} = 2.3V.

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

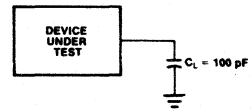
Parameters	Description	Test Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	fc = 1MHz			10	pF
$C_{I/O}$	I/O Capacitance	Unmeasured pins returned to V_{SS}			20	pF

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007100

SWITCHING TEST LOAD CIRCUIT



TC001840

Note: AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0." Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."

C_L = 100 pF.
C_L includes jig capacitance.

SWITCHING CHARACTERISTICS
TIMING REQUIREMENTS

Parameters	Description	Test Conditions	8259A-8		8259A		8259A-2		Units
			Min	Max	Min	Max	Min	Max	
TAHRL	AO/$\overline{CS}$ Set-up to $\overline{RD}$/$\overline{INTA}$↓		50		0		0		ns
TRHAX	AO/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$↑		5		0		0		ns
TRLRH	$\overline{RD}$ Pulse Width		420		235		160		ns
TAHWL	AO/$\overline{CS}$ Set-up to $\overline{WR}$↓		50		0		0		ns
TWHAX	AO/$\overline{CS}$ Hold after $\overline{WR}$↑		20		0		0		ns
TWLWH	$\overline{WR}$ Pulse Width		400		290		190		ns
TDVWH	Data Set-up to $\overline{WR}$↑		300		240		160		ns
TWHDX	Data Hold after $\overline{WR}$↑		40		0		0		ns
TJLJH	Interrupt Request Width (LOW)	See Note 1	100		100		100		ns
TCVIAL	Cascade Set-up Second or Third $\overline{INTA}$↓ (Slave Only)		55		55		40		ns
TRHRL	End of $\overline{RD}$ to next $\overline{RD}$ End of $\overline{INTA}$ to next $\overline{INTA}$ within an $\overline{INTA}$ sequence only		160		160		160		ns
TWHWL	End of $\overline{WR}$ to next $\overline{WR}$		190		190		190		ns
*TCHCL	End of Command to next Command (Not same command type)		500		500		500		ns
	End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence								

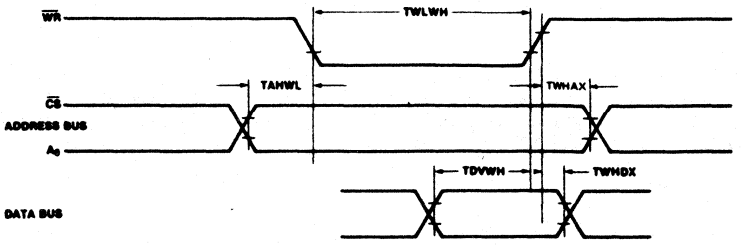
*Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 500 ns (i.e. 8085AH = 1.6μs, 8085AH-2 = 1μs, 8086 = 1μs, 8086-2 = 625 ns)
Note 1: This is the low time required to clear the input latch in the edge triggered mode.

TIMING RESPONSES

Parameters	Description	Test Conditions	8259A-8		8259A		8259A-2		Units
			Min	Max	Min	Max	Min	Max	
TRLDV	Data Valid from $\overline{RD}$/$\overline{INTA}$↓	C of Data Bus = 100pF		300		200		120	ns
TRHDZ	Data Float after $\overline{RD}$/$\overline{INTA}$↑	C of Data Bus	10	200	10	100	10	85	ns
TJHIH	Interrupt Output Delay	Max text C = 100pF Min test C = 15pF		400		350		300	ns
TIALCV	Cascade Valid from First $\overline{INTA}$↓ (Master Only)	C_{INT} = 100pF		565		565		360	ns
TRLEL	Enable Active from $\overline{RD}$↓ or $\overline{INTA}$↓	$C_{CASCADE}$ = 100pF		160		125		100	ns
TRHEH	Enable Inactive from $\overline{RD}$↑ or $\overline{INTA}$↑			325		150		150	ns
TAHDV	Data Valid from Stable Address			350		200		200	ns
TCVDV	Cascade Valid to Valid Data			300		300		200	ns

SWITCHING WAVEFORMS

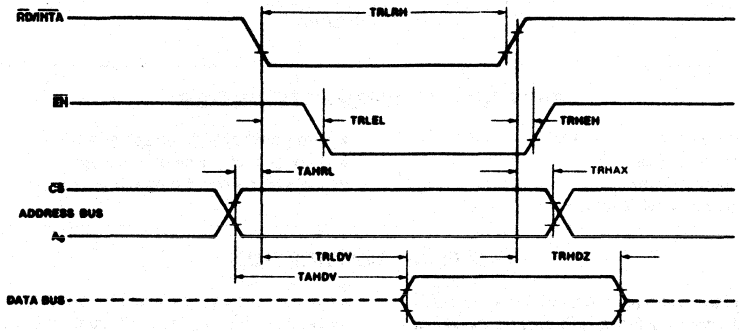
WRITE



WF006070

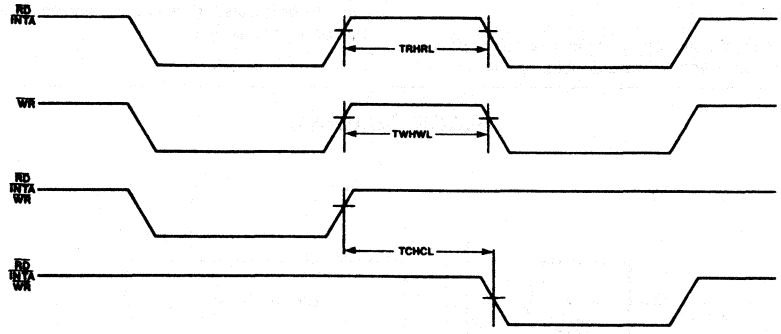
SWITCHING WAVEFORMS (Continued)

READ/INTA



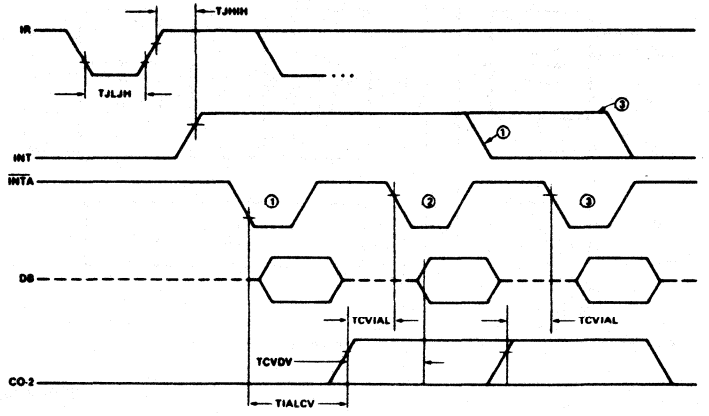
WF006080

OTHER TIMING



WF006090

INTA SEQUENCE



WF006100

Notes: Interrupt output must remain HIGH at least until leading edge of first INTA.
1. Cycle 1 in iAPX86, iAPX88 systems, the Data Bus is not active.

82C59A

CMOS Programmable Interrupt Controller
iAPX86 Family

DISTINCTIVE CHARACTERISTICS

- Pin Compatible with NMOS 8259A
- Expandable to 64 Levels
- Eight Level Priority Controller
- Individual Request Mask Capability
- Programmable Interrupt Modes
- Low Standby Power – 10 μA
- iAPX86 Family Compatible

GENERAL DESCRIPTION

The 82C59A is a high performance CMOS Priority Interrupt Controller manufactured using a self-aligned silicon gate CMOS process. The 82C59A is designed to relieve the system CPU from the task of polling in a multi-level priority interrupt system. The high speed and industry standard configuration of the 82C59A make it compatible with microprocessors, such as the 80286, 80186, 8086, 8088, 8080, and 8085.

The 82C59A can handle up to eight vectored priority interrupting sources and is cascadable to 64 without additional circuitry. Individual interrupting sources can be masked or prioritized to allow custom system configuration. Two modes of operation make the 82C59A compatible with 80286, 80186, 8086, 8088, 8080, and 8085 formats.

Static CMOS circuit design insures low operating power. AMD's advanced CMOS process results in performance equal to or greater than existing equivalent products at a fraction of the power.

BLOCK DIAGRAM

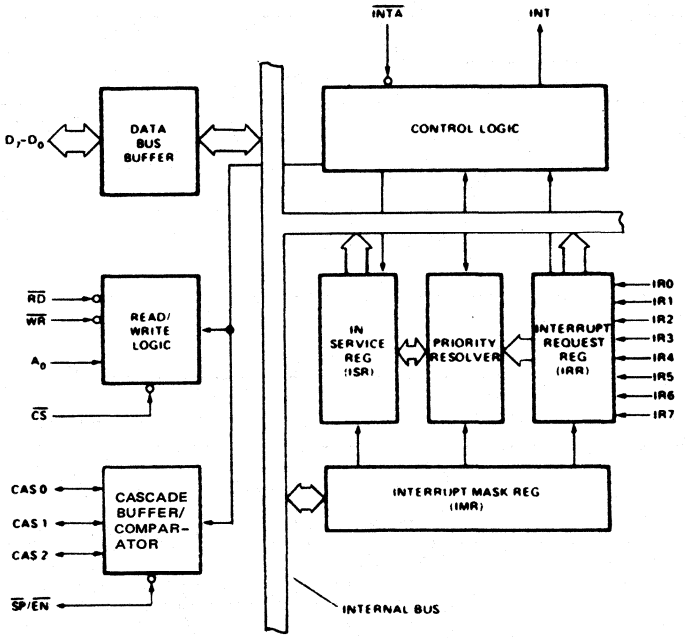


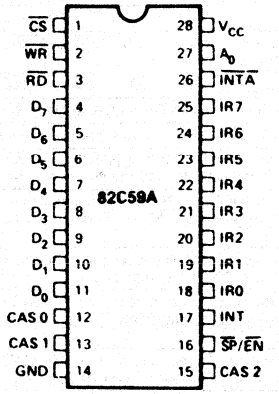
Figure 1.

06102A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View

D-28, P-28



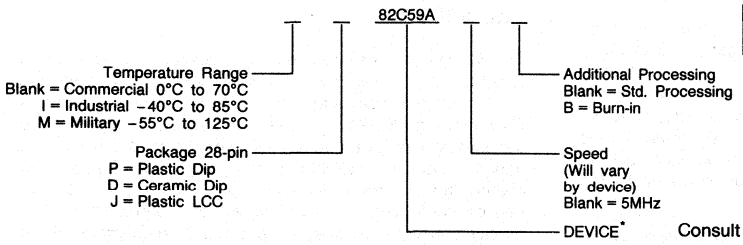
CD005641

Note: Pin 1 is marked for orientation

Figure 2.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
82C59A	P, D, ID

*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
28	V_{CC}	I	Supply: +5V Supply.
14	GND	I	Ground.
1	$\overline{CS}$	I	Chip Select: A LOW on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 82C59A. INTA functions are independent of $\overline{CS}$.
2	$\overline{WR}$	O	Write: A LOW on this pin when $\overline{CS}$ is LOW enables the 82C59A to accept command words from the CPU.
3	$\overline{RD}$	I	Read: A LOW on this pin when $\overline{CS}$ is LOW enables the 82C59A to release status onto the data bus for the CPU.
4-11	D_7-D_0	I/O	Bidirectional Data Bus: Control, status and interrupt-vector information are transferred via this bus.
12, 13, 15	CAS_0-CAS_2	I/O	Cascade Lines: The CAS lines form a private 82C59A bus to control a multiple 82C59A structure. These pins are outputs for a master 82C59A and inputs for a slave 82C59A.
16	$\overline{SP}/\overline{EN}$	I/O	Slave Program/Enable Buffer: This is a dual function pin. When in the Buffered Mode, it can be used as an output to control buffer transceivers ($\overline{EN}$). When not in the buffered mode, it is used as an input to designate a master ($\overline{SP}$ = 1) or slave ($\overline{SP}$ = 0).
17	INT	O	Interrupt: This pin goes HIGH whenever a valid interrupt request is asserted. It is used to interrupt the CPU; thus, it is connected to the CPU's interrupt pin.
18-25	IR_0-IR_7	I	Interrupt Requests: Asynchronous inputs. An interrupt request is executed by raising an IR input (LOW-to-HIGH), and holding it HIGH until it is acknowledged (Edge Triggered Mode) or just by a high level on an IR input (Level Triggered Mode).
26	$\overline{INTA}$	I	Interrupt Acknowledge: This pin is used to enable 82C59A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
27	A_0	I	A_0 Address Line: This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 82C59A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A_0 address line (A_1 for 8086/88 CPU's).

DETAILED DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I/O devices, such as keyboards, displays, sensors and other components, receive servicing in an efficient manner, so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt.* It is easy to see that system throughout would drastically increase, and thus, more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine." The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to in this document as vectoring data.

The 82C59A

The 82C59A is a device specifically designed for use in real time, interrupt-driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 82C59A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 82C59A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined, as required, based on the total system environment.

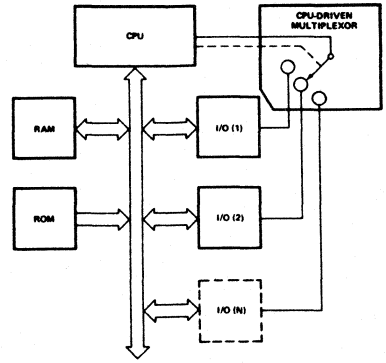
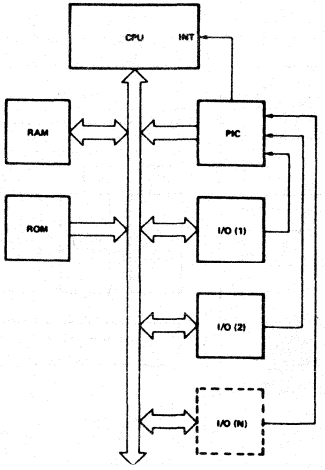


Figure 3a. Polled Method



AF003330

Figure 3b. Interrupt Method

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 82C59A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 82C59A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 82C59A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 82C59A to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

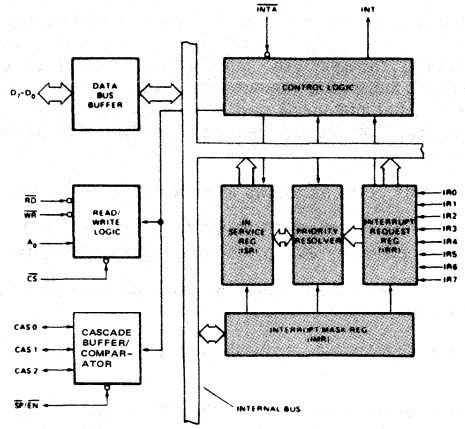
A LOW on this input enables the 82C59A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 82C59A.

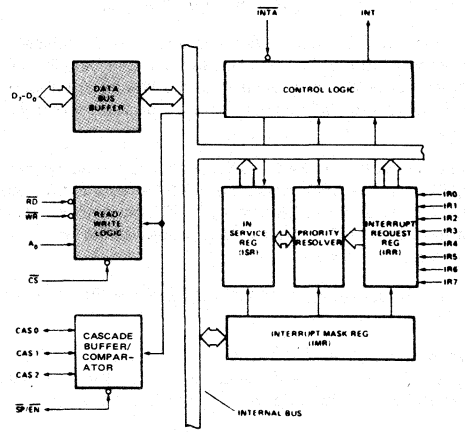
$\overline{RD}$ (READ)

A LOW on this input enables the 82C59A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the interrupt level onto the Data Bus.



BD003540

Figure 4a. 82C59A Block Diagram



BD003540

Figure 4b. 82C59A Block Diagram

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers as well as read the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 82C59A's used in the system. The associated three I/O pins (CAS0 – 2) are outputs when the 82C59A is used as a master and are inputs when the 82C59A is used as a slave. As a master, the 82C59A sends the ID of the interrupting slave device onto the CAS0 – 2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 82C59A".)

Interrupt Sequence

The powerful features of the 82C59A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an 8080A/85AH system:

1. One or more of the INTERRUPT REQUEST lines (IR7 – 0) are raised HIGH, setting the corresponding IRR bit(s).
2. The 82C59A evaluates these requests and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 82C59A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7 – 0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 82C59A from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 82C59A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse, and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 82C59A. In the AEOI mode, the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occurring in an 8086 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 82C59A does not drive the Data Bus during this cycle.
5. The 8086 will initiate a second $\overline{INTA}$ pulse. During this pulse, the 82C59A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode, the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration), the 82C59A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

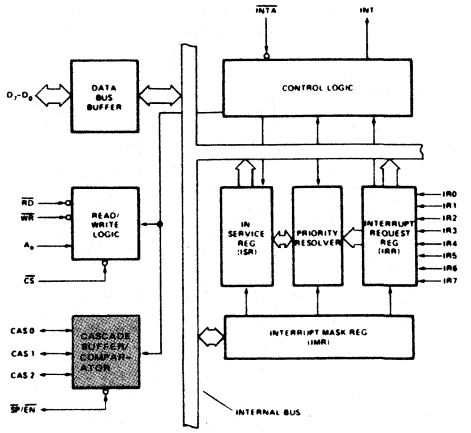


Figure 4c. 82C59A Block Diagram

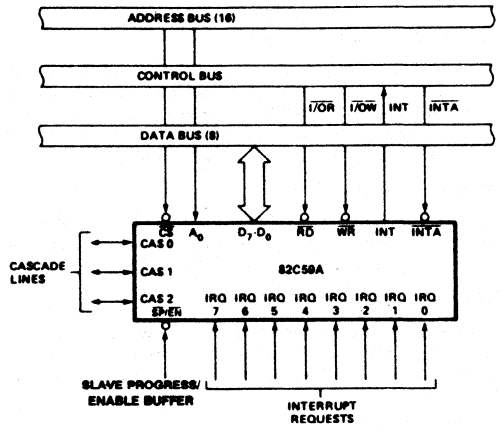


Figure 5. 82C59A Interface to Standard System Bus

Interrupt Sequence Outputs

8080A, 8085AH

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse, the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse, the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits, $A_5 - A_7$ are programmed while $A_0 - A_4$ are automatically inserted by the 82C59A. When Interval = 8, only A_6 and A_7 are programmed while $A_0 - A_5$ are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third INTA pulse, the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence ($A_8 - A_{15}$), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

8086, 8088

8086 mode is similar to 8080A mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of 8080A/85AH systems in that the 82C59A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the INTA pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in 8086 mode, the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and $A_5 - A_{11}$ are unused in 8086 mode):

Content of Interrupt Vector Byte for 8086 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING INFORMATION

Programming the 82C59A

The 82C59A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 82C59A in the system must be brought to a starting point – by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses.
2. *Operation Command Words (OCWs):* These are the command words which command the 82C59A to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 82C59A anytime after initialization.

Initialization Command Words (ICWS)

General

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur:

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a LOW-to-HIGH transition to generate an interrupt.

b. The Interrupt Mask Register is cleared.

c. IR7 input is assigned priority 7.

d. The slave mode address is set to 7.

e. Special Mask Mode is cleared and Status Read is set to IRR.

f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, 8080A/85AH system).

*Note: Master/Slave in ICW4 is only used in the buffered mode.

Initialization Command Words 1 and 2 (ICW1, ICW2)

$A_5 - A_{15}$: *Page starting address of service routines.* In an 8080A/85AH system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations; thus, the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long ($A_0 - A_{15}$). When the routine interval is 4, $A_0 - A_4$ are automatically inserted by the 82C59A, while $A_5 - A_{15}$ are programmed externally. When the routine interval is 8, $A_0 - A_5$ are automatically inserted by the 82C59A, while $A_6 - A_{15}$ are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an 8086 system, $A_{15} - A_{11}$ are inserted in the five most significant bits of the vectoring byte, and the 82C59A sets the three least significant bits according to the interrupt level. $A_{10} - A_5$ are ignored and ADI (address interval) has no effect.

LTIM: If LTIM = 1, then the 82C59A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. If ADI = 1, then interval = 4; if ADI = 0, then interval = 8.

SNGL: Single. Means that this is the only 82C59A in the system. If SNGL = 1, no ICW3 will be issued.

IC4: If this bit is set – ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

Initialization Command Word 3 (ICW3)

This word is read only when there is more than one 82C59A in the system and cascading is used; in which case, SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1 or in buffered mode when M/S = 1 in ICW4), a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for 8080A/85AH system) and will enable the corresponding slave to release bytes 2 and 3 (for 8086 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4), bits 2 – 0 identify the slave. The slave compares its cascade input with these bits, and if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 8086) are released by it on the Data Bus.

Initialization Command Word 4 (ICW4)

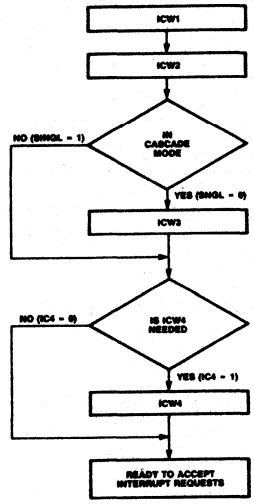
SFNM: If SFNM = 1, the special fully nested mode is programmed.

BUF: If BUF = 1, the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output, and the master/slave determination is by M/S.

M/S: If buffered mode is selected, M/S = 1 means the 82C59A is programmed to be a master; M/S = 0 means the 82C59A is programmed to be a slave. If BUF = 0, M/S has no function.

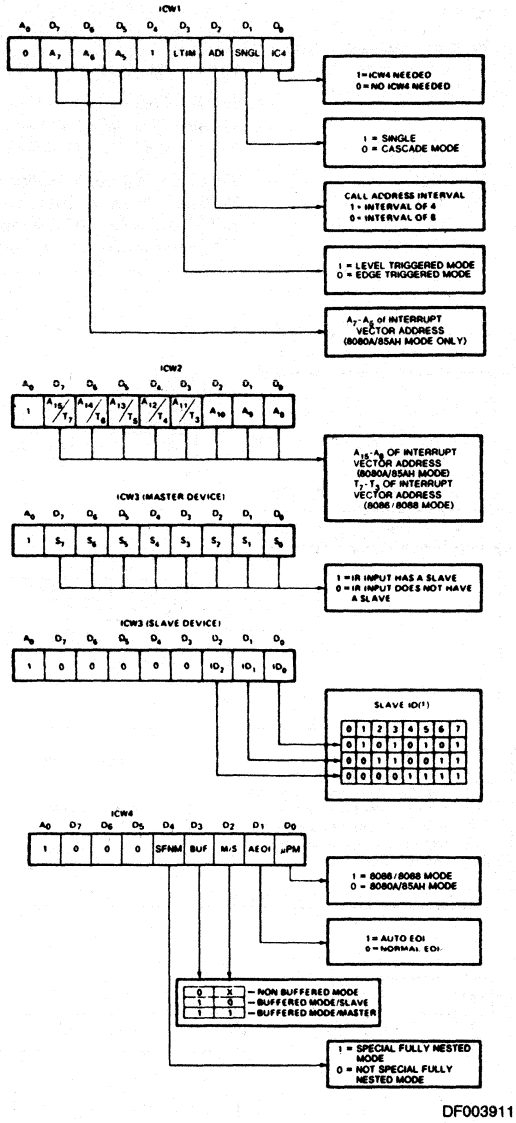
AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 82C59A for 8080A, 85 system operation; μPM = 1 sets the 82C59A for 8086 system operation.



PF001310

Figure 6. Initialization Sequence



DF003911

Note 1. Slave ID is equal to the corresponding master IR input.

Figure 7. Initialization Command Word Format

Operation Command Words (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 82C59A, the chip is ready to accept interrupt requests at its input lines. However, during the 82C59A operation, a selection of algorithms can command the 82C59A to operate in various modes through the Operation Command Words (OCWs).

Operation Control Words (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SL	EOI	0	0	L2	L1	L0

OCW3

0	0	ESMM	SMM	0	1	P	RR	RIS

Operation Control Word 1 (OCW1)

OCW1 sets and clears the mask bits in the Interrupt Mask Register (IMR). $M_7 - M_0$ represent the eight mask bits. M = 1 indicates the channel is masked (inhibited); M = 0 indicates the channel is enabled.

Operation Control Word 2 (OCW2)

R, SL, EOI – These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0 – These bits determine the interrupt level acted upon when the SL bit is active.

Operation Control Word 3 (OCW3)

ESMM – Enable Special Mask Mode. When this bit is set to 1, it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a "don't care."

SMM – Special Mask Mode. If ESMM = 1 and SMM = 1, the 82C59A will enter Special Mask Mode. If ESMM = 1 and SMM = 0, the 82C59A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

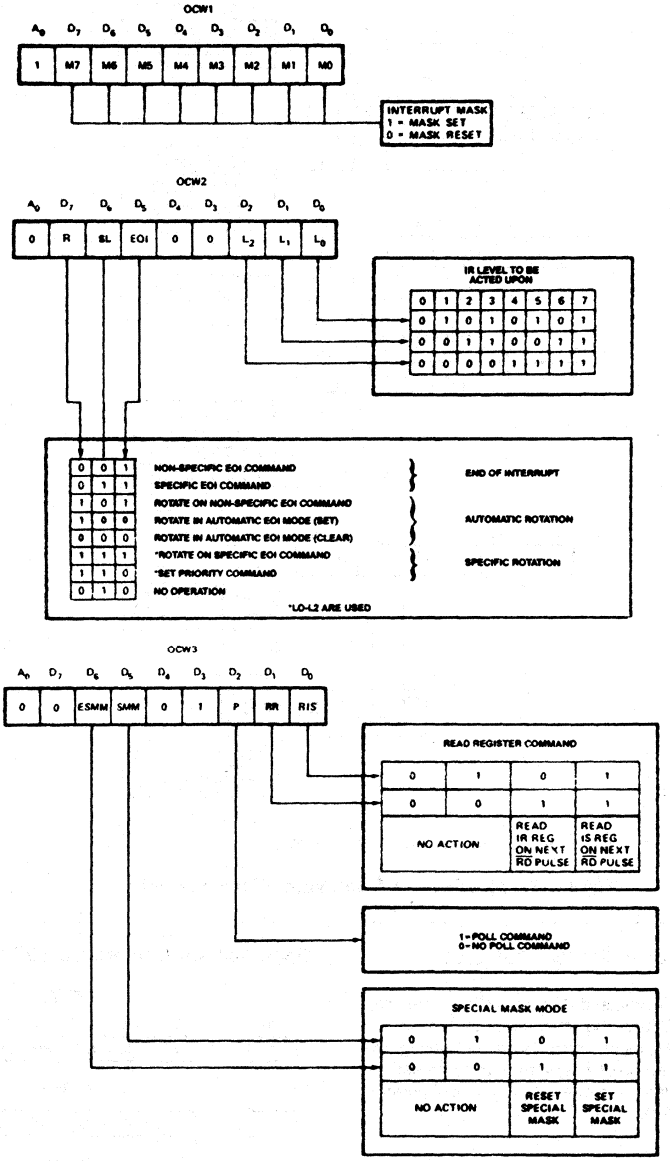


Figure 8. Operation Command Word Format

DF003900

Fully Nested Mode

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt Enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

End of Interrupt (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{INTA}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 82C59A before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 82C59A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued, the 82C59A will automatically reset the highest IS bit of those that are set, since in the fully nested mode, the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 82C59A may no longer be able to determine the last level acknowledged. In this case, a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and LO-L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 82C59A is in the Special Mask Mode.

Automatic End of Interrupt (AEOI) Mode

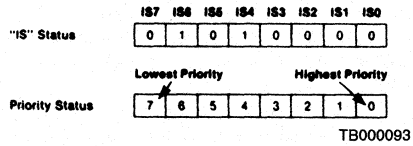
If AEOI = 1 in ICW4, then the 82C59A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 82C59A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-8080A/85AH, second in 8086). Note that from a system standpoint this mode should be used only when a nested multilevel interrupt structure is not required within a single 82C59A.

The AEOI mode can only be used in a master 82C59A and not a slave.

Automatic Rotation (Equal Priority Devices)

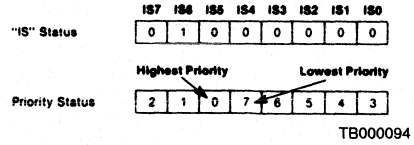
In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case, until each of 7 other devices are serviced *once at most.* For example, if the priority and "in service" statuses are:

Before Rotate (IR4 the highest priority requiring service)



TB000093

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



TB000094

There are two ways to accomplish Automatic Rotation using OCW2: the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode, which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

Specific Rotation (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1; LO-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO-L2 = IR level to receive bottom priority).

Interrupt Masks

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels' operation.

Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that, if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 82C59A would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

Poll Command

In this mode the INT output is not used or the microprocessor internal Interrupt Enable flip-flop reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

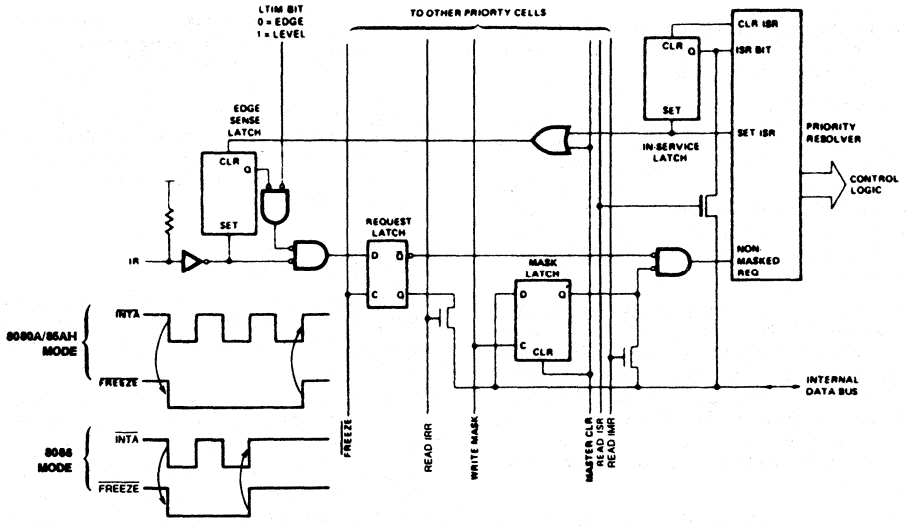
The Poll command is issued by setting P = "1" in OCW3. The 82C59A treats the next $\overline{RD}$ pulse to the 82C59A (i.e., $\overline{RD}$ = 0, $\overline{CS}$ = 0) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	–	–	–	–	W2	W1	W0

W0 – W2: Binary code of the highest priority level requesting service.
I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.



LS001361

Notes: 1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/IS ACTIVE DURING $\overline{INTA}$/AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	Di	Di	FOLLOW
0	X	Qn – 1	HOLD

Figure 9. Priority Cell — Simplified Logic Diagram

Reading the 82C59A Status

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3(RR = 1,RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 82C59A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 82C59A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and AO = 1 (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

Edge and Level Triggered Modes

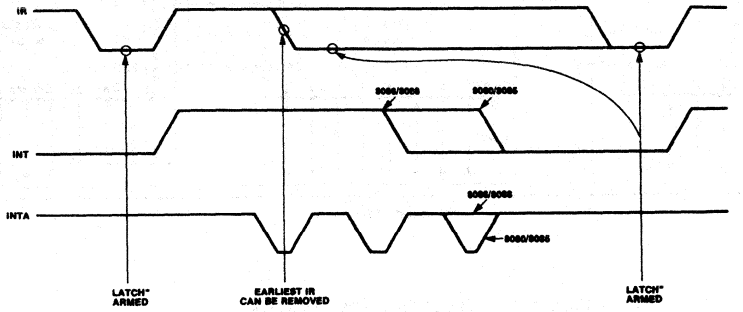
This mode is programmed using bit 3 in ICW1.

If LTIM = "0," an interrupt request will be recognized by a LOW-to-HIGH transition on an IR input. The IR input can remain HIGH without generating another interrupt.

If LTIM = "1," an interrupt request will be recognized by a "HIGH" level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 82C59A. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes, the IR inputs must remain HIGH until after the falling edge of the first INTA. If the IR input goes LOW before this time, a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature, the IR7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes, a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit; a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs, it is a default.



*EDGE TRIGGERED MODE ONLY

Figure 10. IR Triggering Timing Requirements

The Special Fully Nested Mode

This mode will be used in this case of a big system where cascading is used and the priority has to be conserved within each slave. In the case, the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic, and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode, a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine, the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master, too. If not, no EOI should be sent.

Buffered Mode

When the 82C59A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, the problem of enabling buffers exists.

The buffered mode will structure the 82C59A to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 82C59A's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 82C59A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

Cascade Mode

The 82C59A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{INTA}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of INTA. (Byte 2 only for 8086/8088).

The cascade bus lines are normally LOW and will contain the slave address code from the trailing edge of the first INTA pulse to the trailing edge of the third pulse. Each 82C59A in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each 82C59A.

The cascade lines of the Master 82C59A are activated only for slave inputs; non-slave inputs leave the cascade line inactive (LOW).

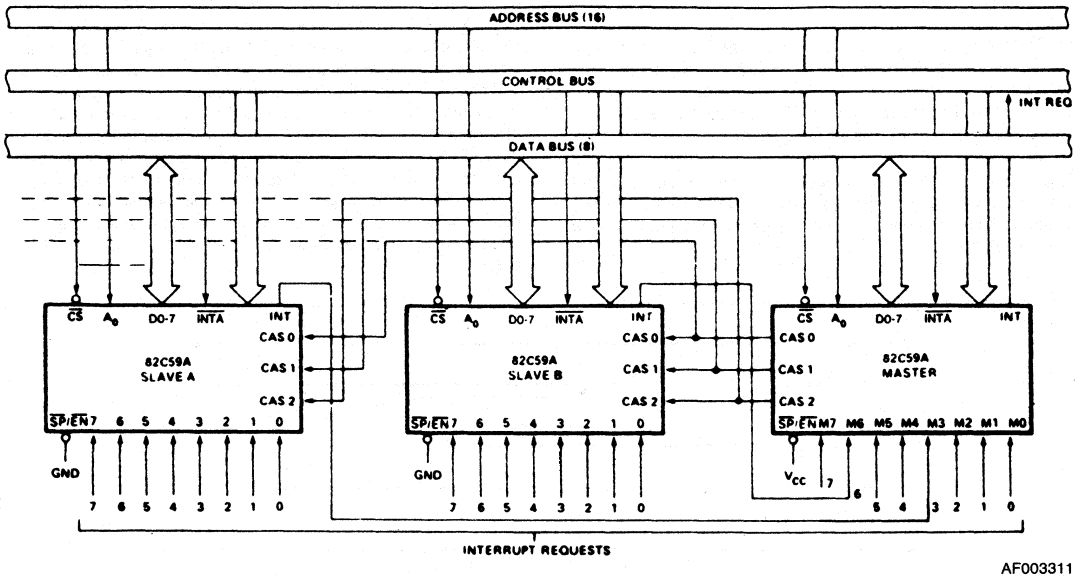


Figure 11. Cascading the 82C59A

ABSOLUTE MAXIMUM RATINGS

Storage Temperature .....–65 to 150°C
Voltage on Any Pin
with Respect to Ground .....–0.5 to +7.0V
Power Dissipation 1.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}
82C59A	0°C to 70°C	5V ± 10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
VIH	Logical One Input Voltage		2.0	V_{CC} + 0.5	V
VIL	Logical Zero Input Voltage		–0.5	0.8	V
VOH	Output High Voltage	IOH = –400 μA	2.4		V
VOL	Output Low Voltage	IOL = +2.2 mA		0.45	V
ILI	Input Linkage Current	OV ≤ VIN ≤ VCC	–10.0	+10.0	μA
ILOL	Output Leakage Current	OV ≤ VO ≤ VCC	–10.0	+10.0	μA
ICCSB	Standby Power Supply Current	VCC = 5.5V VIN = VCC or GND* Outputs Open		10	μA

*All interrupt requested pins are equal to VCC.

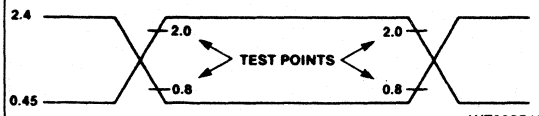
CAPACITANCE

T_A = 25°C; VCC = GND = 0V; VIN = +5V or GND

Parameters	Description	Test Conditions	Min	Max	Units
C_{IN}*	Input Capacitance	FREQ = 1 MHz Unmeasured pins returned to GND		5	pf
C_{OUT}*	Output Capacitance			15	pf
$C_{I/O}$*	I/O Capacitance			20	pf

*Guaranteed and sampled, but not 100% tested.

SWITCHING TEST INPUT/OUTPUT WAVEFORM



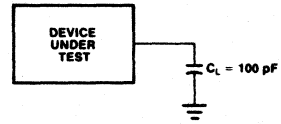
WF009540

AC TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

TEST CONDITION	V1	R1	R2	C1
1	1.7V	523Ω	OPEN	100 pf
2	4.5V	1.8KΩ	1.8KΩ	30 pf

TEST CONDITION DEFINITION TABLE

SWITCHING TEST LOAD CIRCUIT



TC001840

C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
TAHRL	A0/$\overline{CS}$ Set-up to $\overline{RD}$/$\overline{INTA}$		10		ns
TRHAX	A0/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$		5		ns
TRLRH	$\overline{RD}$ Pulse Width		160		ns
TAHWL	A0/$\overline{CS}$ Set-up to $\overline{WR}$		0		ns
TWHAX	A0/$\overline{CS}$ Hold after $\overline{WR}$		0		ns
TWLWH	$\overline{WR}$ Pulse Width		190		ns
TDVWH	Data Set-up to $\overline{WR}$		160		ns
TWHDX	Data Hold after $\overline{WR}$		0		ns
TJLJH	Interrupt Request Width (LOW)	See Note 1	100		ns
TCVIAL	Cascade Set-up to second or third $\overline{INTA}$ (Slave Only)		40		ns
TRHRL	End of $\overline{RD}$ to next $\overline{RD}$: End of $\overline{INTA}$ to next $\overline{INTA}$ within an $\overline{INTA}$ sequence only		160		ns
TWHWL	End of $\overline{WR}$ to next $\overline{WR}$		190		ns
*TCHCL	End of Command to next Command (Not same command type) End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence		400		ns

**Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 400 ns (i.e. 8085A = 1.6μs, 8085A-2 = 1μs, 80C86 = 1μs).*

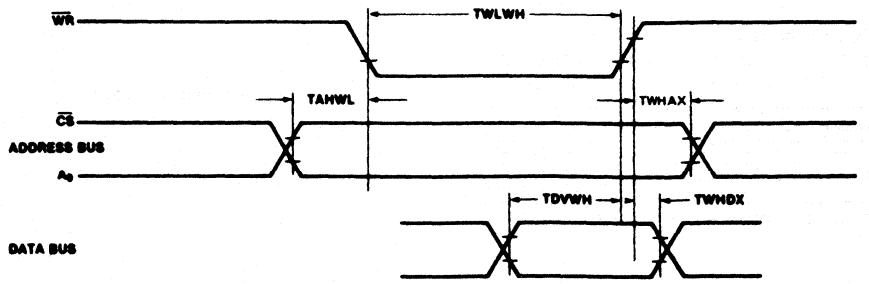
Note 1: This is the low time required to clear the input latch in the edge triggered mode.

Timing Responses

Parameters	Description	Test Conditions	Min	Max	Units
TRLDV	Data Valid from $\overline{RD}$/$\overline{INTA}$	C_L = 100pF on max tests		120	ns
TRHDZ	Data Float after $\overline{RD}$/$\overline{INTA}$			10	85
TJHIH	Interrupt Output Delay			300	ns
TIALCV	Cascade Valid from First $\overline{INTA}$ (Master Only)	C_L = 15pF on min tests		360	ns
TRLEL	Enable Active from $\overline{RD}$ or $\overline{INTA}$			100	ns
TRHEH	Enable Active from $\overline{RD}$ or $\overline{INTA}$			150	ns
TAHDV	Data Valid from Stable Address			200	ns
TCVDV	Cascade Valid to Valid Data			200	ns

SWITCHING WAVEFORMS

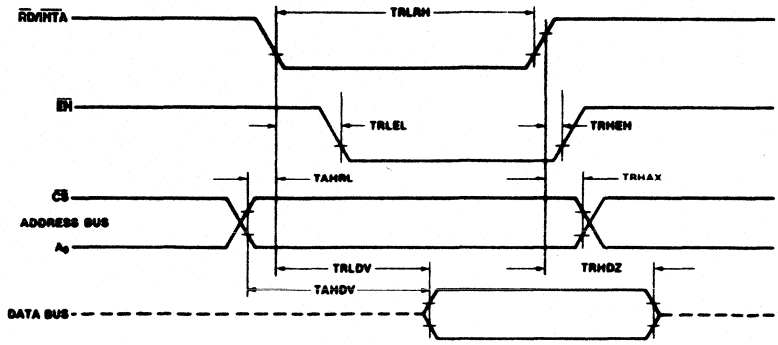
WRITE



WF006070

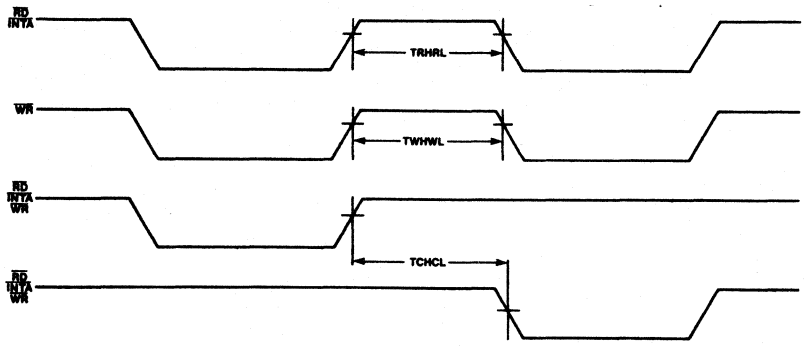
SWITCHING WAVEFORMS (Cont.)

READ/INTA



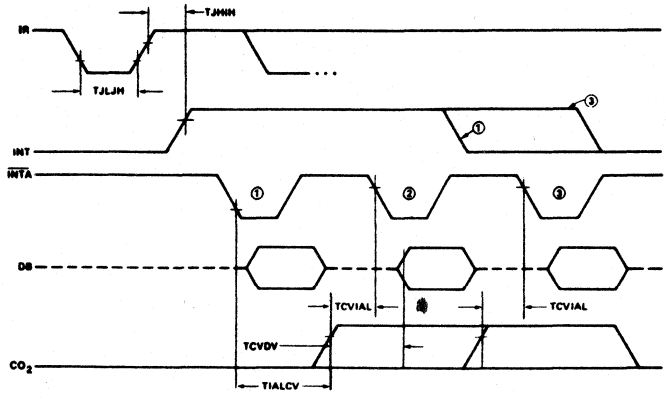
WF006080

OTHER TIMING



WF006090

INTA SEQUENCE



WF006101

Notes: 1. Interrupt output must remain HIGH at least until leading edge of first $\overline{\text{INTA}}$.
2. Cycle 1 in 8086/88 systems, the Data Bus is not active.

Single-Chip Microcomputers Index

The 8051 Family

PART	TECHNOLOGY	ON-CHIP PROGRAM MEMORY	ON-CHIP DATA MEMORY
8051AH	NMOS	4K — ROM	128
8031AH	NMOS	NONE	128
8751H	NMOS	4K — EPROM	128
Am9761H	NMOS	8K — EPROM	128
80C51	CMOS	4K — ROM	128
80C31	CMOS	NONE	128

Figure 1-1. 8051 Family Members

THE MAJOR FEATURES OF THE 8051 FAMILY ARE:

- 8-Bit CPU
- On-Chip oscillator and clock circuitry
- 32 I/O lines
- 64K address space for external data memory
- 64K address space for external program memory
- Two 16-bit timer/counters
- A five-source interrupt structure with two priority levels
- Full duplex serial port
- Boolean processor

INTRODUCTION

The 8051AH is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time applications such as instrumentation, industrial control, and intelligent computer peripherals. It provides the hardware features, architectural enhancements, and new instructions that make it a powerful and cost effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage. A Block Diagram is shown in Figure 1.

The 8031AH is a control-oriented CPU without on-chip program memory. It can address 64K-bytes of External Program Memory in addition to 64K bytes of External Data Memory. For systems requiring extra capability, each member of the 8051AH Family can be expanded using standard memories and the byte oriented 8080 and 8085 peripherals. The 8051AH is an 8031AH with the lower 4K-bytes of Program Memory filled with on-chip mask programmable ROM.

The 8051AH is suited for low-cost, high volume production; and the 8031AH for applications desiring the flexibility of External Program Memory which can be easily modified and updated in the field.

On a single die the 8051AH microcomputer combines CPU; 4K x 8 read-only program memory; 128 x 8 RAM; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; serial I/O port for either multiprocessor communications, I/O expansion or full duplex UART; and on-chip oscillator and clock circuits. This section will provide an overview of the 8051AH by providing a high-level description of its major elements: the CPU architecture and the on-chip functions peripheral to the CPU. The generic term "8051AH" is used to refer collectively to the 8031AH and 8051AH.

Refer to page 7-1 for Essential Information on Military Devices

1.1 MEMORY ORGANIZATION

The 8051AH CPU manipulates operands in three memory spaces. These are the 64K-byte Program Memory, 64K-byte External Data Memory and 256-byte Internal Data Memory. The Internal Data Memory address space is further divided into the 128-byte Internal Data RAM and 128-byte Special Function Register (SFR) address spaces shown in Figure 4. Four Register Banks (each with eight registers), 128 addressable bits and the stack reside in the Internal Data RAM. The stack depth is limited only by the available Internal Data RAM and its location is determined by the 8-bit stack pointer. All registers except the four 8-Register Banks reside in the Special Function Register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers, and a serial port. 92 bit locations in the SFR address space are addressable as bits. The 8051AH contains 128 bytes of Internal Data RAM and 20 SFRs.

Special Function Registers

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer (consisting of DPH and DPL	83H 82H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
TCON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 (high byte)	8CH
TL0	Timer/Counter 0 (low byte)	8AH
TH1	Timer/Counter 1 (high byte)	8DH
TL1	Timer/Counter 1 (low byte)	8BH
*SCON	Serial Control	98H
SBUF	Serial Data Buff	99H
PCON	Power Control	87H

The SFRs marked with an asterisk (*) are both bit- and byte-addressable. The functions of the SFRs are described as follows.

ACCUMULATOR

ACC is the Accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

B REGISTER

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

PROGRAM STATUS WORD

The PSW register contains program status information as detailed in Figure 1-2.

STACK POINTER

The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

DATA POINTER

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

PORTS 0 to 3

P0, P1, P2 and P3 are the SFR latches of Ports 0, 1, 2 and 3, respectively.

SERIAL DATA BUFFER

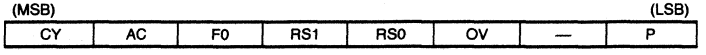
The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

TIMER REGISTERS

Register pairs (TH0, TL0) and (TH1, TL1) are the 16-bit counting registers for Timer/Counters 0 and 1 respectively.

CONTROL REGISTERS

Special Function Registers IP, IE, TMOD, TCON, SCON, and PCON contain control and status bits for the interrupt system, the timer/counters, and the serial port. They are described in later sections.



Symbol	Position	Name and Significance
CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary Carry flag. (For BCD operations).
F0	PSW.5	Flag 0 (Available to the user for general purposes).
RS1	PSW.4	Register bank Select control bits 1 & 0. Set/cleared by software to determine.
RS0	PSW.3	working register bank (see Note).
OV	PSW.2	Overflow flag.
—	PSW.1	(reserved)
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e., even parity.

Note — the contents of (RS1, RS0) enable the working register banks as follows:

(0.0) — Bank 0	(00H-07H)
(0.1) — Bank 1	(08H-0FH)
(1.0) — Bank 2	(10H-17H)
(1.1) — Bank 3	(18H-1FH)

Figure 1-2. PSW: Program Status Word Register

Refer to page 7-1 for Essential Information on Military Devices

1.2 OSCILLATOR AND CLOCK CIRCUIT

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter, which can be configured with off-chip components as a Pierce oscillator, as shown in Figure 1.3. The on-chip circuitry, and selection of off-chip components to configure the oscillator are discussed below.

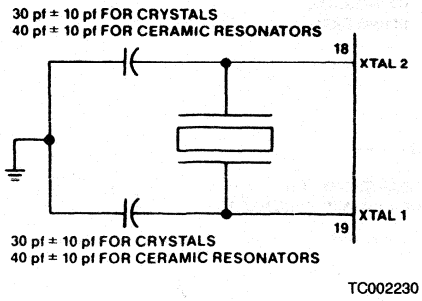


Figure 1-3. Crystal/Ceramic Resonator Oscillator

The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clocking signals to the chip. The internal clocking signals are at half the oscillator frequency, and define the internal phases, states, and machine cycles, which are described below.

The on-chip oscillator circuitry for the NMOS members of the 8051 family is a single stage linear inverter (figure 1-3A), intended for use as a crystal-controlled, positive reactance oscillator (Figure 1-3B). In this application the crystal is operated in its fundamental response mode as an inductive reactance in parallel resonance with capacitance external to the crystal.

The crystal specifications and capacitance values (C1 and C2 in Figure 1-3B) are not critical. 30pF can be used in these positions at any frequency with good quality crystals. A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values, typically, 47pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

To drive the NMOS parts with an external clock source, apply the external clock signal to XTAL2, and ground XTAL1, as shown in Figure 1-3C. A pull-up resistor is suggested because the logic levels at XTAL2 are not TTL.

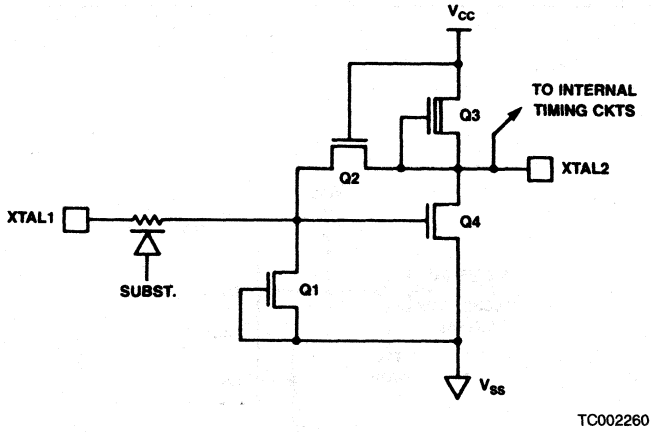


Figure 1-3A. On-Chip Oscillator Circuitry in the NMOS Versions of the 8051 Family

Refer to page 7-1 for Essential Information on Military Devices

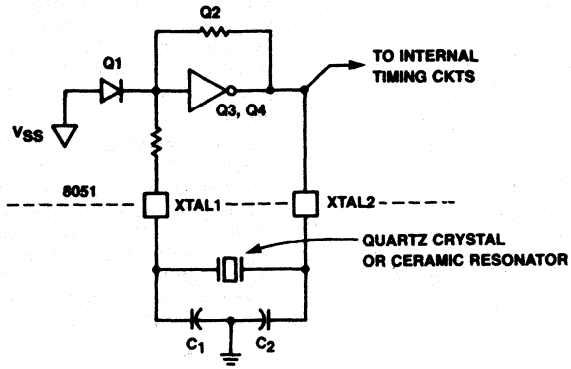


Figure 1-3B. Using the NMOS On-Chip Oscillator

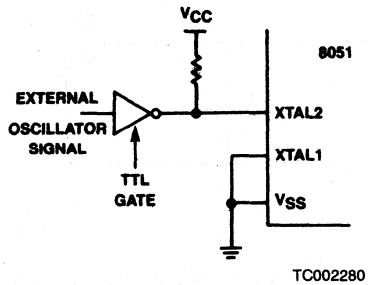


Figure 1-3C. Driving the NMOS 8051 Parts 1-3C with an External Clock Source

Refer to page 7-1 for Essential Information on Military Devices

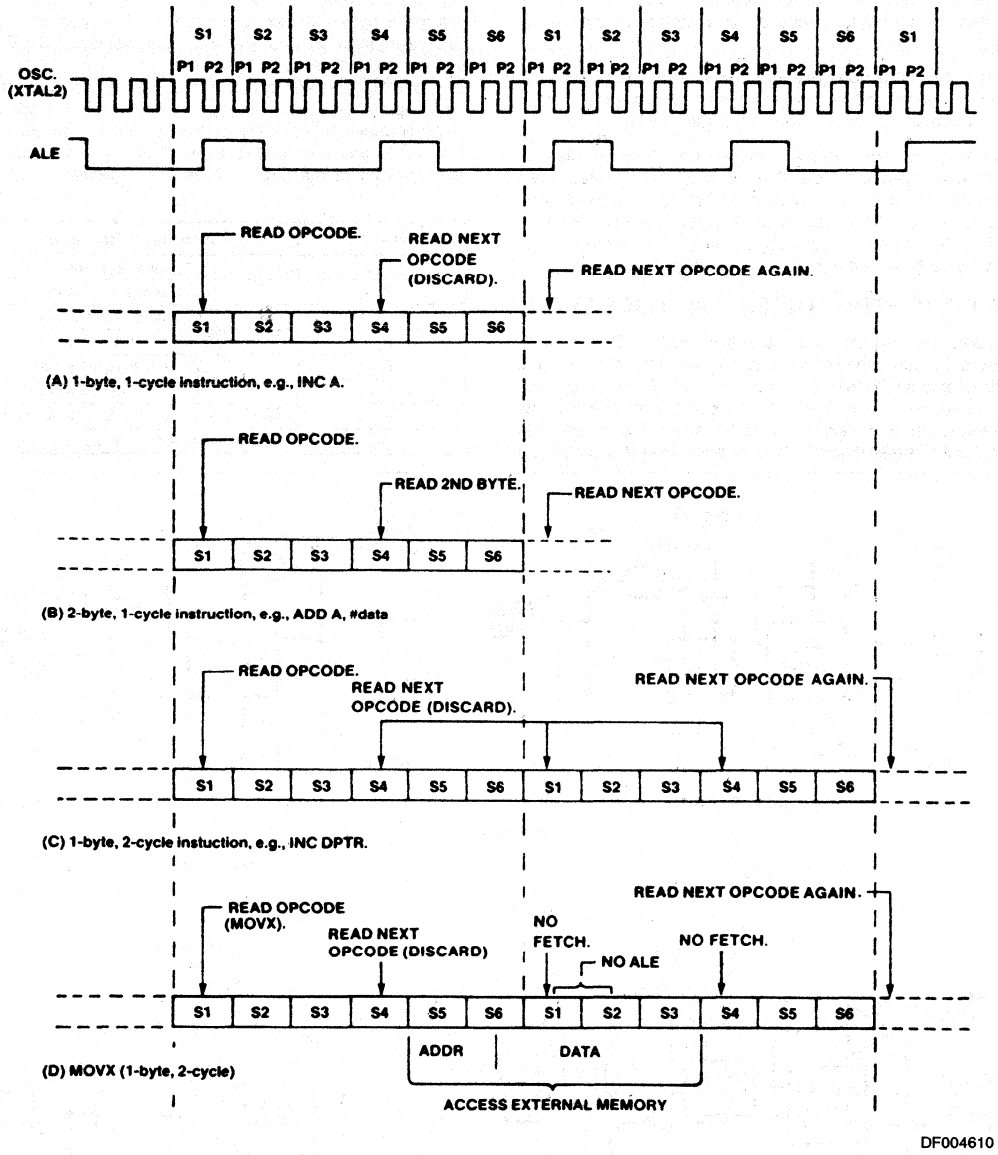


Figure 1-4. 8051 Fetch/Execute Sequences

1.3 CPU TIMING

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a Phase 1 half, during which the Phase 1 clock is active, and a Phase 2 half, during which the Phase 2 clock is active. Thus, a machine cycle consists of 12 oscillator periods, numbered S1P1 (State 1, Phase 1), through S6P2 (State 6, Phase 2). Each phase lasts for one oscillator period. Each state lasts for two oscillator periods. Typically, arithmetic and logical operations take place during Phase 1 and internal register-to-register transfers take place during Phase 2.

The diagrams in Figure 1-4 show the fetch/execute timing referenced to the internal states and phases. Since these internal clock signals are not user accessible, the XTAL2 oscillator signal and the ALE (Address Latch Enable) signal are shown for external reference. ALE is normally activated twice during each machine cycle: once during S1P2 and S2P1, and again during S4P2 and S5P1.

Execution of a one-cycle instruction begins at S1P2, when the opcode is latched into the Instruction Register. If it is a two-byte instruction, the second byte is read during S4 of the same machine cycle. If it is a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next

Refer to page 7-1 for Essential Information on Military Devices

opcode), is ignored, and the Program Counter is not incremented. In any case, execution is complete at the end of S6P2.

Most 8051 instructions execute in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than two cycles to complete. They take four cycles.

Normally, two code bytes are fetched from Program Memory during every machine cycle. The only exception to this is when a MOVX instruction is executed. MOVX is a 1-byte 2-cycle instruction that accesses external Data Memory. During a MOVX, two fetches are skipped while the external Data Memory is being addressed and strobed.

1.4 PORT STRUCTURES AND OPERATION

The 8051AH has instructions that treat its 32 I/O lines as 32 individually addressable bits and as four parallel 8-bit ports addressable as Ports 0, 1, 2 and 3. Ports 0, 2, and 3 can also assume other functions. Port 0 provides the multiplexed low-order address and data bus used for expanding the 8051AH with standard memories and peripherals. Port 2 provides the high-order address bus when expanding the 8051AH with external Program Memory or more than 256 bytes of External Data Memory. Port 3 is an 8-bit quasi-bidirectional I/O port with internal pullups. It also contains the interrupt, timer, serial port, and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LSTTL loads. The secondary functions are assigned to the pins of Port 3, as follows:

Port Pin	Alternate Function
$P_{3.0}$	RxD (serial input port)
$P_{3.1}$	TxD (serial output port)
$P_{3.2}$	$\overline{INT_0}$ (external interrupt 0)
$P_{3.3}$	$\overline{INT_1}$ (external interrupt 1)
$P_{3.4}$	T_0 (Timer 0 external input)
$P_{3.5}$	T_1 (Timer 1 external input)
$P_{3.6}$	$\overline{WR}$ (external data memory write strobe)
$P_{3.7}$	$\overline{RD}$ (external data memory read strobe)

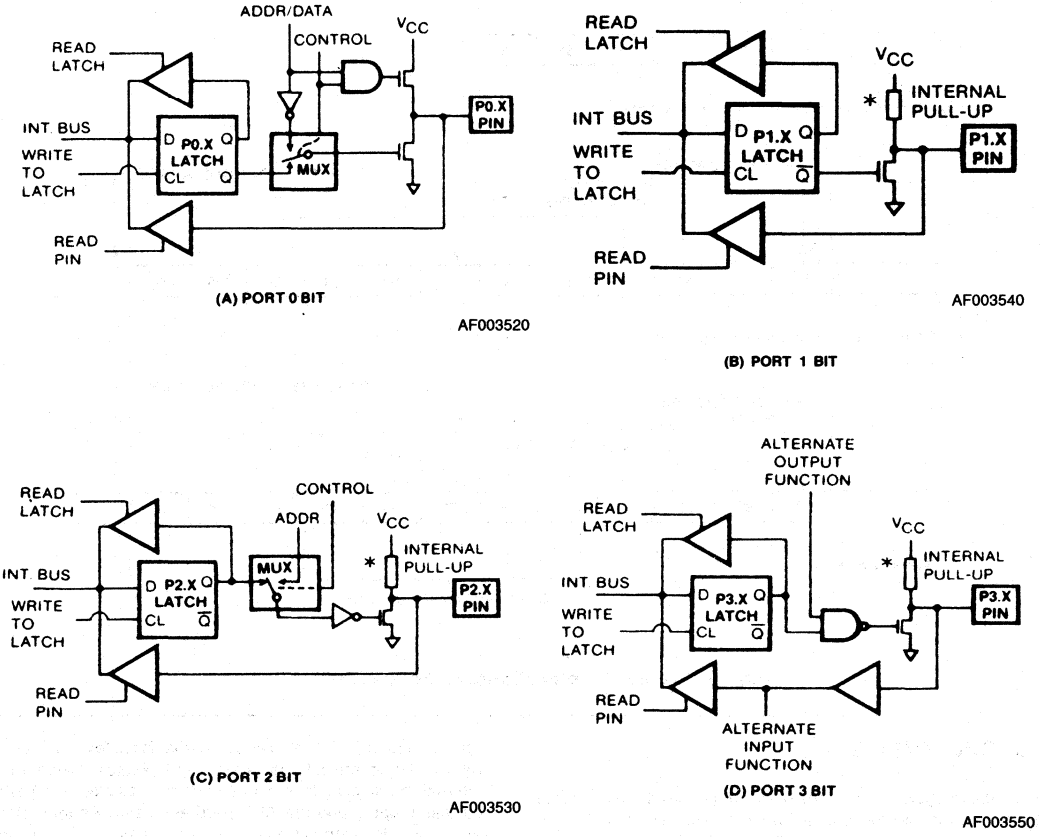


Figure 1-5. 8051 Port Bit Latches and I/O Buffers
*See Figure 1-6 for details of the internal pull-up.

Refer to page 7-1 for Essential Information on Military Devices

1.4.1 I/O Configurations

Figure 1-5 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. More about that in Section 1.4.4.

As shown in Figure 1-5, the output drivers of Ports 0 and 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it.

Also shown in Figure 1-5, is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual P3.X pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pull-ups. Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output. (Ports 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by the internal pull-up, but can be pulled low by an external source.

Port 0 differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 1-5A) is used only when the Port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that condition it can be used as a high-impedance input.

Because Ports 1, 2, and 3 have fixed internal pullups they are sometimes called "quasi-bidirectional" ports. When configured as inputs they pull high and will source current (I_{IL}, in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configured as an input it floats.

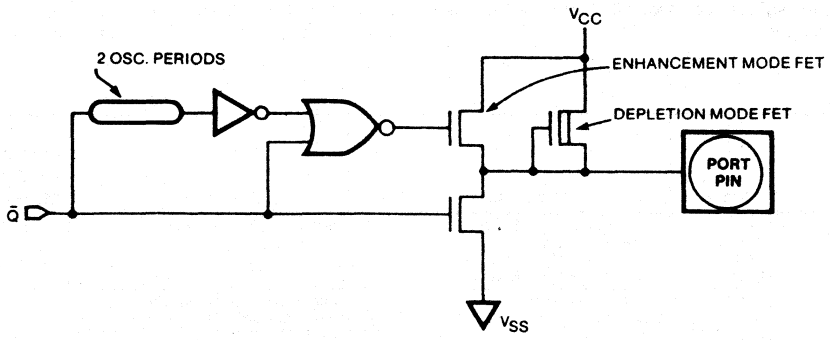
All the port latches in the 8051 have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.

1.4.2 Writing to a Port

In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1). Consequently, the new value in the port latch won't actually appear at the output pin until the next Phase 1, which will be at S1P1 of the next machine cycle.

If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pull-up is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pull-up can source about 100 times the current that the normal pull-up can. It should be noted that the internal pull-ups are field-effect transistors, not linear resistors. The pull-up arrangements are shown in Figure 1-6. It should also be noted that an n-channel FET (nFET) is turned on when a logical 1 is applied to its gate, and is turned off when a logical 0 is applied to its gate.

In NMOS versions of the 8051, the fixed part of the pull-up is a depletion-mode transistor with the gate wired to the source. This transistor will allow the pin to source about 0.25mA when shorted to ground. In parallel with the fixed pull-up is an enhancement-mode transistor, which is activated during S1 whenever the port bit does a 0-to-1 transition. During this interval, if the port pin is shorted to ground, this extra transistor will allow the pin to source an additional 30mA.



NMOS Configuration. The enhancement mode transistor is turned on for 2 osc. periods after $\overline{Q}$ makes a 1-to-0 transition.

Figure 1-6. NMOS Port Writing Internal Pull-up Configurations

1.4.3. Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each drive 4 LS TTL inputs. These ports on NMOS versions can be driven in a normal manner by any TTL or NMOS circuit. Both NMOS and CMOS versions can be driven by open-collector and open-drain outputs, without the need for external pull-ups.

Each pin of Port 0 can be configured as an open drain output or as a high-impedance input. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Rewriting a one (1) to the pin will place its output driver in a high-impedance state and configure the pin as an input. Each I/O pin of Port 0 can sink/source eight LS TTL loads.

1.4.4 Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. The instructions listed below are read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin:

ANL	**(logical AND, e.g., ANL P1,A)**
ORL	**(logical OR, e.g., ORL P2,A)**
XRL	**(logical EX-OR, e.g., XRL P3,A)**
JBC	**(jump if bit = 1 and clear bit, e.g., JBC P1.1, LABEL)**
CPL	**(complement bit, e.g., CPL P3.0)**
INC	**(Increment, e.g., INC P2)**
DEC	**(decrement, e.g., DEC P2)**
DJNZ	**(decrement and jump if not zero, e.g., DJNZ P3, LABEL)**
MOV PX.Y,C	**(move carry bit to bit Y of Port X)**
CLR PX.Y	**(clear bit Y of Port X)**
SET PX.Y	**(set bit Y of Port X)**

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

1.5 ACCESSING EXTERNAL MEMORY

A microprocessor bus is provided to permit the 8051AH to solve a wide range of problems and to allow the upward growth of user products. This multiplexed address and data bus provides an interface compatible with standard memories, 8080 peripherals, and the 8085 compatible memories that include on-chip programmable I/O ports and timing functions. These are summarized in the 8051AH Microcomputer Expansion Components chart of Figure 1-6A.

Accesses to external memory are of two types: accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal $\overline{\text{PSEN}}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (alternate functions of P3.7 and P3.6) to strobe the memory.

Fetches from external Program Memory always use a 16-bit address. Accesses to external Data Memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Whenever a 16-bit address is used, the high byte of the address comes out on Port 2, where it is held for the duration of the read or write cycle. During this time the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle. This will facilitate paging.

Category	AMD Part No.	Description	Comments	Program or Data Memory	Crystal Frequency MHz (Max)
Standard EPROMs	2708	1K x 8 450ns Light Erasable	User programmable and erasable	P	9
	2716-1	2K x 8 350ns Light Erasable		P	11
	2732	4K x 8 450ns Light Erasable		P	9
	2732A	4K x 8 250ns Light Erasable		P	12
Standard RAMs	2114A	1K x 4 100ns RAM	Data memory can be easily expanded using standard NMOS RAMs.	D	12
	2148	1K x 4 70ns RAM		D	12
Standard I/O	8212	8-Bit I/O Port	Serves as Address Latch or I/O port.	D	12
	8255A	Programmable Peripheral Interface	Three 8-bit programmable I/O ports.	D	12
	8251A	Programmable Communications Interface	Serial Communications Receiver/ Transmitter.	D	12
Standard Peripherals	8286	Bi-directional Bus Driver	8080 and 8085 peripheral devices are compatible with the 8051AH allowing easy addition of specializing interfaces.	D	12
	8287	Bi-directional Bus Driver (Inverting)		D	12
Memones with on-chip I/O and Peripheral Functions.	8155-2	256 x 8 330ns RAM		D	12

Figure 1-6A. 8051AH Microcomputer Expansion Components

Refer to page 7-1 for Essential Information on Military Devices

In any case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDR/DATA signal drives both FETs in the Port 0 output buffers. Thus, in this application the Port 0 pins are not open-drain outputs, and do not require external pull-ups. Signal ALE (address latch enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating whatever information the Port 0 SFR may have been holding.

External Program Memory is accessed under two conditions:

1) Whenever signal $\overline{EA}$ is active; or
2) Whenever the program counter (PC) contains a number that is larger than 0FFFH.

This requires that the ROMless versions have $\overline{EA}$ wired low to enable the lower 4K program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC, and during accesses to external Data Memory they output either DPH or the Port 2 SFR (depending on whether the external Data Memory access is a MOVX @DPTR or a MOVX @Ri).

1.5.1 $\overline{PSEN}$

The read strobe for external fetches is $\overline{PSEN}$. $\overline{PSEN}$ is not activated for internal fetches. When the CPU is accessing external Program Memory, $\overline{PSEN}$ is activated twice every cycle (except during a MOVX instruction) whether or not the byte fetched is actually needed for the current instruction. When $\overline{PSEN}$ is activated its timing is not the same as $\overline{RD}$. A complete $\overline{RD}$ cycle, including activation and deactivation of ALE and $\overline{RD}$, takes 12 oscillator periods. A complete $\overline{PSEN}$ cycle, including activation and deactivation of ALE and $\overline{PSEN}$, takes 6 oscillator periods. The execution sequence for these two types of read cycles are shown in Figure 1-7 for comparison.

1.5.2 ALE

The main function of ALE is to provide a properly timed signal to latch the low byte of an address from P0 to an external latch during fetches from external Program Memory. For that purpose ALE is activated twice every machine cycle. This activation takes place even when the cycle involves no external fetch. The only time an ALE pulse doesn't come out is during an access to external Data Memory. The first ALE of the second cycle of a MOVX instruction is missing (see Figure 1-7). Consequently, in any system that does not use external Data Memory, ALE is activated at a constant rate of 1/6 the oscillator frequency, and can be used for external clocking or timing purposes.

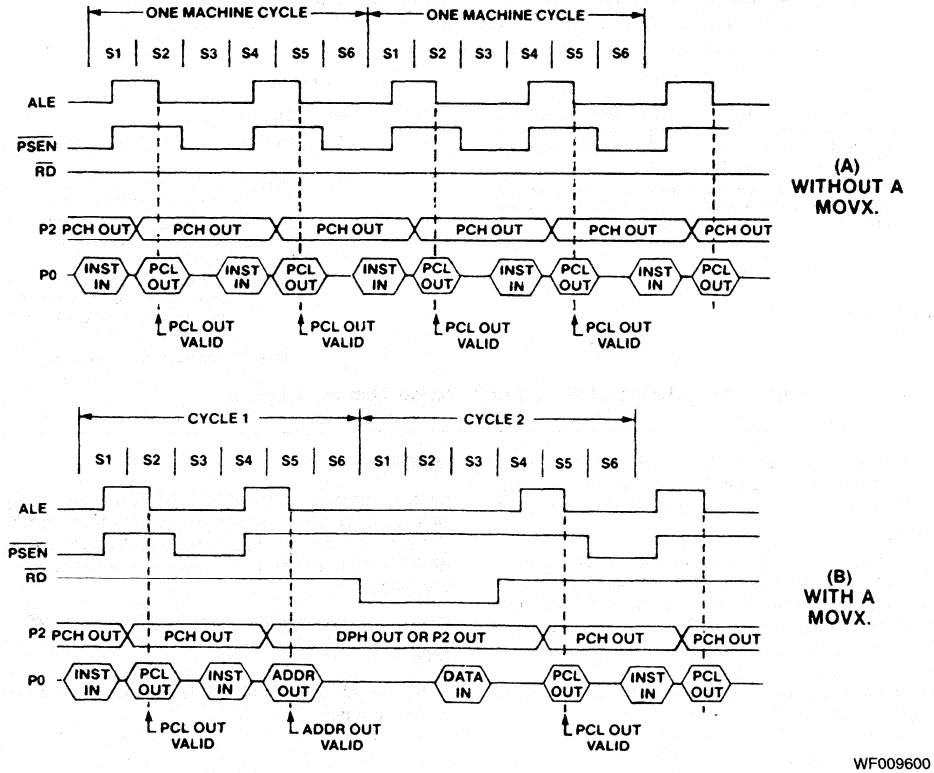


Figure 1-7. External Program Memory Execution

Refer to page 7-1 for Essential Information on Military Devices

1.5.3 Overlapping External Program and Data Memory Spaces

In some applications it is desirable to execute a program from the same physical memory that is being used to store data. In the 8051, the external Program and Data Memory spaces can be combined by ANDing $\overline{PSEN}$ and $\overline{RD}$. A positive-logic AND of these two signals produces an active-low read strobe that can be used for the combined physical memory. Since the $\overline{PSEN}$ cycle is faster than the $\overline{RD}$ cycle, the external memory needs to be fast enough to accommodate the $\overline{PSEN}$ cycle.

1.6 TIMER/COUNTERS

The 8051 has two 16-bit timer/counter registers: Timer 0 and Timer 1. They can be configured to operate either as timers or event counters.

In the "timer" function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the "counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

In addition to the "timer" or "counter" selection, Timer 0 and Timer 1 have four operating modes from which to select.

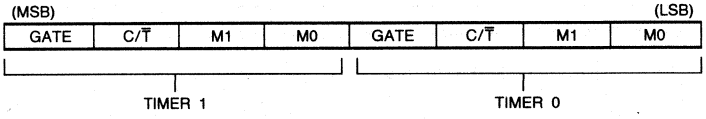
1.6.1 Timer 0 and Timer 1

The "timer" or "counter" function is selected by control bits $C/\overline{T}$ in the Special Function Register TMOD (Figure 6-8). These two timer/counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both timer/counters. Mode 3 is different. The four operating modes are described below.

MODE 0

Putting either Timer into mode 0 makes it look like an 8048 Timer, which is an 8-bit counter with a divide-by-32 prescaler. Figure 1-9 shows the mode 0 operation as it applies to Timer 1.

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements.) TR1 is a control bit in the Special Function Register TCON (Figure 1-10). GATE is in TMOD.



GATE — Gating control When set. Timer/counter "x" is enabled only while "$\overline{INTx}$" pin is high and "TRx" control pin is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.

$C/\overline{T}$ — Timer or Counter Selector Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from "Tx" input pin).

M1	M0	Operating Mode
0	0	MCS-48 Timer "TLx" serves as five-bit prescaler.
0	1	16-bit Timer/Counter "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload timer-counter "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	(Timer 0) TL0 is an eight-bit timer counter-controlled by the standard Timer 0 control bits TH0 is an eight-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer-counter 1 stopped.

Figure 1-8. TMOD: Timer/Counter Mode Control Register

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0 and $\overline{INT0}$ for the corresponding Timer 1 signals in Figure 1-9. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

MODE 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

MODE 2

Mode 2 configures the timer register as an 8-bit counter (TL1) with automatic reload, as shown in Figure 1-11. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Mode 2 operation is the same for Timer/Counter 0.

MODE 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Refer to page 7-1 for Essential Information on Military Devices

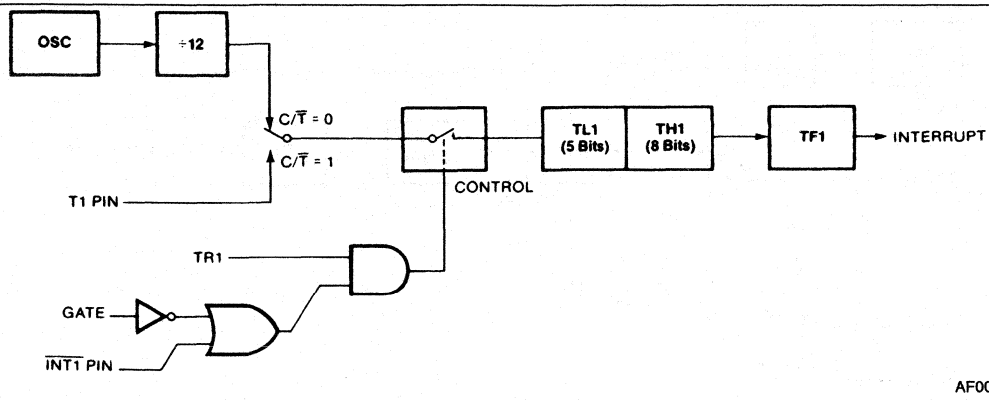


Figure 1-9. Timer/Counter 1 Mode 0: 13-bit Counter

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.	IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn timer/counter on/off.	IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TF0	TCON.5	Timer 0 overflow Flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.	IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn timer/counter on/off.	IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Figure 1-10. TCON: Timer/Counter Control Register

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 1-12. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, an 8051 can look like it has three timer/counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

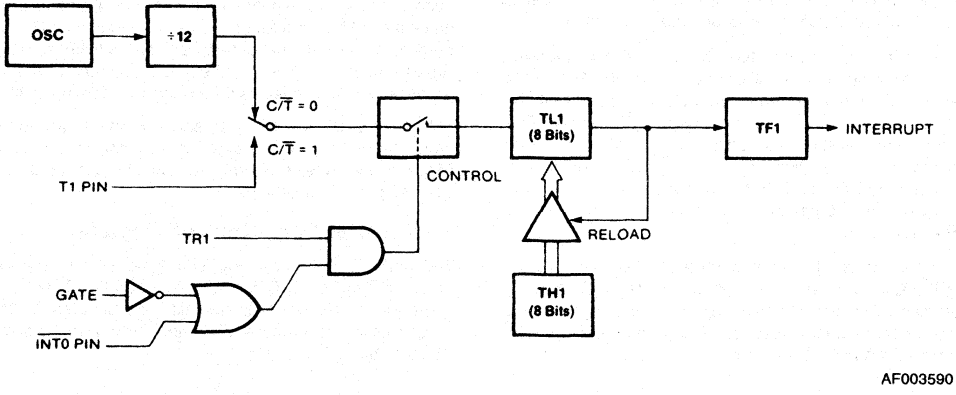


Figure 1-11. Timer/Counter 1 Mode 2: 8-bit Auto-reload

Refer to page 7-1 for Essential Information on Military Devices

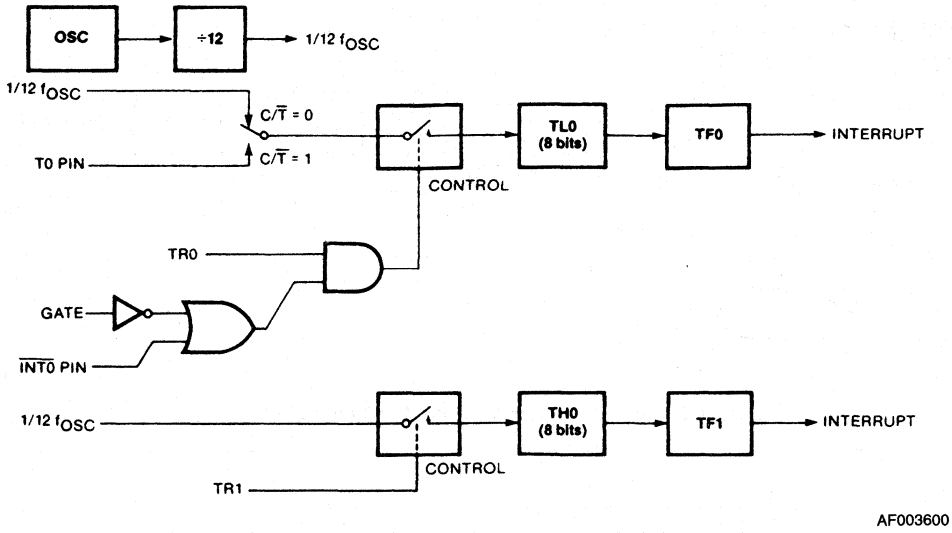


Figure 1-12. Timer/Counter 0 Mode 3: Two 8-bit Counters

1.7 SERIAL INTERFACE

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

Mode 0: Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

1.7.1 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

1.7.2 Serial Port Control Register

The serial port control and status register is the Special Function Register SCON, shown in Figure 1-13. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

Refer to page 7-1 for Essential Information on Military Devices

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	$f_{OSC.}/12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{OSC.}/64$ or $f_{OSC.}/32$
1	1	3	9-bit UART	variable

- SM2 enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
- REN enables serial reception. Set by software to enable reception. Clear by software to disable reception.
- TB8 is the 9th data bit that will be transmitted in modes 2 and 3. Set or clear by software as desired.
- RB8 In modes 2 and 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
- TI is transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
- RI is receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

Figure 1-13. SCON: Serial Port Control Register

1.7.3 Baud Rates

The baud rate in Mode 0 is fixed:

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. (See Figure 1-14A.) If SMOD = 0 (which is its value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD}}{64} \times \text{(Oscillator Frequency)}$$

The baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate.

(MSB)							(LSB)
SMOD	—	—	—	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
—	PCON.6	(Reserved)
—	PCON.5	(Reserved)
—	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000).

Figure 1-14A. PCON: Power Control Register

Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{SMOD}}{32} \times \text{(Timer 1 Overflow Rate)}$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case, the baud rate is given by the formula:

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{SMOD}}{32} \times \frac{\text{OscillatorFrequency}}{12 \times [256 - (TH1)]}$$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Figure 1-14B lists various commonly used baud rates and how they can be obtained from Timer 1.

Refer to page 7-1 for Essential Information on Military Devices

BAUD RATE	f_{osc}	SMOD	TIMER 1		
			C/$\overline{T}$	MODE	RELOAD VALUE
MODE 0 MAX: 1MHZ	12 MHZ	X	X	X	X
MODE 2 MAX: 375K	12 MHZ	1	X	X	X
MODE 1,3:62.5K	12 MHZ	1	0	2	FFH
19.2K	11.059 MHZ	1	0	2	FDH
9.6K	11.059 MHZ	0	0	2	FDH
4.8K	11.059 MHZ	0	0	2	FAH
2.4K	11.059 MHZ	0	0	2	F4H
1.2K	11.059 MHZ	0	0	2	E8H
137.5	11.986 MHZ	0	0	2	1DH
110	6 MHZ	0	0	2	72H
110	12 MHZ	0	0	1	FEE4H

Figure 1-14B. Timer 1 Generated Commonly Used Baud Rates

1.7.4 More About Mode 0

Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Figure 1-15 shows a simplified functional diagram of the serial port in mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal at S6P2 also loads a 1 into the 9th bit position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF," and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0, and also enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1 and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. Both of these actions occur at S1P1 of the 10th machine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and RI = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

1.7.5 More About Mode 1

Ten bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 8051 the baud rate is determined by the Timer 1 overflow rate.

Figure 1-16 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit and receive.

Refer to page 7-1 for Essential Information on Military Devices

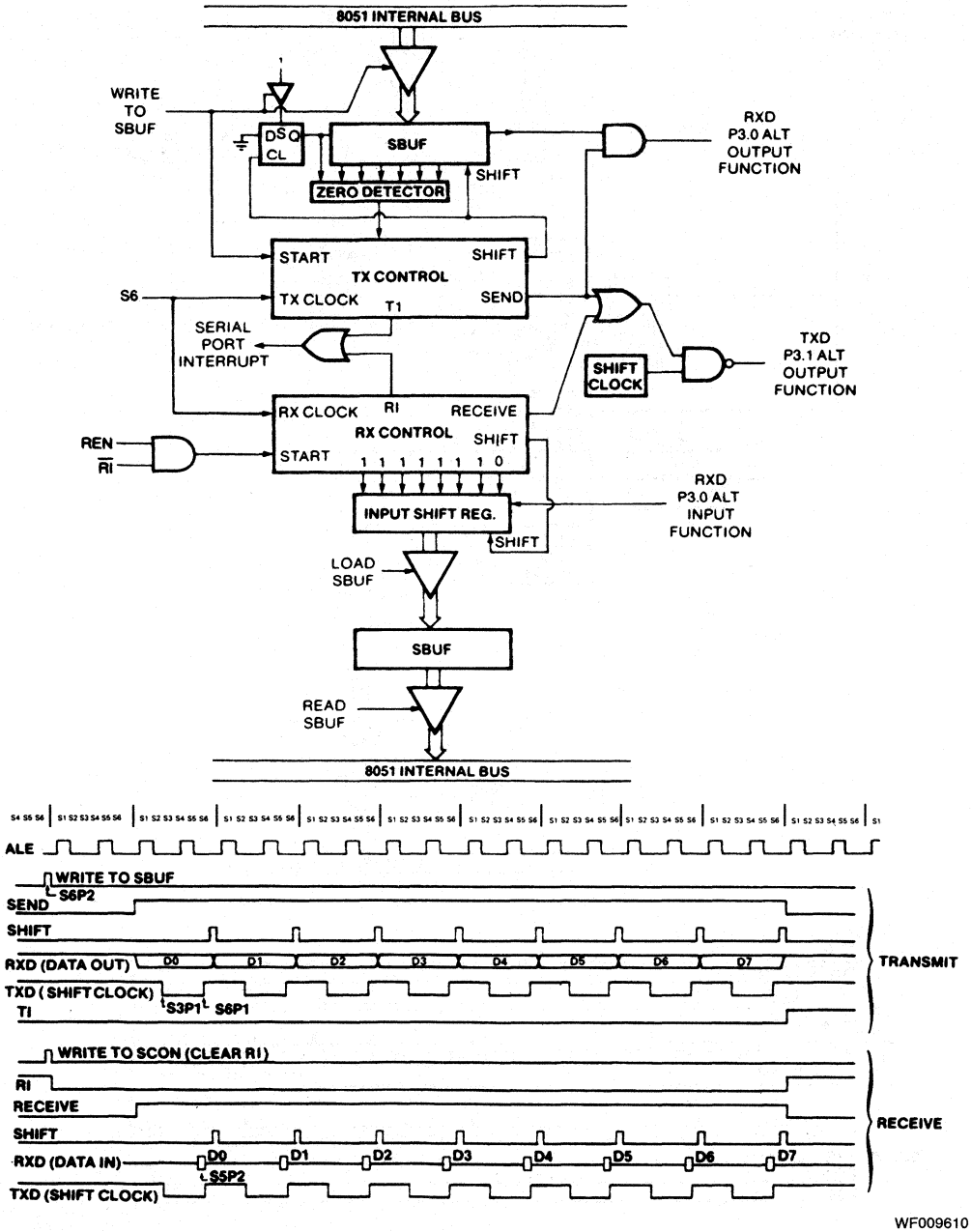
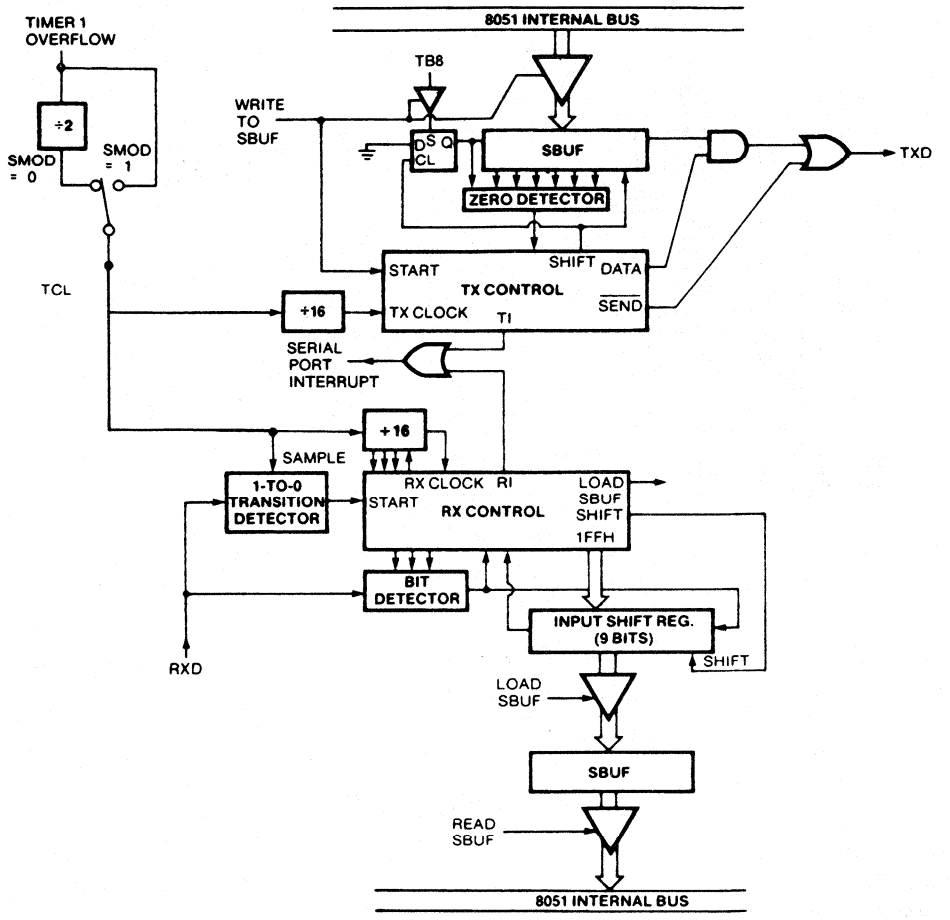
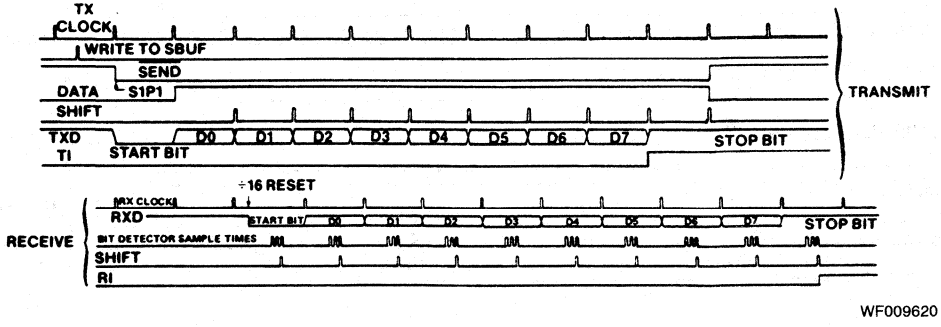


Figure 1-15. Serial Port Mode 0

Refer to page 7-1 for Essential Information on Military Devices





WF009620

Figure 1-16. Serial Port Mode 1

Refer to page 7-1 for Essential Information on Military Devices

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal).

The transmission begins with activation of $\overline{\text{SEND}}$, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeroes. This condition flags the TX Control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set T1. This occurs at the 10th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register, (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1) RI = 0, and
2) Either SM2 = 0, or the received stop bit = 1

If either of these two conditions are not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RXD.

1.7.6 More About Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TD8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figures 1-17A and B show a functional diagram of the serial port in modes 2 and 3. The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.)

The transmission begins with activation of SEND, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeroes are clocked in. Thus, as data bits shift out to the right, zeroes are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeroes. This condition flags the TX Control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set T1. This occurs at the 11th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

Refer to page 7-1 for Essential Information on Military Devices

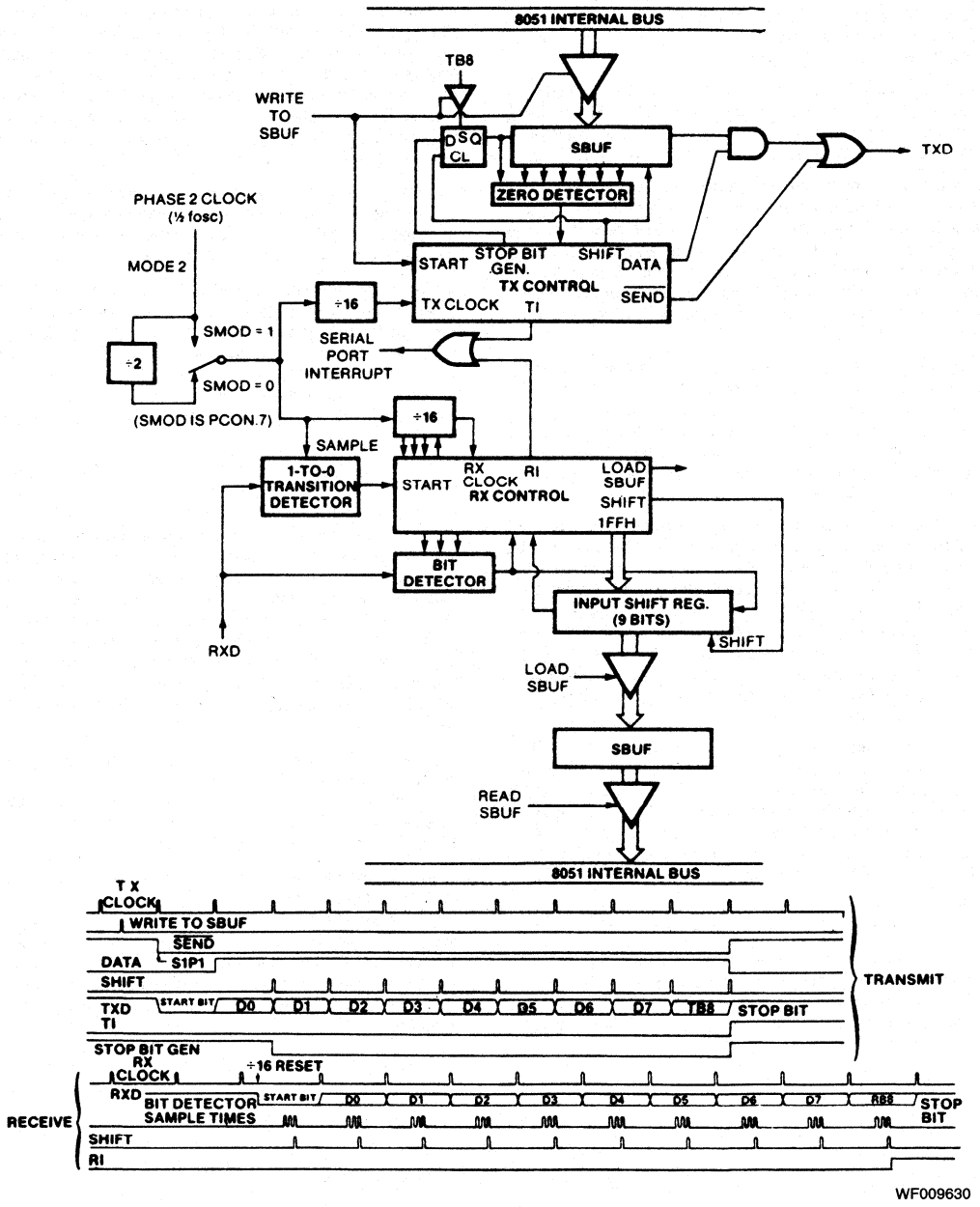
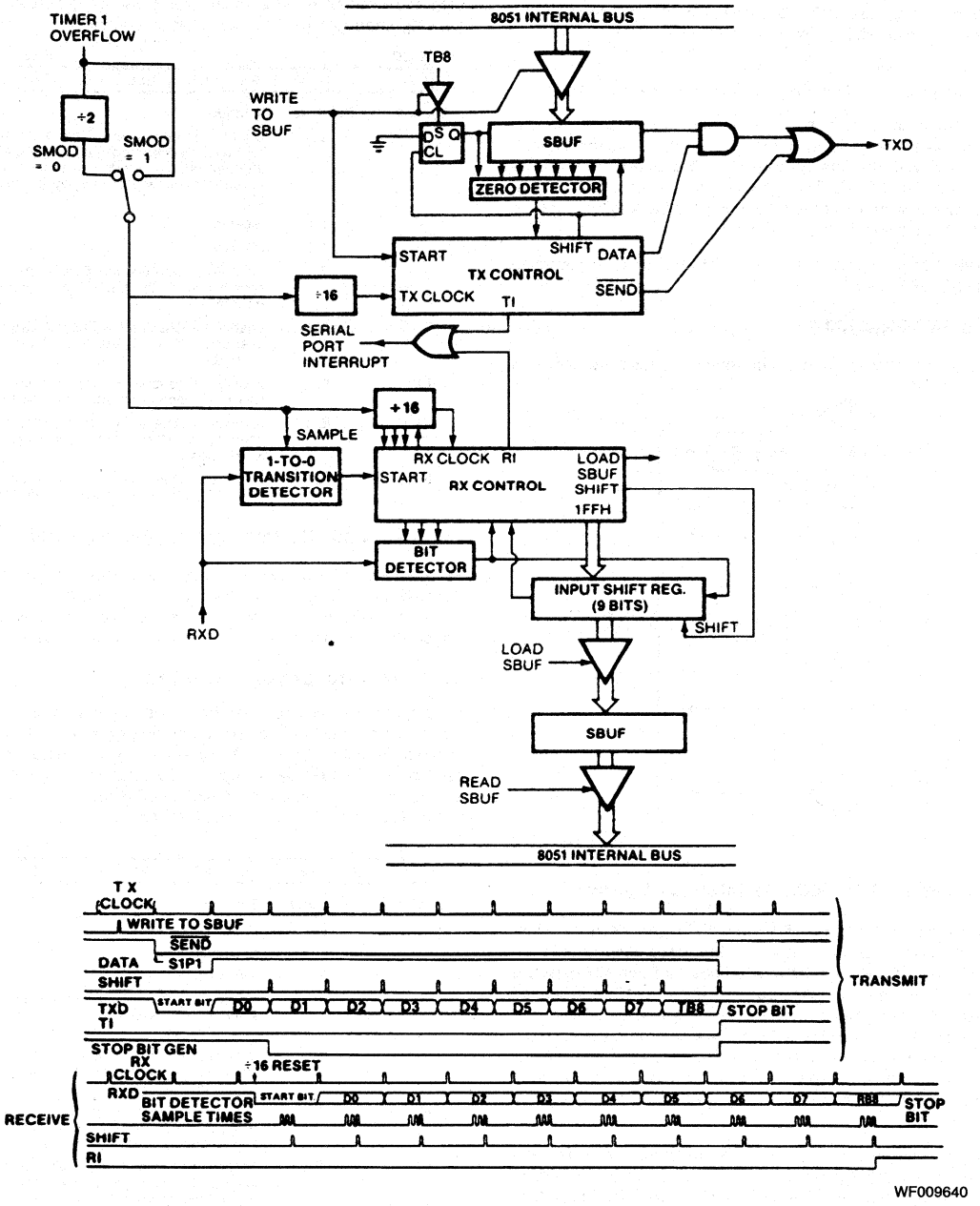


Figure 1-17A. Serial Port Mode 2

Refer to page 7-1 for Essential Information on Military Devices



WF009640

Figure 1-17B. Serial Port Mode 3

Refer to page 7-1 for Essential Information on Military Devices

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and
2) Either SM2 = 0 or the received 9th data bit = 1

If either of these conditions are not met, the received frame is irretrievably lost, and RI is not set. IF both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RXD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

1.8 INTERRUPTS

The 8051 provides 5 interrupt sources. These are shown in Figure 1-18.

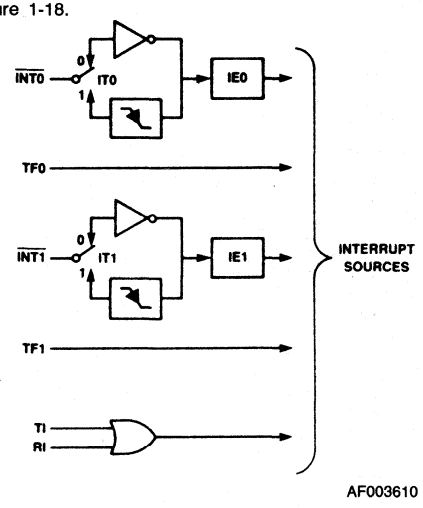


Figure 1-18. MCS-51 Interrupt Sources

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective timer/counter registers (except see Section 1.6.1 for Timer 0 in mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

(MSB)							(LSB)
EA	X	X	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved.
—	IE.5	reserved.
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disableds.
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External interrupt 1. If EX1 = 0, External interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External interrupt 0. If EX0 = 0, External interrupt 0 is disabled.

Figure 1-19. IE: Interrupt Enable Register

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (Figure 1-19). Note that IE contains also a global disable bit, EA, which disables all interrupts at once.

1.8.1 Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP (Figure 1-20). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

(MSB)							(LSB)
X	X	X	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
—	IP.7	reserved.
—	IP.6	reserved.
—	IP.5	reserved.
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	defines the External interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	defines the External interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

Figure 1-20. IP: Interrupt Priority Register

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If

Refer to page 7-1 for Essential Information on Military Devices

requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence, as follows:

	SOURCE	PRIORITY WITHIN LEVEL
1.	IE0	(highest)
2.	TF0	
3.	IE1	
4.	TF1	
5.	RI + TI	(lowest)

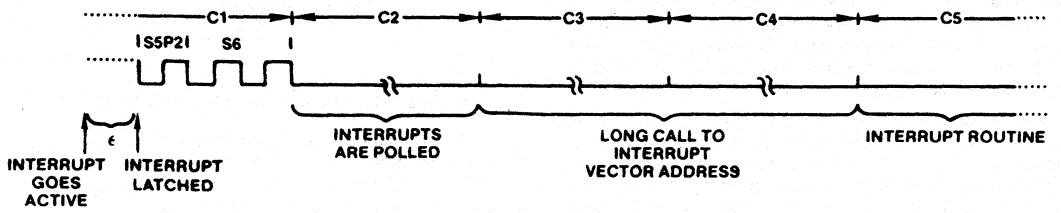
Note that the "priority within level" structure is only used to resolve *simultaneous requests of the same priority level.*

1.8.2 How Interrupts Are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any access to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least *one more* instruction will be executed before any interrupt is vectored to.



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP.

Figure 1-21. Interrupt Response Timing Diagram

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note then that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not *still* active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 1-21.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 1-21, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it doesn't. It never clears the Serial Port flag. This has to be done in the user's software. It clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below.

SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

1.8.3 External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the $\overline{INTx}$ pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the $\overline{INTx}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Refer to page 7-1 for Essential Information on Military Devices

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

1.8.4 Response Time

The $\overline{INT0}$ and $\overline{INT1}$ levels are inverted and latched into IE0 and IE1 at S5P2 of every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 1-21 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long, and if the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 8 cycles.

1.9 SINGLE-STEP OPERATION

The 8051 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be responded to while an interrupt of equal priority level is still in progress, nor will it be responded to after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-step operation is to program one of the external interrupts (say, $\overline{INT0}$) to be level-activated. The service routine for the interrupt will terminate with the following code:

```
JNB     P3.2,$      ;WAIT HERE TILL INTO
                    GOES HIGH
JB      P3.2,$      ;NOW WAIT HERE TILL
                    IT GOES LOW
RETI                :GO BACK AND EXE-
                    CUTE ONE INSTRUCTION
```

Now if the $\overline{INT0}$ pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until $\overline{INT0}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately re-enter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

1.10 RESET

The reset input is the RST pin, which is the input to a Schmitt Trigger.

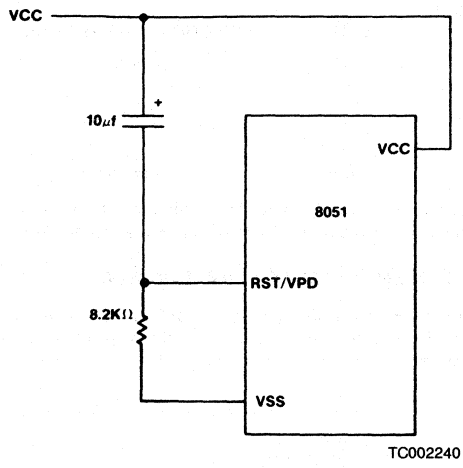


Figure 1-22. Power on Reset Circuit

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), *while the oscillator is running.* The CPU responds by executing an internal reset. It also configures the ALE and $\overline{PSEN}$ pins as inputs. (They are quasi-bidirectional). The internal reset is executed during the second cycle in which RST is high and is repeated every cycle until RST goes low. It leaves the internal registers as follows:

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0 – P3	0FFH
IP	(XX000000)
IE	(0X000000)
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	Indeterminate
PCON	(0XXX0000)

The internal RAM is not affected by reset. When VCC is turned on, the RAM content is indeterminate unless the part is returning from a reduced power mode of operation.

POWER-ON RESET

An automatic reset can be obtained when VCC is turned on by connecting the RST pin to VCC through a 10μf capacitor and to VSS through an 8.2KΩ resistor, providing the VCC risetime

Refer to page 7-1 for Essential Information on Military Devices

does not exceed a millisecond and the oscillator start-up time does not exceed 10 milliseconds. This power-on reset circuit is shown in Figure 1-22. When power comes on, the current drawn by RST commences to charge the capacitor. The voltage at RST is the difference between VCC and the capacitor voltage, and decreases from VCC as the cap charges. The larger the capacitor, the more slowly VRST decreases. VRST must remain above the lower threshhold of the Schmitt Trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

1.11 POWER-SAVING MODES OF OPERATION

For applications where power consumption is a critical factor, both the NMOS and CMOS versions provide reduced power modes of operation. For the CMOS version of the 8051 the reduced power modes, Idle and Power Down, are standard features.

1.11.1 NMOS Power Down Mode

The power down mode in the NMOS devices allows one to reduce VCC to zero while saving the on-chip RAM through a backup supply connected to the RST pin. To use the feature, the user's system, upon detecting that a power failure is imminent, would interrupt the processor in some manner to transfer relevant data to the on-chip RAM and enable the backup power supply to the RST pin before VCC falls below its operating limit. When power returns, the backup supply needs to stay on long enough to accomplish a reset, and then can be removed so that normal operation can be resumed.

1.12 Program Verification

If the program security bit has not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0 – P2.3. The other Port 2 pins and $\overline{PSEN}$ are held low. ALE, $\overline{EA}$, and RST are held high. The contents of the addressed location will come out on Port 0. External pull-ups are required on Port 0 for this operation, as shown in the figure (below). Programming information for the 8751H and Am9761H is in the datasheet.

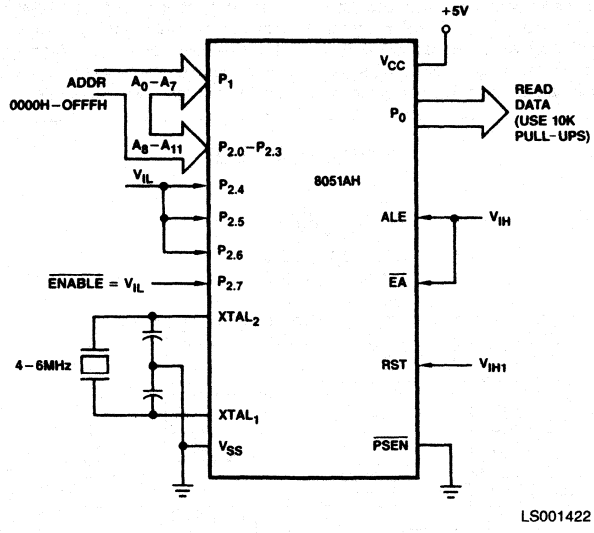


Figure 1-23. Program Verification

Refer to page 7-1 for Essential Information on Military Devices

8051AH/8031AH

Single-Chip 8-Bit Microcomputer

DISTINCTIVE CHARACTERISTICS

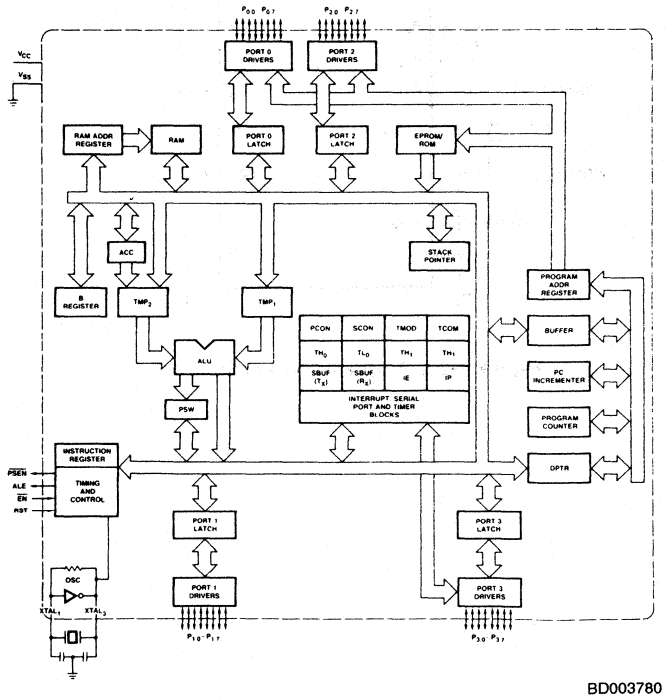
- 4K x 8 ROM, 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters, 5 interrupt sources
- High-Performance full-duplex serial channel
- Boolean processor, cycle multiply and divide
- External memory expandable to 128K bytes
- Most instructions execute in 1μs
- 4μs multiply and divide
- 8031AH
  - Control oriented CPU with RAM and I/O
- 8051AH
  - An 8031AH with factory mask-programmable ROM
- 100mA typical supply current

GENERAL DESCRIPTION

The 8051AH/8031AH are members of a family of advanced single-chip microcomputers. The 8051AH contains 4K x 8 read-only program memory; 128 x 8 RAM; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multiprocessor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The 8031AH is identical, except that it lacks the program memory. For systems that require extra capability, the 8051AH can be expanded using standard TTL compatible memories and the byte oriented 8080 and 8085 peripherals.

The 8051AH microcomputer, like its 8048 predecessor, is efficient both as a controller and as a Boolean processor. The 8051AH has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs, and multiply and divide require only 4μs. Among the many instructions added to the standard 8048 instruction set are multiply, divide, subtract, and compare.

BLOCK DIAGRAM



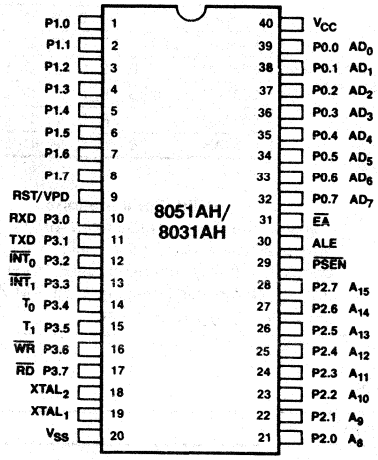
BD003780

Figure 1.

03303C

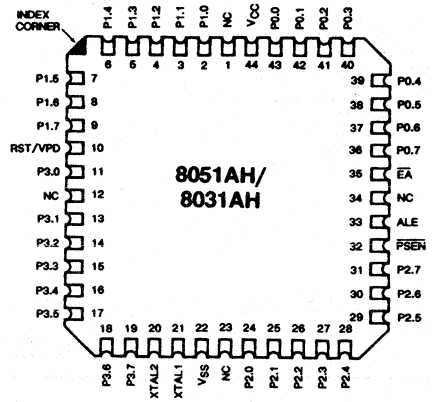
Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40, P-40



CD005550

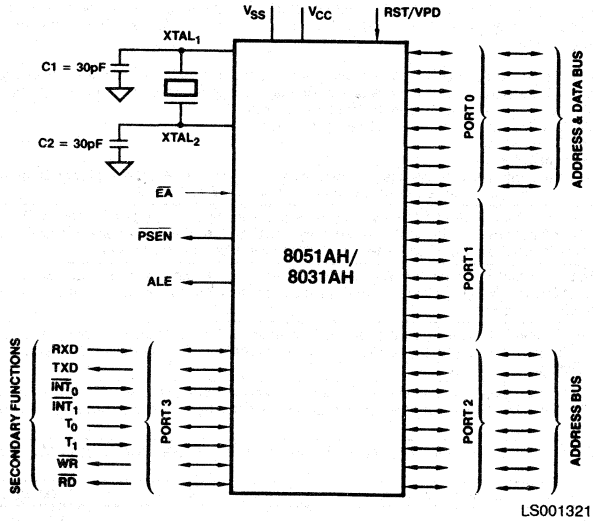
SURFACE MOUNT PINOUT



CD005830

Figure 2.

LOGIC SYMBOL

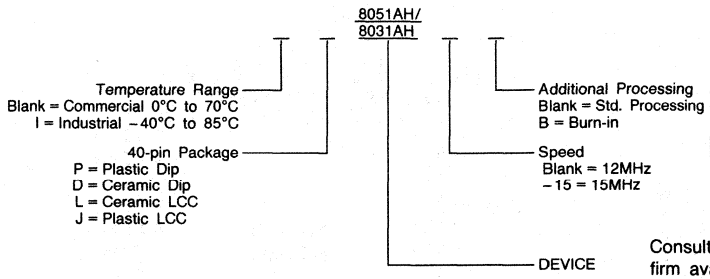


LS001321

Figure 3.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations			
8031AH			P, D, IP, ID, L
8031AH	15		P, D, IP, ID, L
8031AH		B	P, D, IP, ID, L
8031AH	15	B	P, D, IP, ID, L
8051AH			P, D, IP, ID, L
8051AH		B	P, D, IP, ID, L

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Name	Description
V_{SS}	Circuit ground potential.
V_{CC}	+5V power supply during operation.
PORT 0	Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
PORT 1	Port 1 is an 8-bit quasi-bidirectional I/O port with internal pull-ups. It is also used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.
PORT 2	Port 2 is an 8-bit quasi-bidirectional I/O port with internal pull-ups. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
PORT 3	Port 3 is an 8-bit quasi-bidirectional I/O port with internal pull-ups. It also contains the interrupt, timer, serial port, and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. The secondary functions are assigned to the pins of Port 3, as follows: – RXD/Data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous). – TXD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous). – $\overline{INT_0}$ (P3.2). Interrupt 0 input or gate control input for counter 0. – $\overline{INT_1}$ (P3.3). Interrupt 1 input or gate control input for counter 1. – T_0 (P3.4). Input to counter 0. – T_1 (P3.5). Input to counter 1. – $\overline{WR}$ (P3.6) The write control signal latches the data byte from Port 0 into the External Data Memory. – $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.
RST/V_{PD}	A high level on this pin resets the 8051AH. If V_{PD} is held within its spec (approximately +5V) while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is LOW, the RAM's current is drawn from V_{CC}. A small external pull-down resistor (≈8.2kΩ) permits power-on reset using a capacitor connected to V_{CC}.
ALE	Address Latch Enable output used for latching the low order byte of address during access to external memory. It is activated every six oscillator periods, except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source eight LS TTL inputs.
$\overline{PSEN}$	The Program Store Enable output is a control signal that enables the external Program Memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
$\overline{EA}$	When held at a TTL high level, the 8051AH executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the 8051AH fetches all instructions from external Program Memory. Do not float $\overline{EA}$ during normal operation.
$XTAL_1$	Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on $XTAL_2$.
$XTAL_2$	Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5 to +7.0V
Power Dissipation 1W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	Ambient Temperature	V_{CC}	V_{SS}
Commercial			
D8051AH D8031AH P8051AH P8031AH	0°C to 70°C	5V ± 10%	0V
Industrial			
ID8051AH ID8031AH IP8051AH IP8031AH	−40°C to 85°C	5V ± 10%	0V
Commercial			
L8031AH J8031AH L8051AH J8051AH	0°C to 70°C	5V ± 10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS $T_A = 0$ to 70°C; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$)

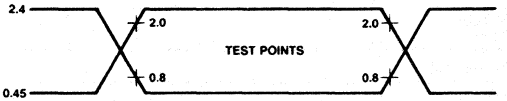
Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		−0.5		0.8	V
V_{IH}	Input High Voltage (Except RST/V_{PD} and $XTAL_2$)		2.0		V_{CC}+0.5	V
V_{IH1}	Input High Voltage to RST/V_{PD} for Reset, $XTAL_2$	$XTAL_1$ to V_{SS}	2.5		V_{CC}+0.5	V
V_{PD}	Power Down Voltage to RST/V_{PD}	V_{CC} = OV	4.5		5.5	V
V_{OL}	Output Low Voltage, Ports 1, 2, 3 (Note 1)	I_{OL} = 1.6mA			0.45	V
V_{OL1}	Output Low Voltage, Port 0, ALE, $\overline{PSEN}$ (Note 1)	I_{OL} = 3.2mA			0.45	V
V_{OH}	Output High Voltage, Ports 1, 2, 3	$I_{OH} = -80\mu A$	2.4			V
V_{OH1}	Output High Voltage, Port 0, ALE, $\overline{PSEN}$	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Logical 0 Input Current, Ports 1, 2, 3	V_{IL} = 0.45V			−800	μA
I_{IL2}	Logical 0 Input Current for $XTAL_2$	$XTAL_1 = V_{SS}$ V_{IN} = 0.45V			−2.5	mA
I_{IH1}	Input High Current to RST/V_{PD} for Reset	$V_{IN} = V_{CC} - 1.5V$			500	μA
I_{LI}	Input Leakage Current to Port 0, $\overline{EA}$	$0.45 < V_{IN} < V_{CC}$			±10	μA
I_{CC}	Power Supply Current	All Outputs Disconnected		100	125	mA
I_{PD}	Power Down Current	V_{CC} = OV			10	mA
C_{IO}	Capacitance of I/O Buffer	fc = 1MHz			10	pF

Note: 1. V_{OL} is degraded when the 8051AH rapidly discharges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the 8051AH as possible.

Datum	Emitting Ports	Degraded I/O Lines	V_{OL} (peak) (max)
Address	P2, P0	P1, P3	0.8V
Write Data	P0	P1, P3, ALE	0.8V

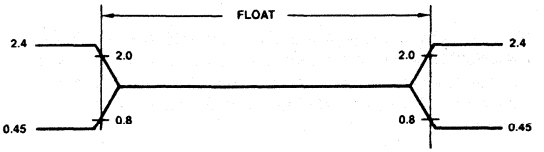
SWITCHING TEST INPUT/OUTPUT AND FLOAT WAVEFORMS

INPUT/OUTPUT



WF007040

FLOAT



WF007050

AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0."
Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."
For timing purposes, the float state is defined as the point at which a P_0 pin sinks 3.2mA or sources 400μA at the voltage test levels.

SWITCHING CHARACTERISTICS (T_A = 0 to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; C_L for Port 0, ALE and $\overline{PSEN}$; Outputs = 100pF; C_L for All Other Outputs = 80pF)

		12MHz Clock		15MHz Clock[1]		Varicable Clock 1/TCLCL = 1.2MHz to 15MHz		
Parameters	**Description**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
PROGRAM MEMORY								
TCY	Min Instruction Cycle Time (Note 3)	1000		800		12TCLCL	12TCLCL	ns
TLHLL	ALE Pulse Width	127		93		2TCLCL-40		ns
TAVLL	Address Set-up to ALE	43		37		TCLCL-40		ns
TLLAX	Address Hold After ALE	48		32		TCLCL-35		ns
TLLIV	ALE to Valid Instruction In		233		168		4TCLCL-100	ns
TLLPL	ALE to $\overline{PSEN}$	58		42		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		166		3TCLCL-35		ns
TPLIV	$\overline{PSEN}$ to Valid Instruction In		125		76		3TCLCL-125	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$ (Note 2)		63		47		TCLCL-20	ns
TPXAV	Address Valid After $\overline{PSEN}$ (Note 2)	75		59		TCLCL-8		ns
TAVIV	Address to Valid Instruction In		302		220		5TCLCL-115	ns
TPLAZ	Address Float After $\overline{PSEN}$		+10		+10		+10	ns
EXTERNAL DATA MEMORY								
TRLRH	$\overline{RD}$ Pulse Width	400		302		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		302		6TCLCL-100		ns
TLLAX	Address Hold After ALE (Note 1)	48		32		TCLCL-35		ns
TRLDV	$\overline{RD}$ to Valid Data In		250		170		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		64		2TCLCL-70	ns
TLLDV	ALE to Valid Data In		517		386		8TCLCL-150	ns
TAVDV	Address to Valid Data In		585		438		9TCLCL-165	ns
TAVLL	Address Set-up to ALE	43		37		TCLCL-40		ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	200	300	151	251	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	203		138		4TCLCL-130		ns
TWHLH	$\overline{WR}$ or $\overline{RD}$ High to ALE High	43	123	27	107	TCLCL-40	TCLCL+40	ns
TDVWX	Data Valid to $\overline{WR}$ Transition	23		7		TCLCL-60		ns
TQVWH	Data Set-up Before $\overline{WR}$	433		319		7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	43		27		TCLCL-50		ns
TRLAZ	Address Float After $\overline{RD}$		+10		+10		+10	ns

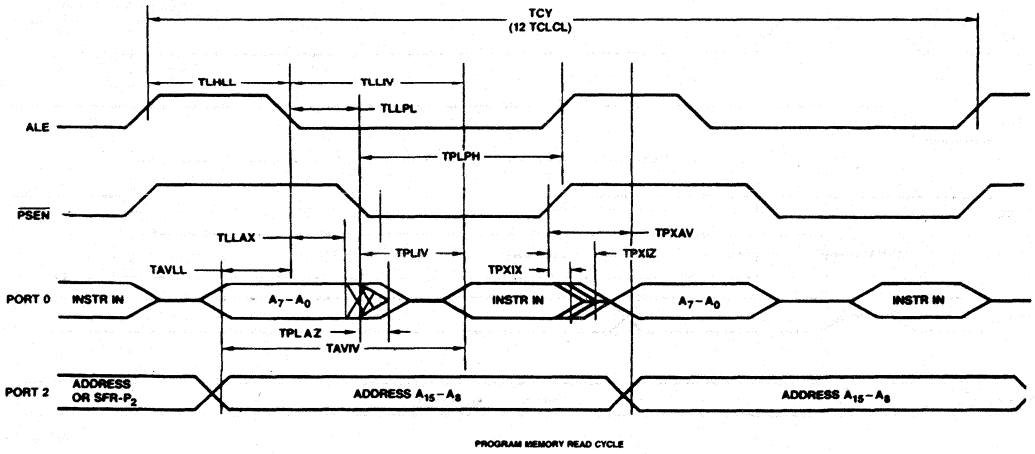
Notes: 1. 15MHz clock pertains only to 8031AH.
2. Interfacing the 8051AH to devices with float times up to 75ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.
3. TCY is the minimum instruction cycle time which consists of 12 oscillator clocks or two ALE cycles.

03303C

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING WAVEFORMS

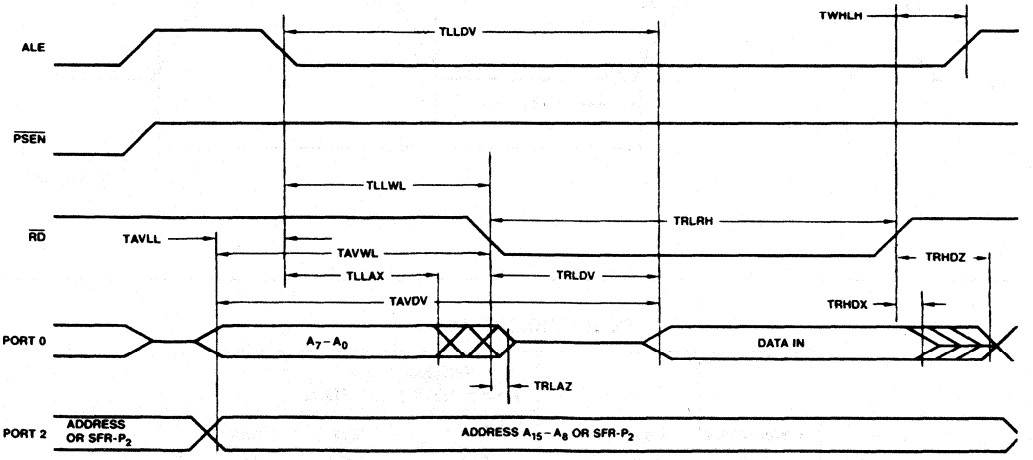
EXTERNAL PROGRAM MEMORY READ CYCLE



PROGRAM MEMORY READ CYCLE

WF007012

EXTERNAL DATA MEMORY READ CYCLE



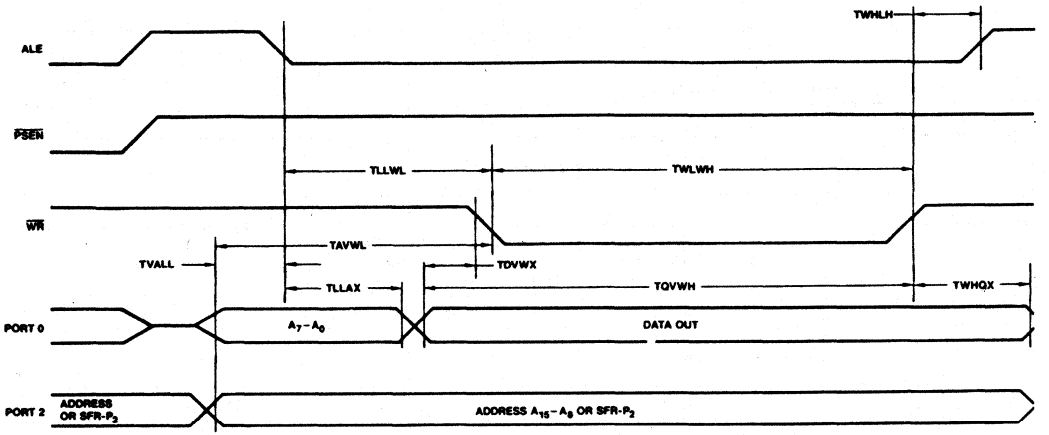
WF007021

03303C

Refer to page 7-1 for Essential Information on Military Devices

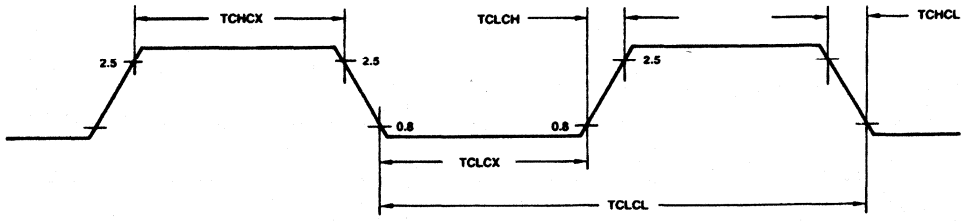
SWITCHING WAVEFORMS (Cont.)

EXTERNAL DATA MEMORY WRITE CYCLE



WF007031

EXTERNAL CLOCK DRIVE $XTAL_2$



WF007060

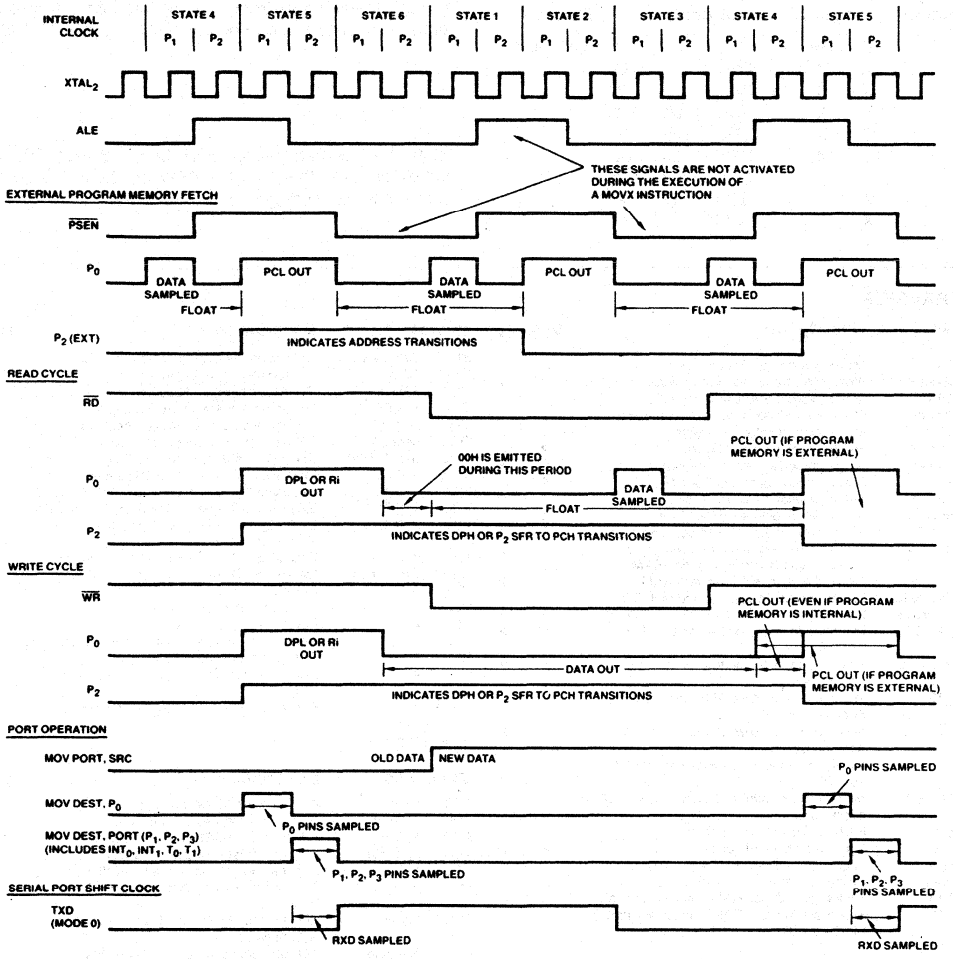
CLOCKING DETAILS

Parameters	Description	Variable Clock Freq = 1.2MHz to 15MHz		Units
		Min	Max	
TCLCL	Oscillator Period	66.7	833.3	ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

03303C

Refer to page 7-1 for Essential Information on Military Devices

CLOCK WAVEFORMS



WF007070

All internal timing is referenced to the internal time state shown on the top of the page. This waveform represents the signal on the X_2 input of the oscillator. This diagram represents when these signals are actually clocked within the chip. However, the time it takes a signal to propagate to the pins is in the range of 25 to 125ns. Prop delays are dependent on many variables, such as temperature, pin loading. Propagation also varies from output to output and component to component. Typically though, /RD and /WR have prop delays of approximately 50ns and the other timing signals approximately 85ns, at room temperature, fully loaded. These differences in prop delays between signals have been integrated into the timing specs.

TABLE 1. 8051AH/8031AH INSTRUCTION SET

INSTRUCTIONS THAT AFFECT FLAG SETTING*

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C, bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request and push the PC; to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (3 to 7μs @ 12MHz).

*Note that operations on SFR byte address 208 or bit addresses 209 – 215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Mode direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Move 16-bit constant to Data Pointer	3	2
MOVC	A,@A + DPTR	Move Code byte relative to DPTR to Accumulator	1	2
MOVC	A,@A + PC	Move Code byte relative to PC to Accumulator	1	2
MOVX	A,@Ri	Move External RAM (8-bit address) to Accumulator	1	2
MOVX	A,@DPTR	Move External RAM (16-bit address) to Accumulator	1	2
MOVX	@Ri,A	Move Accumulator to External RAM (8-bit address)	1	2
MOVX	@DPTR,A	Move Accumulator to External RAM (16-bit address)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte off of stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	1
XCHD	A,@Ri	Exchange indirect RAM's least sig nibble with A's LSN	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry Flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry Flag	2	2
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry Flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry Flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

LOGIC

Mnemonic		Description	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	1
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate Accumulator Left through Carry Flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate Accumulator Right through Carry Flag	1	1
SWAP	A	Exchange nibbles within the Accumulator	1	1

ARITHMETIC

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with carry	1	1
ADDC	A,direct	Add direct byte to Accumulator with Carry Flag	2	1
ADDC	A,@Ri	Add indirect RAM and Carry Flag to Accumulator	1	1
ADDC	A,#data	Add immediate data and Carry Flag to Accumulator	2	1
SUBB	A,Rn	Subtract register from Accumulator with Borrow	1	1
SUBB	A,direct	Subtract direct byte from Accumulator with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from Accumulator with Borrow	1	1
SUBB	A,#data	Subtract immediate data from Accumulator with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
MUL	AB	Multiply Accumulator times B	1	4
DIV	AB	Divide Accumulator by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

OTHER

Mnemonic	Description	Byte	Cyc
NOP	No Operation	1	1

03303C

Refer to page 7-1 for Essential Information on Military Devices

CONTOL TRANSFER (BRANCH)

Mnemonic		Description	Byte	Cyc
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is zero	2	2
JNZ	rel	Jump if Accumulator is not zero	2	2
JC	rel	Jump if Carry Flag is set	2	2
JNC	rel	Jump if carry is not set	2	2
JB	bit,rel	Jump relative if direct bit is set	3	2
JNB	bit,rel	Jump relative if direct bit is not set	3	2
JBC	bit,rel	Jump relative if direct bit is set, then clear bit	3	2
CJNE	A,direct,rel	Compare direct byte to Accumulator and Jump if not Equal	3	2
CJNE	A,#data,rel	Compare immediate to Accumulator and Jump if not Equal	3	2
CJNE	Rn,#data,rel	Compare immediate to reg and Jump if not Equal	3	2
CJNE	@Ri,#data,rel	Compare immediate to indirect RAM and Jump if not Equal	3	2
DJNZ	Rn,rel	Decrement register and Jump if not zero	2	2
DJNZ	direct,rel	Decrement direct byte and Jump if not zero	3	2

CONTROL TRANSFER (SUBROUTINE)

Mnemonic		Description	Byte	Cyc
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from Subroutine Call	1	2
RETI		Return from Interrupt Call	1	2

Notes on data addressing modes:

Rn	–Working register R0 – R7 of the currently selected Register bank.
direct	–128 internal RAM locations, any I/O port, control, or status register.
@Ri	–Indirect internal RAM location addressed by register R0 or R1.
#data	–8-bit constant included in instruction.
#data16	–16-bit constant included as bytes 2 and 3 of instruction.
bit	–128 software flags, any I/O pin, control, or status bit.

Notes on program addressing modes:

addr16	–Destination address for LCALL and LJMP may be anywhere within the 64-Kilobyte program memory address space.
addr11	–Destination address for ACALL and AJMP will be within the same 2-Kilobyte page of program memory as the first byte of the following instruction.
rel	–SJMP and all conditional jumps include as 8-bit offset by Range is +127, –128 bytes relative to first byte of the following instruction.

TABLE 2. INSTRUCTION OPCODES IN HEXADECIMAL ORDER

Hex Code	Bytes	Mnemonic	Operands	Hex Code	Bytes	Mnemonic	Operands
00	1	NOP		2F	1	ADD	A,R7
01	2	AJMP	Code addr	30	3	JNB	Bit addr,code addr
02	3	LJMP	Code addr	31	2	ACALL	Code addr
03	1	RR	A	32	1	RETI	
04	1	INC	A	33	1	RLC	A
05	2	INC	Data addr	34	2	ADDC	A,#data
06	1	INC	@R0	35	2	ADDC	A,data addr
07	1	INC	@R1	36	1	ADDC	A,@R0
08	1	INC	R0	37	1	ADDC	A,@R1
09	1	INC	R1	38	1	ADDC	A,R0
0A	1	INC	R2	39	1	ADDC	A,R1
0B	1	INC	R3	3A	1	ADDC	A,R2
0C	1	INC	R4	3B	1	ADDC	A,R3
0D	1	INC	R5	3C	1	ADDC	A,R4
0E	1	INC	R6	3D	1	ADDC	A,R5
0F	1	INC	R7	3E	1	ADDC	A,R6
10	3	JBC	Bit addr,code addr	3F	1	ADDC	A,R7
11	2	ACALL	Code addr	40	2	JC	Code addr
12	3	LCALL	Code addr	41	2	AJMP	Code addr
13	1	RRC	A	42	2	ORL	Data addr,A
14	1	DEC	A	43	3	ORL	Data addr,#data
15	2	DEC	Data addr	44	2	ORL	A,#data
16	1	DEC	@R0	45	2	ORL	A,data addr
17	1	DEC	@R1	46	1	ORL	A,@R0
18	1	DEC	R0	47	1	ORL	A,@R1
19	1	DEC	R1	48	1	ORL	A,R0
1A	1	DEC	R2	49	1	ORL	A,R1
1B	1	DEC	R3	4A	1	ORL	A,R2
1C	1	DEC	R4	4B	1	ORL	A,R3
1D	1	DEC	R5	4C	1	ORL	A,R4
1E	1	DEC	R6	4D	1	ORL	A,R5
1F	1	DEC	R7	4E	1	ORL	A,R6
20	3	JB	Bit addr,code addr	4F	1	ORL	A,R7
21	2	AJMP	Code addr	50	2	JNC	Code addr
22	1	RET		51	2	ACALL	Code addr
23	1	RL	A	52	2	ANL	Data addr,A
24	2	ADD	A,#data	53	3	ANL	Data addr,#data
25	2	ADD	A,data addr	54	2	ANL	A,#data
26	1	ADD	A,@R0	55	2	ANL	A,data addr
27	1	ADD	A,@R1	56	1	ANL	A,@R0
28	1	ADD	A,R0	57	1	ANL	A,@R1
29	1	ADD	A,R1	58	1	ANL	A,R0
2A	1	ADD	A,R2	59	1	ANL	A,R1
2B	1	ADD	A,R3	5A	1	ANL	A,R2
2C	1	ADD	A,R4	5B	1	ANL	A,R3
2D	1	ADD	A,R5	5C	1	ANL	A,R4
2E	1	ADD	A,R6	5D	1	ANL	A,R5

Hex Code	Bytes	Mnemonic	Operands	Hex Code	Bytes	Mnemonic	Operands
5E	1	ANL	A,R6	AF	2	MOV	R7,data addr
5F	1	ANL	A,R7	B0	2	ANL	C,/bit addr
60	2	JZ	Code addr	B1	2	ACALL	Code addr
61	2	AJMP	Code addr	B2	2	CPL	Bit addr
62	2	XRL	Data addr,A	B3	1	CPL	C
63	3	XRL	Data addr,#data	B4	3	CJNE	A,#data,code addr
64	2	XRL	A,#data	B5	3	CJNE	A,data addr,code addr
65	2	XRL	A,data addr	B6	3	CJNE	@R0,#data,code addr
66	1	XRL	A,@R0	B7	3	CJNE	@R1,#data,code addr
67	1	XRL	A,@R1	B8	3	CJNE	R0,#data,code addr
68	1	XRL	A,R0	B9	3	CJNE	R1,#data,code addr
69	1	XRL	A,R1	BA	3	CJNE	R2,#data,code addr
6A	1	XRL	A,R2	BB	3	CJNE	R3,#data,code addr
6B	1	XRL	A,R3	BC	3	CJNE	R4,#data,code addr
6C	1	XRL	A,R4	BD	3	CJNE	R5,#data,code addr
6D	1	XRL	A,R5	BE	3	CJNE	R6,#data,code addr
6E	1	XRL	A,R6	BF	3	CJNE	R7,#data,code addr
6F	1	XRL	A,R7	C0	2	PUSH	Data addr
70	2	JNZ	Code addr	C1	2	AJMP	Code addr
71	2	ACALL	Code addr	C2	2	CLR	Bit addr
72	2	ORL	C,bit addr	C3	1	CLR	C
73	1	JMP	@A + DPTR	C4	1	SWAP	A
74	2	MOV	A,#data	C5	2	XCH	A,data addr
75	3	MOV	Data addr,#data	C6	1	XCH	A,@R0
76	2	MOV	@R0,#data	C7	1	XCH	A,@R1
77	2	MOV	@R1,#data	C8	1	XCH	A,R0
78	2	MOV	R0,#data	C9	1	XCH	A,R1
79	2	MOV	R1,#data	CA	1	XCH	A,R2
7A	2	MOV	R2,#data	CB	1	XCH	A,R3
7B	2	MOV	R3,#data	CC	1	XCH	A,R4
7C	2	MOV	R4,#data	CD	1	XCH	A,R5
7D	2	MOV	R5,#data	CE	1	XCH	A,R6
7E	2	MOV	R6,#data	CF	1	XCH	A,R7
7F	2	MOV	R7,#data	D0	2	POP	Data addr
80	2	SJMP	Code addr	D1	2	ACALL	Code addr
81	2	AJMP	Code addr	D2	2	SETB	Bit addr
82	2	ANL	C,bit addr	D3	1	SETB	C
83	1	MOVC	A,@A + PC	D4	1	DA	A
84	1	DIV	AB	D5	3	DJNZ	Data addr,code addr
85	3	MOV	Data addr,data addr	D6	1	XCHD	A,@R0
86	2	MOV	Data addr,@R0	D7	1	XCHD	A,@R1
87	2	MOV	Data addr,@R1	D8	2	DJNZ	R0,code addr
88	2	MOV	Data addr,R0	D9	2	DJNZ	R1,code addr
89	2	MOV	Data addr,R1	DA	2	DJNZ	R2,code addr
8A	2	MOV	Data addr,R2	DB	2	DJNZ	R3,code addr
8B	2	MOV	Data addr,R3	DC	2	DJNZ	R4,code addr
8C	2	MOV	Data addr,R4	DD	2	DJNZ	R5,code addr
8D	2	MOV	Data addr,R5	DE	2	DJNZ	R6,code addr
8E	2	MOV	Data addr,R6	DF	2	DJNZ	R7,code addr
8F	2	MOV	Data addr,R7	E0	1	MOVX	A,@DPTR
90	3	MOV	DPTR,#data	E1	2	AJMP	Code addr
91	2	ACALL	Code addr	E2	1	MOVX	A,@R0
92	2	MOV	Bit addr,C	E3	1	MOVX	A,@R1
93	1	MOVC	A,@A + DPTR	E4	1	CLR	A
94	2	SUBB	A,#data	E5	2	MOV	A,data addr
95	2	SUBB	A,data addr	E6	1	MOV	A,@R0
96	1	SUBB	A,@R0	E7	1	MOV	A,@R1
97	1	SUBB	A,@R1	E8	1	MOV	A,R0
98	1	SUBB	A,R0	E9	1	MOV	A,R1
99	1	SUBB	A,R1	EA	1	MOV	A,R2
9A	1	SUBB	A,R2	EB	1	MOV	A,R3
9B	1	SUBB	A,R3	EC	1	MOV	A,R4
9C	1	SUBB	A,R4	ED	1	MOV	A,R5
9D	1	SUBB	A,R5	EE	1	MOV	A,R6
9E	1	SUBB	A,R6	EF	1	MOV	A,R7
9F	1	SUBB	A,R7	F0	1	MOVX	@DPTR,A
A0	2	ORL	C,/bit addr	F1	2	ACALL	Code addr
A1	2	AJMP	Code addr	F2	1	MOVX	@R0,A
A2	2	MOV	C,bit addr	F3	1	MOVX	@R1,A
A3	1	INC	DPTR	F4	1	CPL	A
A4	1	MUL	AB	F5	2	MOV	Data addr,A
A5		Reserved		F6	1	MOV	@R0,A
A6	2	MOV	@R0,data addr	F7	1	MOV	@R1,A
A7	2	MOV	@R1,data addr	F8	1	MOV	R0,A
A8	2	MOV	R0,data addr	F9	1	MOV	R1,A
A9	2	MOV	R1,data addr	FA	1	MOV	R2,A
AA	2	MOV	R2,data addr	FB	1	MOV	R3,A
AB	2	MOV	R3,data addr	FC	1	MOV	R4,A
AC	2	MOV	R4,data addr	FD	1	MOV	R5,A
AD	2	MOV	R5,data addr	FE	1	MOV	R6,A
AE	2	MOV	R6,data addr	FF	1	MOV	R7,A

80C31/80C51

CMOS Single-Component 8-Bit Microcomputer

ADVANCED INFORMATION

DISTINCTIVE CHARACTERISTICS

- 80C31 – Control-Oriented CPU with RAM and I/O
- 80C51 – An 80C31 with on-chip ROM
- CMOS version of the 8051AH/8031AH
- IDLE and POWER-DOWN modes for reduced power consumption
- Low power consumption
  - Normal operation: 24mA @ 5V, 12MHz
  - Idle mode: 3mA @ 5V, 12MHz
  - Power-down mode: 50μA @ 2V
- CMOS and TTL compatible

GENERAL DESCRIPTION

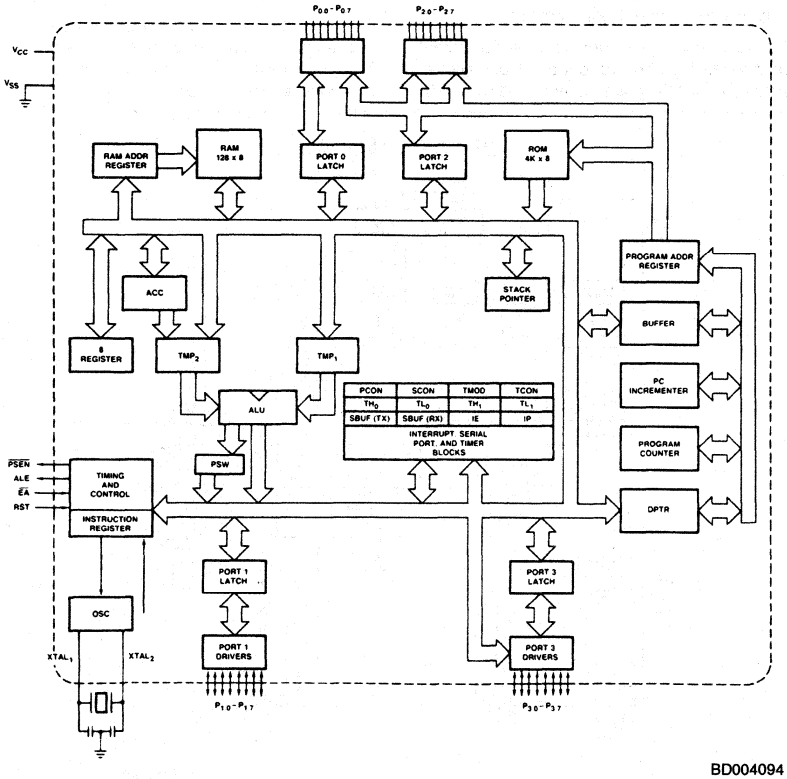
AMD's 80C31 and 80C51 are CMOS versions of the 8051 MOS family members. The 80C51 retains all the features of the 8051: 4K bytes of ROM; 128 bytes of RAM; 32 I/O lines; two 16-bit timers; a 5-source, 2-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuits.

In addition, the 80C51 has two software-selectable modes of reduced activity for further reduction in power consumption.

In the Idle mode, the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the power-down mode, the RAM is saved and all other functions are inoperative.

The 80C31 is identical to the 80C51 except that it has no on-chip ROM.

BLOCK DIAGRAM



BD004094

Figure 1.

04815A

Refer to page 7-1 for Essential Information on Military Devices

8751H/Am9761H

Single-Chip 8-Bit Microcomputer with
4K/8K Bytes of EPROM Program Memory
PRELIMINARY

DISTINCTIVE CHARACTERISTICS

- 4K x 8 EPROM (8751H); 8K x 8 EPROM (Am9761H)
- 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines; programmable serial port
- Two 16-bit Timer/Event counters
- 64K addressable program and data Memory
- Boolean processor
- 5 interrupt sources/2 priority levels
- 4 cycle multiply and divide
- Program memory security feature
- Supports fast adaptive EPROM programming
- Supports silicon signature verification
- 8751H-8 (8MHz option)

GENERAL DESCRIPTION

The 8751H/Am9761H are members of a family of advanced single-chip microcomputers. Both the 8751H, which has 4K bytes of EPROM, and the Am9761H, which has 8K bytes of EPROM, are direct pin-compatible EPROM versions of the 8051AH. Thus, the 8751H/Am9761H are full-speed prototyping tools which provide effective single-chip solutions for highly sensitive controller applications that require code modification flexibility. Refer to Figure 1, which is a block diagram of the 8051 family.

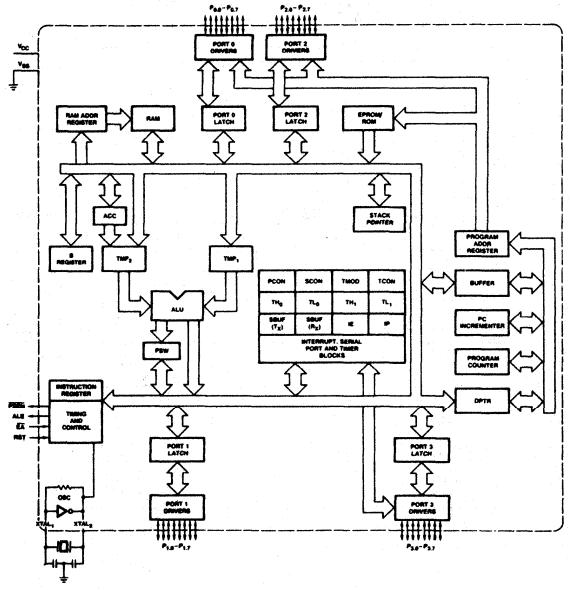
The 8751H/Am9761H devices feature: 32 I/O lines; two 16-bit timer/event counters; a Boolean processor; a 5-source, bi-level interrupt structure; a full-duplex serial channel; and on-chip oscillator and clock circuitry.

Program and data memory are located in independent addresses. The AMD family of microcontrollers can access up to 64K bytes of external Program Memory and up to 64K bytes of external Data Memory. The 8751H and the Am9761H contain the lower 4K and 8K bytes of program memory, respectively, on the chip. Both parts have 128 bytes of on chip read/write data memory.

The AMD 8051 microcontroller family is specifically suited for control applications. A variety of fast addressing modes, which access the internal RAM, facilitates byte processing and numerical operations on small data structures. Included in the instruction set is a menu of 8-bit arithmetic instructions, including 4-cycle multiply and divide instructions.

Extensive on-chip support enables direct bit manipulation and testing of one-bit variables as separate data types. Thus, the device is also suited for control and logic systems that require Boolean processing.

BLOCK DIAGRAM



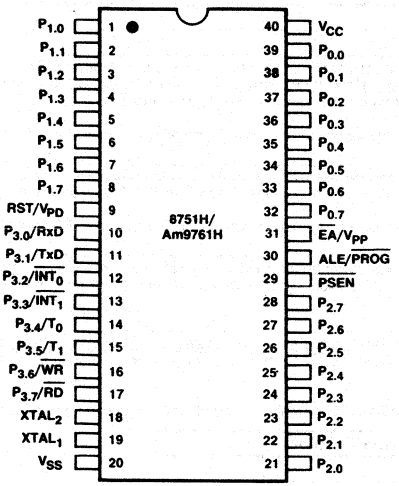
BD004080

Figure 1.

03896A

Refer to page 7-1 for Essential Information on Military Devices

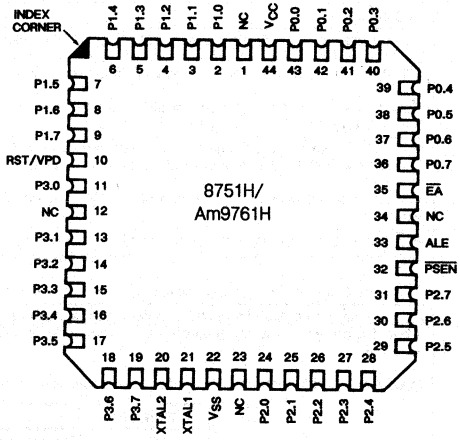
CONNECTION DIAGRAM
Top View



CD005650

Note: Pin 1 is marked for orientation

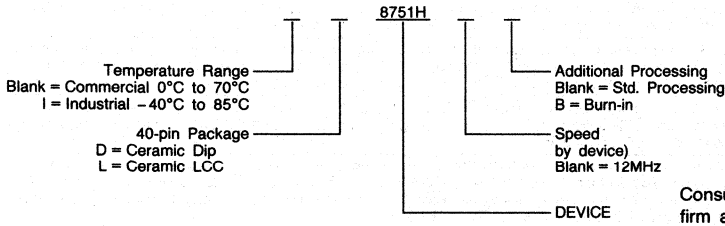
SURFACE MOUNT PINOUT



CD005820

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



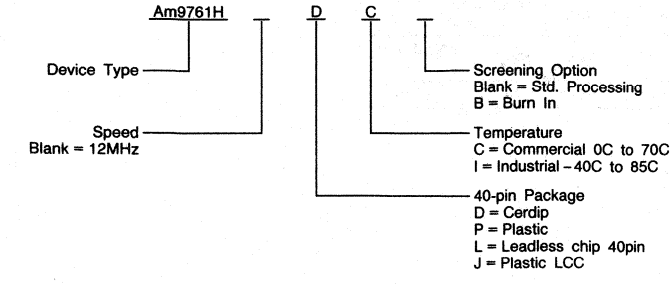
Valid Combinations			
8751H			D
8751H	B		D
8751H		8	D
8751H	B	8	D

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Am9761H	DC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Name	Description
V_{CC}	Supply voltage.
V_{SS}	Circuit ground.
Port 0	Port 0 is an 8-bit open drain bidirectional I/O port. When used as an output port, each of the pins can sink 8 LS TTL inputs. Port 0 pins can be used as high-impedance inputs when placed in the float state by writing 1's to them. Port 0 also serves as the multiplexed low-order address and data bus when accessing external Program and Data Memory. As such, it can source and sink 8 LS TTL inputs. Additionally, instruction bytes are input through the port during EPROM programming and output during program verification. External pull-ups are required during program verification.
Port 1	Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The pins can be used as inputs when written with 1's because they are internally pulled HIGH. In this mode they can actually source current due to the pull-ups (see I_{IL} in DC Characteristics). The port also receives the low-order address bytes during EPROM programming and program verification. Port 1 can sink/source 4 LS TTL inputs.
Port 2	Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. It outputs the high-order address byte when reading from external Program and Data Memory. It also receives the high-order address bits and mode control bits during EPROM programming and program verification. The port 2 output buffers can sink/source 4 LS TTL inputs.
Port 3	Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The pins can sink/source 4 LS TTL inputs. Port 3 pins can be used as inputs when 1's are written to them since the internal pull-ups pull them HIGH. In this state, port 3 pins can source current when pulled LOW externally. In addition, port 3 supports the functions of various special features of the AMD 8051 family. These alternate functions are detailed below:

Port Pin	Alternate Function
$P_{3.0}$	RxD (serial input port)
$P_{3.1}$	TxD (serial output port)
$P_{3.2}$	$\overline{INT_0}$ (external interrupt 0)
$P_{3.3}$	$\overline{INT_1}$ (external interrupt 1)
$P_{3.4}$	T_0 (Timer 0 external input)
$P_{3.5}$	T_1 (Timer 1 external input)
$P_{3.6}$	$\overline{WR}$ (external data memory write strobe)
$P_{3.7}$	$\overline{RD}$ (external data memory read strobe)

RST/V_{PD}	This pin is used to reset the device when held HIGH for two machine cycles while the oscillator is running. If RST/V_{PD} is held within the V_{PD} spec, it will supply standby power to the RAM in the event that V_{CC} drops below its spec. When RST/V_{PD} is LOW, the RAM's bias is drawn from V_{CC}. A small internal register permits power-on reset using a capacitor connected to V_{CC}.
ALE/$\overline{PROG}$	Address Latch Enable output pulse latches the low byte of the address when accessing external memory. ALE pulses are emitted constantly at 1/6 the oscillator frequency except during external Data Memory access when one pulse is skipped. ALE can sink/source 8 LS TTL inputs. The pin also accepts the program pulse input ($\overline{PROG}$) when programming the EPROM.
$\overline{PSEN}$	Program Store Enable is the read strobe to external Program Memory. $\overline{PSEN}$ is activated twice each machine cycle during execution from external Program Memory. In this mode, two $\overline{PSEN}$ activations are skipped during each access to external Data Memory (during a MOVX instruction). $\overline{PSEN}$ is able to source/sink 8 LS TTL inputs.
$\overline{EA}$/V_{PP}	The 8751H executes out of internal Program Memory locations 0000H to OFFFH when External Access enable ($\overline{EA}$) is held HIGH. The Am9761H executes out of internal Program Memory locations 0000H to 1FFFH when $\overline{EA}$ is held HIGH. $\overline{EA}$ must be externally held LOW to enable the device (8751H, Am9761H) to retrieve code from the above mentioned external memory address ranges. This pin also receives the 21V programming supply voltage V_{PP} during programming of the EPROM.
$XTAL_1$	Input to the inverting oscillator amplifier. When an external oscillator is used, $XTAL_1$ should be grounded.
$XTAL_2$	Output of the inverting oscillator amplifier. $XTAL_2$ is also the input for the oscillator signal when using an external oscillator.

TABLE 1. 8751H/Am9761H PERFORMANCE

Clock Frequency		Max UART Baud Rate	
Min	Max	Mode 2	Modes 1, 3
1.2MHz	12MHz	375K	62.5K

03896A

Refer to page 7-1 for Essential Information on Military Devices

PROGRAMMING INFORMATION

Programming the Eprom

To program the EPROM, either the internal or external oscillator must be running at 4 to 6MHz because the internal bus is used to transfer address and program data to the appropriate internal registers.

The 8751H/Am9761H devices support an adaptive EPROM programming algorithm in addition to the conventional EPROM programming algorithm. Adaptive device programming (sometimes called interactive or intelligent programming) adapts to the actual charge storage efficiency of each byte, so that no wasted programming time occurs and minimum device programming time is realized. The typical resulting device programming time is a mere 7% of what is required for a conventional programming algorithm. For example, to program a 4K EPROM similar to the one inside the 8751H using the conventional programming algorithm will require 4K x 50 msec = 200 sec. If adaptive programming is used, the typical programming time required will be 4K x 3 msec = 12 sec. The speed advantage of the adaptive programming is still very significant even allowing for the additional software overhead to implement the adaptive algorithm (2 to 8 sec depending on the brand of EPROM programmer).

To program the 8751H, pins $P_{2.4} - P_{2.6}$ and $\overline{PSEN}$ should be held LOW, and $P_{2.7}$ and RST held HIGH as shown in Table 2a.

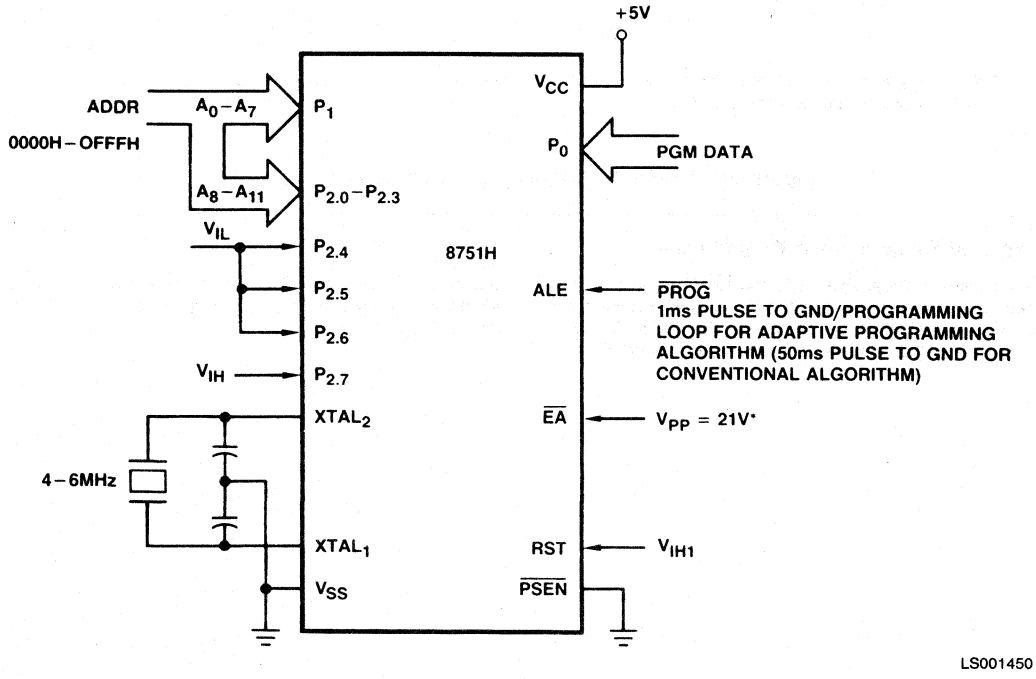
The address of the location to be programmed is applied to Port 1 and $P_{2.0} - P_{2.3}$ while the code byte to be programmed is applied to Port 0 (see Figure 2a).

V_{PP} should be at 21V during device programming and the ALE/$\overline{PROG}$ pin should be pulsed LOW for 1 msec to program the code byte into the addressed EPROM location. The programmed byte is verified immediately after programming.

Figure 2c illustrates the flow of the adaptive programming algorithm. At each address, up to 15 program/verify loops are attempted to verify the programmability of the byte using 1 msec $\overline{PROG}$ pulses. After the programmability of a byte is determined, an overprogramming pulse of 2 msec is applied to $\overline{PROG}$ to guarantee data retention. (This conforms with the AMD standard of 2 msec/byte overprogramming for all N-channel EPROMs.)

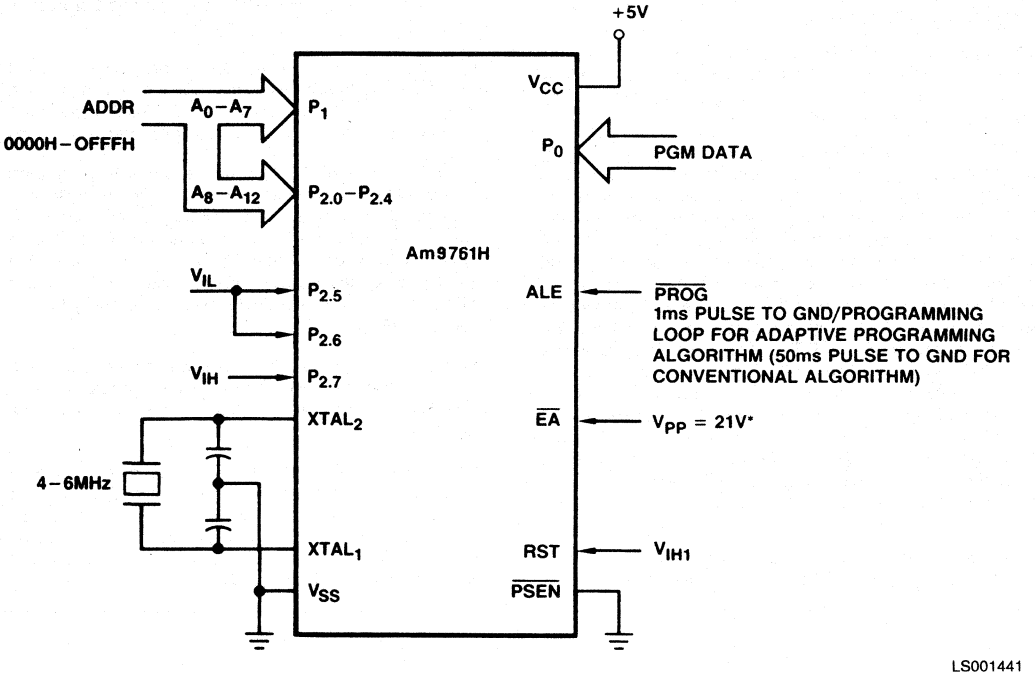
The programming of Am9761H is similar to the above procedures except that pin $P_{2.4}$ is the additional address pin (A_{12}) for accessing the upper 4K bytes of the EPROM (see Figure 2b).

The 8751H/Am9761H can also be programmed using the less efficient conventional EPROM programming algorithm. In this method, V_{PP} is held at 21V and $\overline{PROG}$ is pulsed low for 50 msec to program each code byte into the addressed EPROM location. After the memory is programmed, all addresses would be sequenced and verified.



*When programming, a 0.1×10^{-6}F capacitor is required across V_{PP} and ground to suppress spurious transients which may damage the device.

Figure 2a. 8751H Programming Configuration



*When programming, a 0.1×10^{-6}F capacitor is required across V_{PP} and ground to suppress spurious transients which may damage the device.

Figure 2b. Am9761H Programming Configuration

A Note of Caution when Programming

The maximum voltage applied to the $\overline{EA}/V_{PP}$ pin must not exceed 21.5V at any time as specified for V_{PP}. Even a slight spike can cause permanent damage to the device. The V_{PP} source should thus be well-regulated and glitch-free.

When programming, a 0.1×10^{-6}F capacitor is required across V_{PP} and ground to suppress spurious transients which may damage the device.

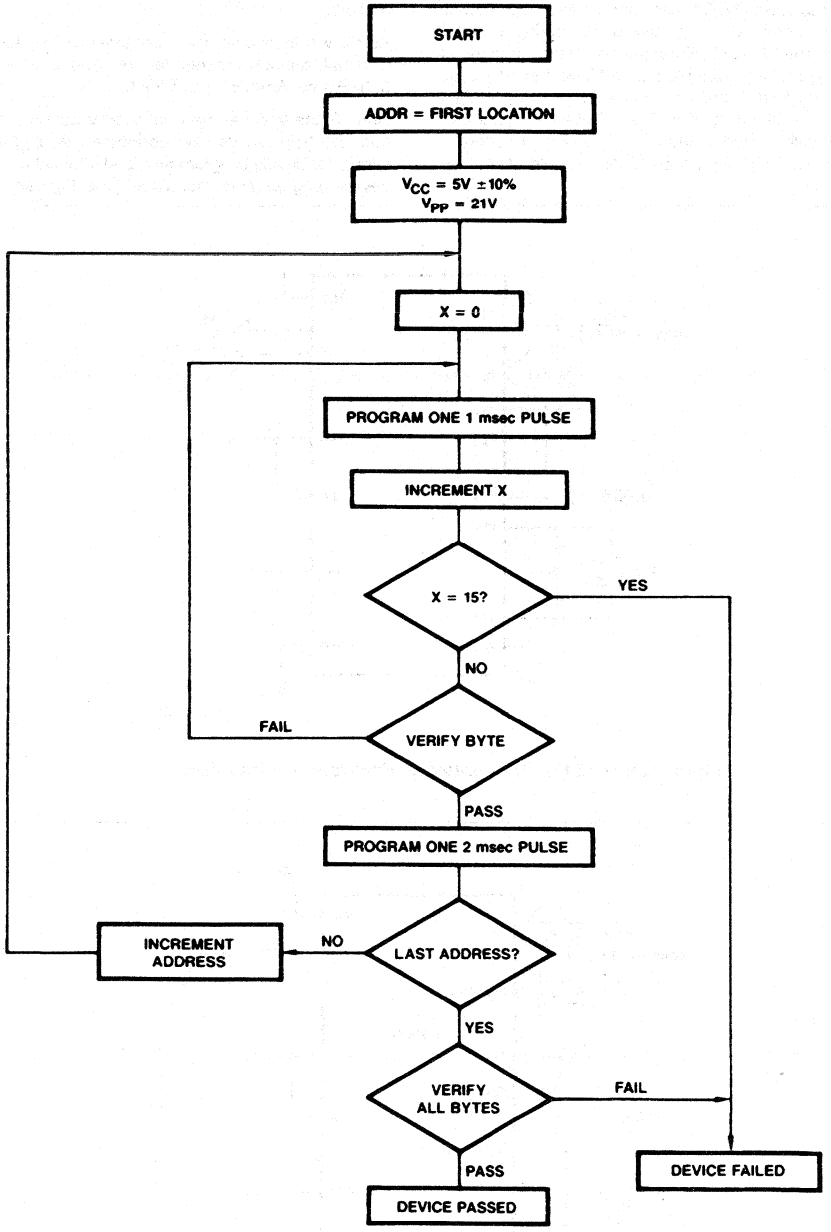


Figure 2c. Adaptive Programming Algorithm for 8751H/Am9761H

PROGRAM VERIFICATION

The Program Memory may be read out for verification purposes when the security bit has not been programmed. Reading the Program Memory may occur during or after programming of the EPROM. When the oscillator is running at 4 – 6MHz, the 8751H Program Memory address location to be read is applied to Port 1 and pins $P_{2.0} - P_{2.3}$ of Port 2. Pins $P_{2.4} - P_{2.6}$ and $\overline{PSEN}$ are held at TTL LOW (see Figure 3a). The Am9761H utilizes Port 1 and pins $P_{2.0} - R_{2.4}$ to address the EPROM, while $P_{2.5} - P_{2.6}$ and $\overline{PSEN}$ are held LOW (see Figure 3b).

The ALE/PROG and RST pins of both devices are held HIGH (RST requires only 2.5V for HIGH) and the $\overline{EA}/V_{PP}$ pin voltage can have any value from 2.0V to 21.5V as shown in Tables 2a and 2b.

Port 0 will then output the contents of the address location. External pull-ups are needed on Port 0 when verifying the 8751H and Am9761H EPROM.

Note: Since V_{PP} can be held at 21V during program verification, the V_{PP} pin can be connected to a static 21V power supply for device programming and verification in the adaptive device programming technique (see Figures 4a and 4b).

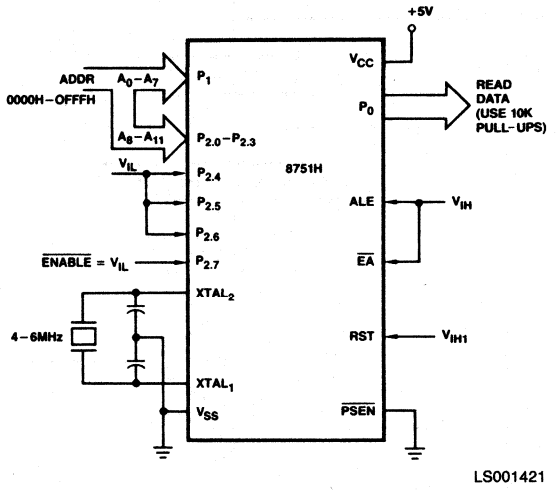


Figure 3a. 8751H Conventional Program Verification

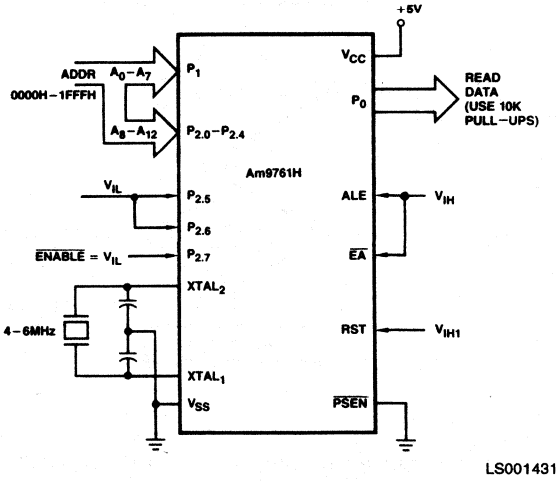


Figure 3b. Am9761H Conventional Program Verification

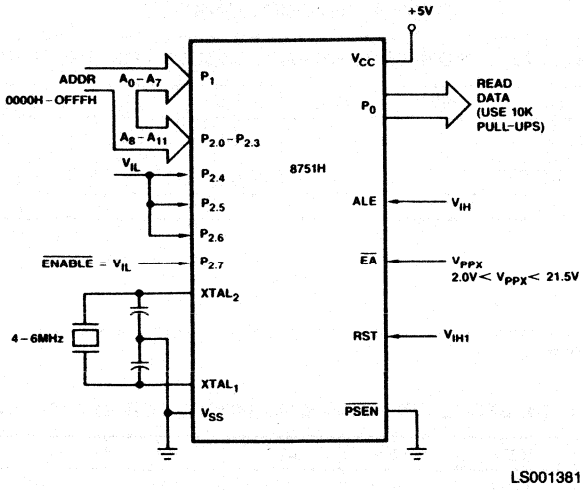


Figure 4a. 8751H Adaptive Program Verification

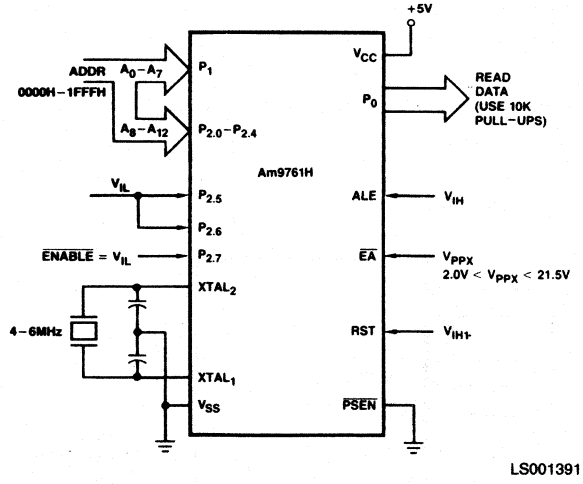


Figure 4b. Am9761H Adaptive Program Verification

SECURITY OF THE EPROM

The 8751H/Am9761H incorporates a security bit, which when activated, prohibits all external readout of the on-chip EPROM contents. Figure 5 illustrates the security bit programming configuration for both the 8751H and Am9761H. To activate the security bit, the same set-up is used as when programming the EPROM except that $P_{2.6}$ is held HIGH. Port 0, Port 1 and pins $P_{2.0} - P_{2.3}$ may assume any state. V_{PP} should be at 21V and the ALE/PROG pin should be pulsed low for 50 msec. The logic states of the other pins are detailed in Tables 2a and 2b.

With the EPROM security bit programmed, retrieval of internal Program Memory cannot be achieved.

A secured program memory looks like a blank array of all ones, and this property can be used to verify that the EPROM is secured. The programmed security bit also prohibits further device programming and the execution of external program memory.

Full functionality and programmability may be restored by erasing the EPROM and thus clearing the security bit.

EPROM CHARACTERISTICS

TABLE 2a. EPROM PROGRAMMING MODES FOR THE 8751H

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	$P_{2.7}$	$P_{2.6}$	$P_{2.5}$	$P_{2.4}$
Program	V_{IH1}	L	L*	V_{PP}	H	L	L	L
Inhibit	V_{IH1}	L	H	X	H	L	L	L
Verify	V_{IH1}	L	H	V_{PPX}	L	L	L	L
Security Set	V_{IH1}	L	L†	V_{PP}	H	H	L	X

Note: "H" = logic high for that pin
"L" = logic low for that pin
"X" = "don't care"
"V_{PP}" = +21V ±0.5V
2.0V < "V_{PPX}" < 21.5V

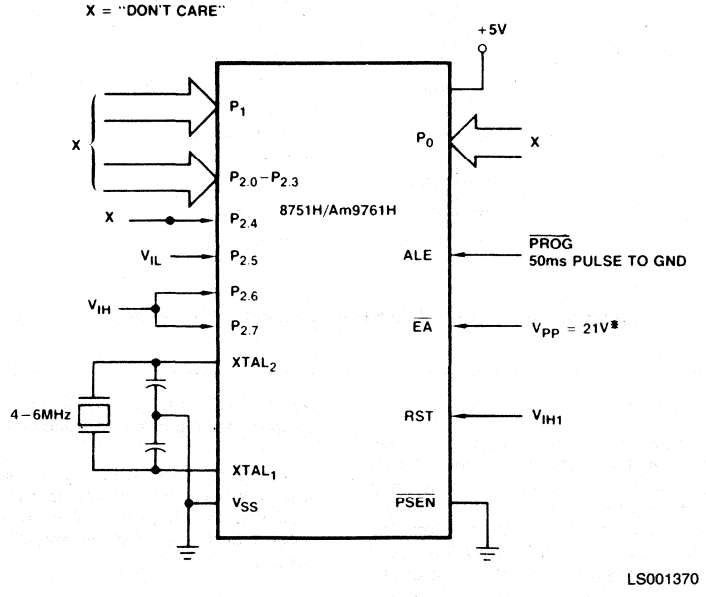
*ALE is pulsed low for 1 msec in the programming loop of the adaptive programming algorithm and is pulsed low for 50 msec if conventional EPROM programming algorithm is used.
†ALE is pulsed low for 50 msec.

TABLE 2b. EPROM PROGRAMMING MODES FOR THE Am9761H

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	$P_{2.7}$	$P_{2.6}$	$P_{2.5}$
Program	V_{IH1}	L	L*	V_{PP}	H	L	L
Inhibit	V_{IH1}	L	H	X	H	L	L
Verify	V_{IH1}	L	H	V_{PPX}	L	L	L
Security Set	V_{IH1}	L	L†	V_{PP}	H	H	L

Note: "H" = logic high for that pin
"L" = logic low for that pin
"X" = "don't care"
"V_{PP}" = +21V±0.5V
2.0V < "V_{PPX}" < 21.5V

*ALE is pulsed low for 1 msec in the programming loop of the adaptive programming algorithm and is pulsed low for 50 msec if conventional EPROM programming algorithm is used.
†ALE is pulsed low for 50 msec.



*When programming, a 0.1×10^{-6}F capacitor is required across V_{PP} and ground to suppress spurious transients which may damage the device.

Figure 5. Programming the Security Bit

SILICON SIGNATURE VERIFICATION

AMD will support silicon signature verification for the 8751H/Am9761H. To ensure that the device can be programmed according to the adaptive EPROM programming algorithm, the manufacturer code and part code can be read from the device before any programming is done.

To read the silicon signature, set up the conditions as specified in Figures 6a and 6b. Note that $P_{2.5}$ is now required to be a TTL high level. Read the first byte of the silicon signature by applying address 0000H to the device; the byte should be a 01H, indicating AMD as the manufacturer. Then read the second byte of the silicon signature by applying address 0001H to the device; the byte should be 0DH, indicating the AMD 8751H/Am9761H product family.

ERASURE CHARACTERISTICS

Light and other forms of electromagnetic radiation can lead to erasure of the EPROM when exposed for extended periods of time.

Wavelengths of light shorter than 4000 angstroms, such as sunlight or indoor fluorescent lighting, can ultimately cause inadvertent erasure and should, therefore, not be allowed to expose the EPROM for lengthy durations (approximately one week in sunlight or three years in room-level fluorescent lighting). It is suggested that the window be covered with an opaque label if an application is likely to subject the device to this type of radiation.

It is recommended that ultraviolet light (of 2537 angstroms) be used to a dose of at least 15 W-sec/cm^2 when erasing the EPROM. An ultraviolet lamp rated at 12,000 uW/cm^2 held one inch away for 20 – 30 minutes should be sufficient.

EPROM erasure leaves the Program Memory in an "all ones" state.

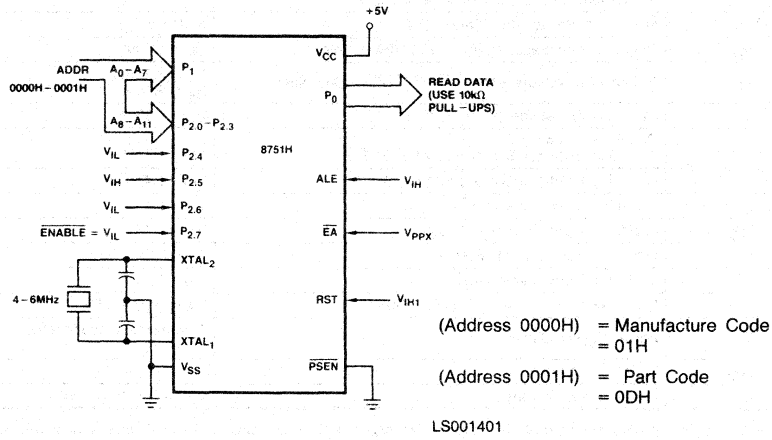


Figure 6a. 8751H Silicon Signature Verification Configuration

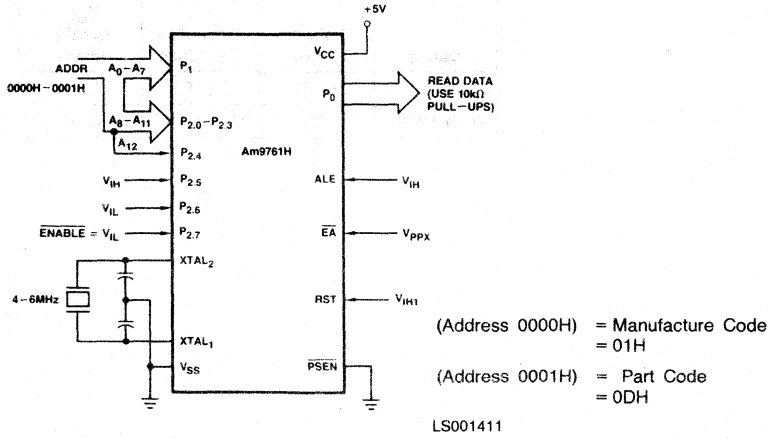


Figure 6b. Am9761H Silicon Signature Verification Configuration

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage on EA/V_{PP} Pin to V_{SS} ..... −0.5 to +21.5V
Voltage on Any Other Pin to V_{SS} ..... −0.5 to +7V
Power Dissipation 2W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	Ambient Temperature	V_{CC}	V_{SS}
8751H			
COMMERCIAL	0° to 70°C	5.0V ± 10%	0V
Am9761H			
COMMERCIAL	0° to 70°C	5.0V ± 10%	0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage (Except $\overline{EA}$)		−0.5	0.8	V
V_{IL1}	Input Low Voltage to $\overline{EA}$		−0.5	0.7	V
V_{IH}	Input High Voltage (Except $XTAL_2$, RST)		2.0	V_{CC}+0.5	V
V_{IH1}	Input High Voltage to $XTAL_2$, RST	$XTAL_1 = V_{SS}$	2.5	V_{CC}+0.5	V
V_{OL}	Output Low Voltage (Ports 1, 2, 3) (See Note 1)	I_{OL} = 1.6mA		0.45	V
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$) (See Note 1)	I_{OL} = 3.2mA I_{OL} = 2.4mA		0.60 0.45	V
V_{OH}	Output High Voltage (Ports 1, 2, 3)	I_{OH} = −80μA	2.4		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	I_{OH} = −400μA	2.4		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45V		−500	μA
I_{IL1}	Logical 0 Input Current ($\overline{EA}$)	V_{IN} = 0.45V		−15	mA
I_{IL2}	Logical 0 Input Current ($XTAL_2$)	V_{IN} = 0.45V		−3.2	mA
I_{LI}	Input Leakage Current (Port 0)	0.45 < V_{IN} < V_{CC}		±100	μA
I_{IH}	Logical 1 Input Current ($\overline{EA}$)	V_{IN} = 2.4V		500	μA
I_{IH1}	Input Current to RST to Activate Reset	V_{IN} < (V_{CC} − 1.5V)		500	μA
I_{CC}	Power Supply Current	All Outputs Disconnected; $\overline{EA}$ = V_{CC}		250	mA
C_{IO}	Pin Capacitance	Test Freq = 1MHz		10	pF
I_{PD}	Power Down Current	V_{CC} = OV, V_{PD} = 5V		10	mA

Note 1. V_{OL} is degraded when the 8051AH rapidly discharges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the 8051AH as possible.

Datum	Emitting Ports	Degraded I/O Lines	V_{OL} (peak) (max)
Address	P2, P0	P1, P3	0.8V
Write Data	P0	P1, P3, ALE	0.8V

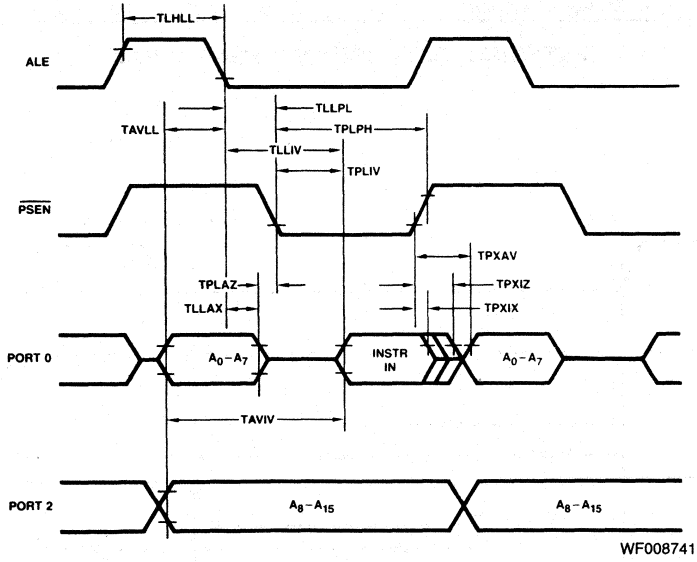
03896A

Refer to page 7-1 for Essential Information on Military Devices

SWITCHING CHARACTERISTICS over operating range unless otherwise specified
(Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100pF, Load Capacitance for All Other Outputs = 80pF)
EXTERNAL PROGRAM MEMORY CHARACTERISTICS

Parameters	Description	12MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 8751H/Am9761H			1.2	12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL-40		ns
TAVLL	Address Set-up to ALE	43		TCLCL-40		ns
TLLAX	Address Hold After ALE	48		TCLCL-35		ns
TLLIV	ALE to Valid Instr In		183		4TCLCL-150	ns
TLLPL	ALE to $\overline{PSEN}$	58		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	190		3TCLCL-60		ns
TPLIV	$\overline{PSEN}$ to Valid Instr In		100		3TCLCL-150	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		63		TCLCL-20	ns
TPXAV	Address Valid After $\overline{PSEN}$	75		TCLCL-8		ns
TAVIV	Address to Valid Instr In		267		5TCLCL-150	ns
TPLAZ	Addr Float After $\overline{PSEN}$		+10		+10	ns

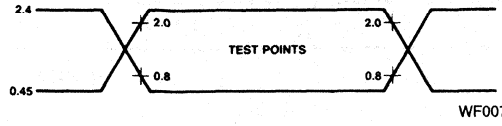
EXTERNAL PROGRAM MEMORY READ CYCLE



WF008741

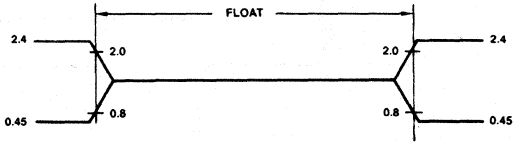
SWITCHING TEST INPUT/OUTPUT AND FLOAT WAVEFORMS

INPUT/OUTPUT



WF007040

FLOAT



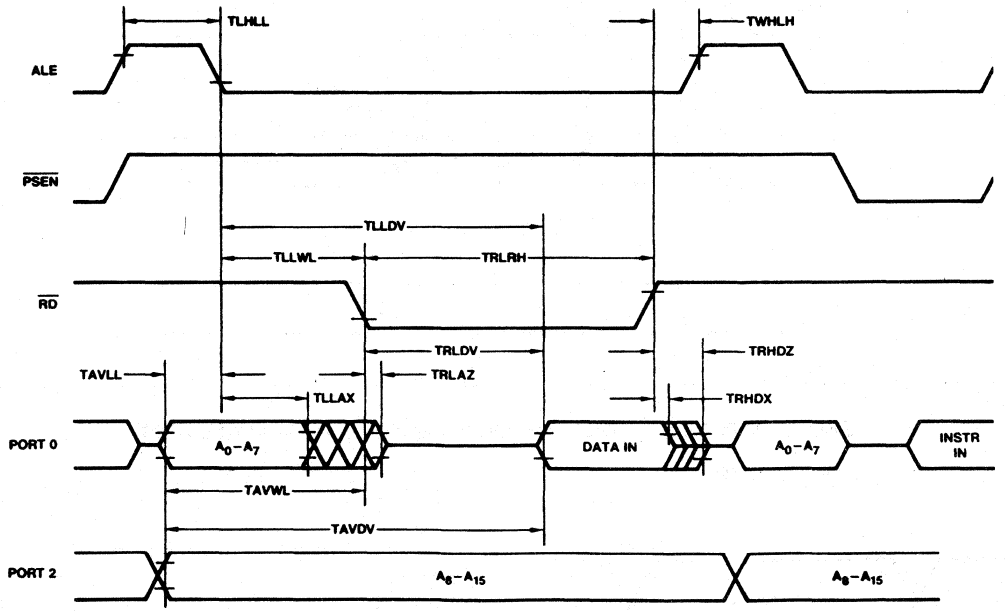
WF007050

AC testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0."
Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0."
For timing purposes, the float state is defined as the point at which a P_0 pin sinks 3.2mA or sources 400μA at the voltage test levels.

EXTERNAL DATA MEMORY CHARACTERISTICS

Parameters	Description	12MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL–100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL–100		ns
TLLAX	Address Hold After ALE	48		TCLCL–35		ns
TRLDV	$\overline{RD}$ to Valid Data In		252		5TCLCL–165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL–70	ns
TLLDV	ALE to Valid Data In		517		8TCLCL–150	ns
TAVDV	Address to Valid Data In		585		9TCLCL–165	ns
TAVLL	Address Set-up to ALE	43		TCLCL–40		ns
TLLWL	ALE to $\overline{RD}$ or $\overline{WR}$	200	300	3TCLCL–50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$	203		4TCLCL–130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	13		TCLCL–70		ns
TQVWH	Data Set-up Before $\overline{WR}$	433		7TCLCL–150		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL–50		ns
TRLAZ	Address Float After $\overline{RD}$	0		0		ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	33	133	TCLCL–50	TCLCL+50	ns

EXTERNAL DATA MEMORY READ CYCLE

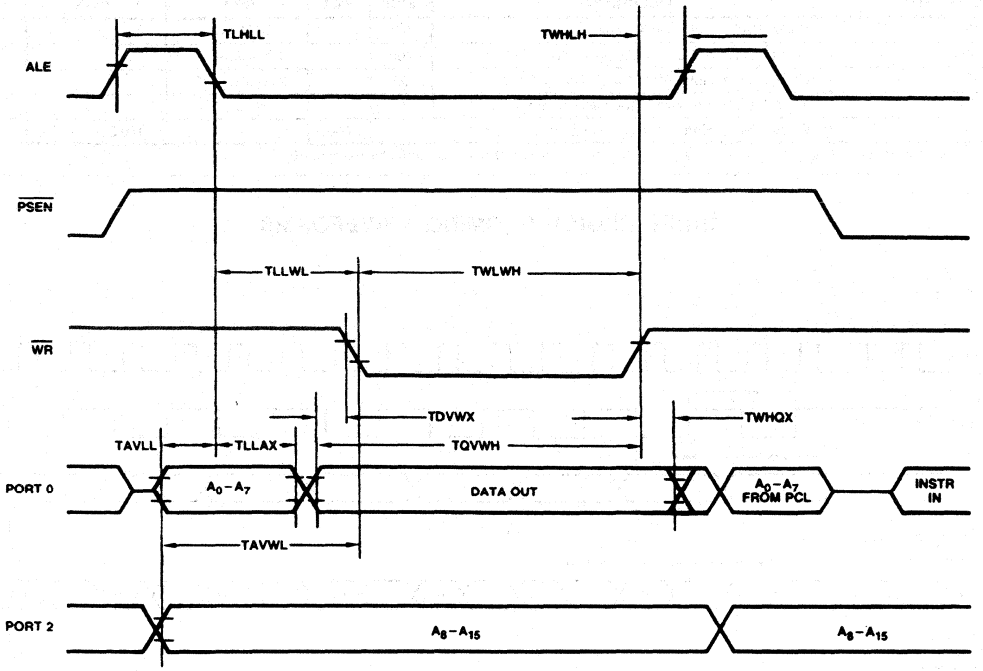


WF008731

03896A

Refer to page 7-1 for Essential Information on Military Devices

EXTERNAL DATA MEMORY WRITE CYCLE

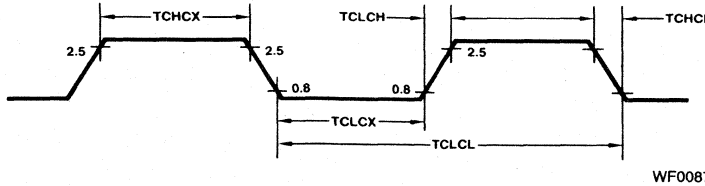


WF008751

EXTERNAL CLOCK DRIVE

Parameters	Description	Min	Max	Units
1/TCLCL	Oscillator Frequency: 8751H/Am9761H	1.2	12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORMS



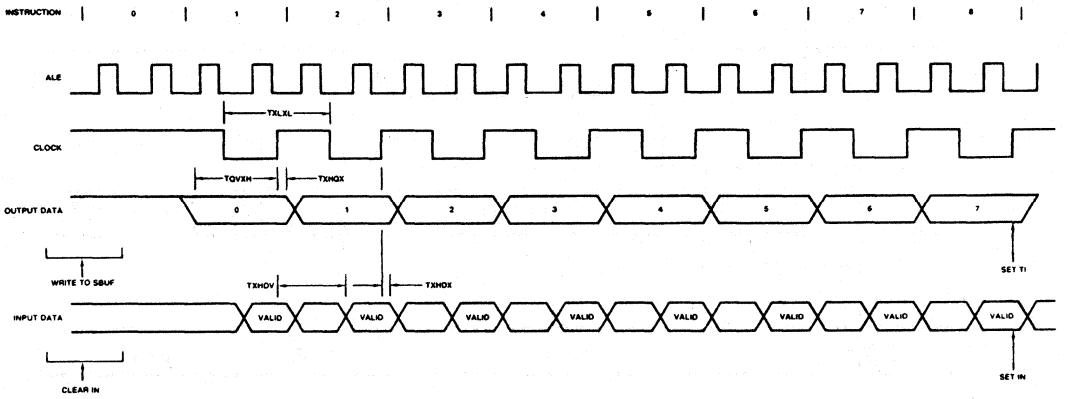
WF008760

SERIAL PORT TIMING — SHIFT REGISTER MODE

Parameters	Description	12MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Set-up to Clock Rising Edge	700		10TCLCL–133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL–117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL–133	ns

Note: Test Conditions: (T_A = 0 to +70°C, V_{CC} = +5V ±10%, V_{SS} = 0V, Load Capacitance = 80pF).

SHIFT REGISTER TIMING WAVEFORMS



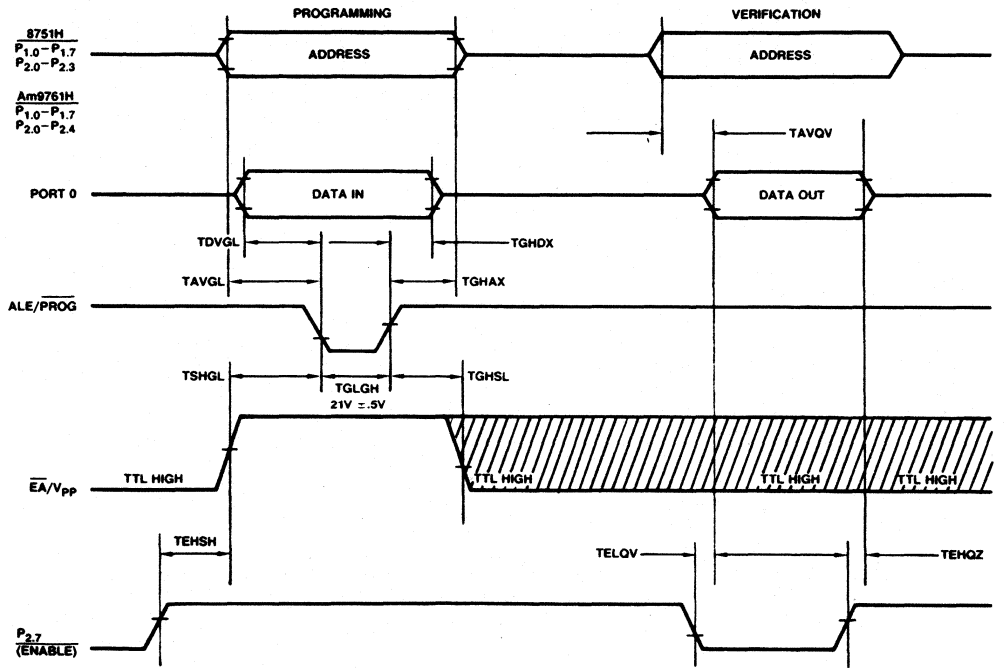
WF008720

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

(T_A = 21 to +27°C, V_{CC} = +5V ±10%, V_{SS} = 0V)

Parameters	Description	Min	Max	Units
V_{PP}	Programming Supply Voltage	20.5	21.5	V
I_{PP}	Programming Supply Current		30	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Set-up to $\overline{PROG}$	48TCLCL		
TGHAX	Address Hold After $\overline{PROG}$	48TCLCL		
TDVGL	Data Set-up to $\overline{PROG}$	48TCLCL		
TGHDX	Data Hold After $\overline{PROG}$	48TCLCL		
TEHSH	$P_{2.7}$ ($\overline{ENABLE}$) High to V_{PP}	48TCLCL		
TSHGL	V_{PP} Set-up to $\overline{PROG}$	10		µsec
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		µsec
TGLGH	$\overline{PROG}$ Width	45	55	msec
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ to Data Valid		48TCLCL	
TEHQZ	Data Float After $\overline{ENABLE}$	0	48TCLCL	

EPROM PROGRAMMING AND VERIFICATION WAVEFORMS



WF008710

For Programming conditions, see Figures 2a, 2b, and 2c.
For Verification conditions, see Figures 3a, 3b, 4a, and 4b.
For Security Bit Programming, see Figure 5.

Z8000 Family Index

Z8001/Z8002

16-Bit Microprocessors

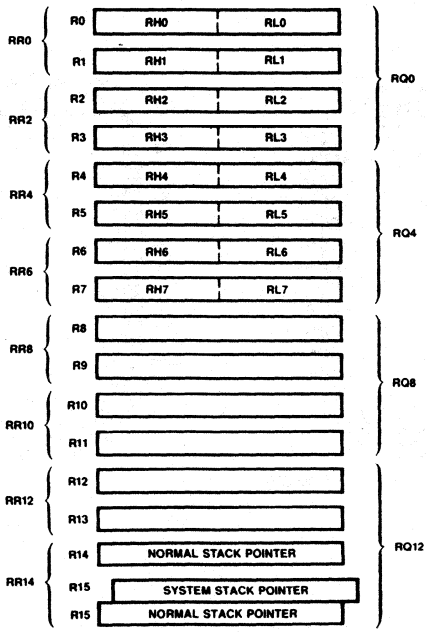
DISTINCTIVE CHARACTERISTICS

- **4 MHz CPU Clock**
  High throughput with low system clock rate for easier system design
- **Powerful General Register Architecture**
  16 general registers provide high throughput in all types of applications.
- **Wide Variety of Data Types**
  Instructions operate on bits, bytes, 16- and 32-bit words for efficient programming of a wide variety of functions.
- **Partitioned for Operating System Protection**
  Hardware bit protects privileged instructions from execution except by operating system.
- **Supports 3 Types of Interrupts**
  Separate pins provided for vectored, non-vectored and non-maskable interrupts
- **Two Compatible CPUs**
  Compact 40-pin Z8002 supports 64kB memory; larger 48-pin Z8001 supports 8MB memory.

GENERAL DESCRIPTION

The Z8001* is a general-purpose 16-bit CPU belonging to the Z8000 family of microprocessors. Its architecture is centered around sixteen 16-bit general registers. The CPU deals with 23-bit address spaces and hence can address directly 8MB of memory. The 23-bit address consists of two components: 7-bit segment number and 16-bit offset. Facilities are provided to maintain three distinct address spaces — code, data and stack. The Z8001 implements a powerful instruction set with flexible addressing modes. These instructions operate on several data types — bit, byte, word (16-bit), long word (32-bit), byte string and word string. The CPU can execute instructions in one of two modes — System and Normal. Sometimes these modes are also known as Privileged and Non-Privileged, respectively. The CPU also contains an on-chip memory refresh facility. The Z8001 is software compatible with the Z8002 microprocessor. The Z8001 is fabricated using silicon-gate N-MOS technology and is packaged in a 48-pin DIP. The Z8001 requires a single +5 power supply and a single phase clock for its operation.

GENERAL REGISTERS



AF002630

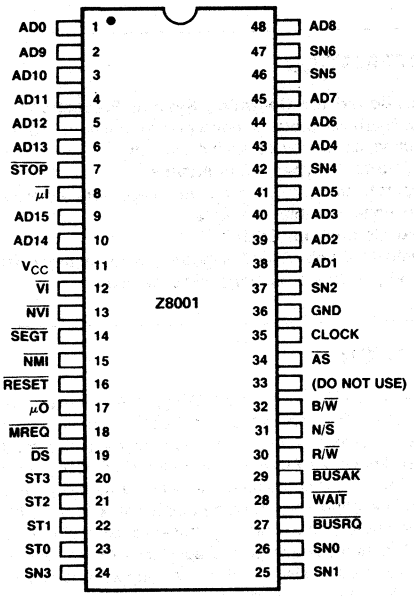
*Z8000 is a trademark of Zilog, Inc.

00971B

Refer to page 7-1 for Essential Information on Military Devices

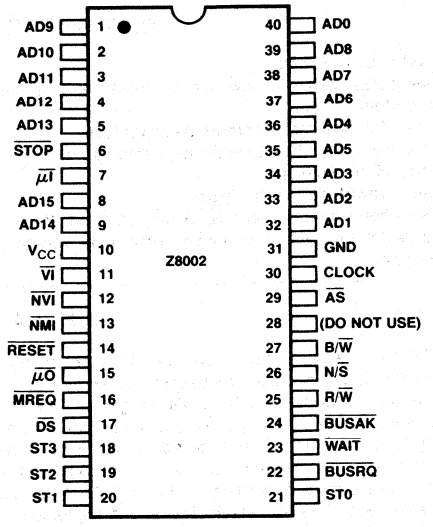
CONNECTION DIAGRAM
Top View

D-48-1, P-48-1



CD005261

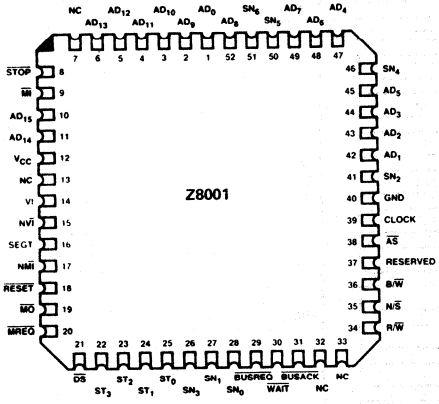
D-40-1, P-40-1



CD005271

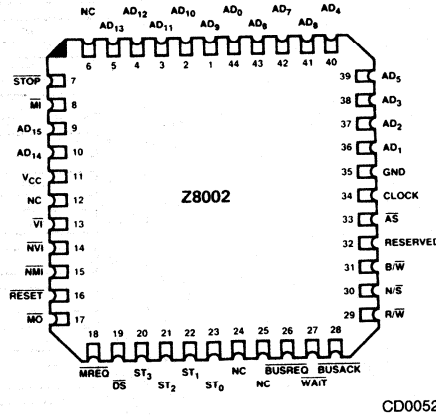
Note: Pin 1 is marked for orientation

L-52-1



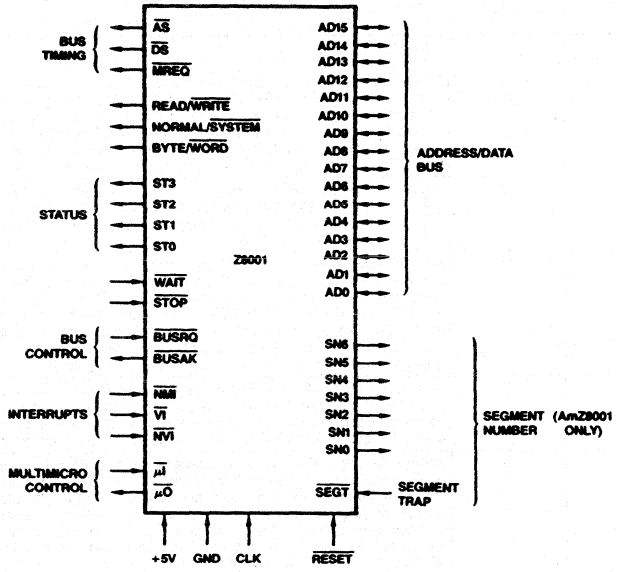
CD005281

L-44-1



CD005291

LOGIC SYMBOL



LS001271

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Z8001/Z8002

Device

Speed
Blank = 4MHz
A = 6MHz
A-8 = 8MHz

Temperature
C – Commercial (0°C to 70°C)
I – Industrial (–40°C to 85°C)

Package
P – Plastic Dip
D – Ceramic Dip
L – Ceramic LCC

Valid Combinations	
Z8001 Z8001A Z8001A-8	DC, PC, LC, DI, LI
Z8002 Z8002A Z8002A-8	DC, PC, LC, DI, LI

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

5

Z8001 PIN DESCRIPTION

Pin No.	Name	I/O	Description
11	V_{CC}		+5V Power Supply.
36	V_{SS}		Ground.
1, 38-40, 43, 41, 44, 45, 48, 2-6, 10, 9	AD0-AD15	I/O	Bidirectional, 3-State. Address/Data Bus. This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 is most significant. The $\overline{AS}$ output and $\overline{DS}$ output will indicate whether the bus is used for address offset or data. The status output lines ST0-ST3 will indicate the type of transaction: memory or I/O.
34	$\overline{AS}$	O	3-State. Address Strobe. LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the $\overline{AS}$ output (see timing diagram). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.
19	$\overline{DS}$	O	3-State. Data Strobe. LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/$\overline{W}$ output indicates the direction of data transfer – read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus where $\overline{DS}$ goes LOW. A LOW-to-HIGH transition on the $\overline{DS}$ output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the $\overline{DS}$ output indicates that data is set up on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the $\overline{DS}$ output (see timing diagram).
30	R/$\overline{W}$	O	3-State. Read/Write. This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU, and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as $\overline{AS}$ going LOW and remains stable for the duration of the whole transaction (see timing diagram).
32	B/$\overline{W}$	O	3-State. Byte/Word. This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as $\overline{AS}$ going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory, the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte, and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port, and B/$\overline{W}$ determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.
23-20	ST0-ST3	O	3-State. Status. These four outputs contain information regarding the current transaction in a coded form. The status line codes are shown in the table on the following page.
28	$\overline{WAIT}$	I	Wait. LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer, and hence, the current transaction should be stretched. The $\overline{WAIT}$ input is sampled by the CPU at certain instances during the transaction (see timing diagram). If $\overline{WAIT}$ input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the $\overline{WAIT}$ input is HIGH at the sampling instant.
31	N/$\overline{S}$	O	3-State. Normal/System Mode. HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the processor status information section of this document.
18	$\overline{MREQ}$	O	3-State. Memory Request. LOW on this output indicates that a CPU transaction with memory is taking place.
27	$\overline{BUSRQ}$	I	Bus Request. LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The $\overline{BUSRQ}$ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating $\overline{BUSAK}$ output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The $\overline{BUSRQ}$ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the $\overline{BUSAK}$ output LOW. After completing the transactions, the device must disable the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ into the high impedance state and stop driving the $\overline{BUSRQ}$ input LOW. The CPU will make $\overline{BUSAK}$ output HIGH sometime later and take back the bus control.
29	$\overline{BUSAK}$	O	Bus Acknowledge. LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.
15	$\overline{NMI}$	I	Non-Maskable Interrupt. A HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the Non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the $\overline{NMI}$ can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.
12	$\overline{VI}$	I	Vectored Interrupt. LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The NVIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{VI}$ input can be driven LOW anytime and should be held LOW until acknowledged.

00971B

Refer to page 7-1 for Essential Information on Military Devices

Z8001 PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
13	$\overline{NVI}$	I	Non-Vectored Interrupt. LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{NVI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
8	$\overline{\mu I}$	I	Micro-In. This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
17	$\overline{\mu O}$	O	Micro-Out. This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
16	$\overline{RESET}$	I	Reset. LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.
35	CLK	I	Clock. All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.
7	$\overline{STOP}$	I	Stop. This active LOW input facilitates one instruction at a time operation. See the section on single stepping.
26, 25, 37, 24, 42, 46, 47	SN0-SN6	O	3-State. Segment Number. These seven outputs contain the segment number part of a memory address. A HIGH on the output corresponds to 1 and a LOW corresponds to 0. SN0 is the least significant bit position and SN6 is the most significant bit position.
14	$\overline{SEGT}$	I	Segment Trap. LOW on this input constitutes a segment trap request. If the line is driven LOW, the CPU will respond with the Segment Trap Acknowledge code on the Status lines and commence a trap sequence. The $\overline{SEGT}$ input may be driven LOW at any time and is customarily held LOW until acknowledged. This input has priority over the interrupts.

Status Line Codes

ST3	ST2	ST1	ST0	
L	L	L	L	Internal Operation
L	L	L	H	Memory Refresh
L	L	H	L	Normal I/O Transaction
L	L	H	H	Special I/O Transaction
L	H	L	L	Segment Trap Acknowledge
L	H	L	H	Non-Maskable Interrupt Acknowledge
L	H	H	L	Non-Vectored Interrupt Acknowledge
L	H	H	H	Vectored Interrupt Acknowledge
H	L	L	L	Memory Transaction for Operand
H	L	L	H	Memory Transaction for Stack
H	L	H	L	Reserved
H	L	H	H	Reserved
H	H	L	L	Memory Transaction for Instruction Fetch (Subsequent Word)
H	H	L	H	Memory Transaction for Instruction Fetch (First Word)
H	H	H	L	Reserved
H	H	H	H	Reserved

Z8002 PIN DESCRIPTION

Pin No.	Name	I/O	Description
10	V_{CC}		+5V Power Supply.
31	V_{SS}		Ground.
40, 32-39, 1-5, 9, 8	AD0-AD15	I/O	Bidirectional, 3-State. Address/Data Bus. This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 is most significant. The $\overline{AS}$ output and $\overline{DS}$ output will indicate whether the bus is used for address offset or data. The status output lines ST0-ST3 will indicate the type of transaction: memory or I/O.
29	$\overline{AS}$	O	3-State. Address Strobe. LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the $\overline{AS}$ output (see timing diagram). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.
17	$\overline{DS}$	O	3-State. Data Strobe. LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/$\overline{W}$ output indicates the direction of data transfer – read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus where $\overline{DS}$ goes LOW. A LOW-to-HIGH transition on the $\overline{DS}$ output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the $\overline{DS}$ output indicates that data is set up on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the $\overline{DS}$ output (see timing diagram).
25	R/$\overline{W}$	O	3-State. Read/Write. This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU, and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as $\overline{AS}$ going LOW and remains stable for the duration of the whole transaction (see timing diagram).
27	B/$\overline{W}$	O	3-State. Byte/Word. This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as $\overline{AS}$ going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory, the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte, and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port, and B/$\overline{W}$ determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.
21-18	ST0-ST3	O	3-State. Status. These four outputs contain information regarding the current transaction in a coded form (see table on previous page).
23	$\overline{WAIT}$	I	Wait. LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer, and hence, the current transaction should be stretched. The $\overline{WAIT}$ input is sampled by the CPU at certain instances during the transaction (see timing diagram). If $\overline{WAIT}$ input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the $\overline{WAIT}$ input is HIGH at the sampling instant.
26	N/$\overline{S}$	O	3-State. Normal/System Mode. HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the processor status information section of this document.
16	$\overline{MREQ}$	O	3-State. Memory Request. LOW on this output indicates that a CPU transaction with memory is taking place.
22	$\overline{BUSRQ}$	I	Bus Request. LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The $\overline{BUSRQ}$ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating $\overline{BUSAK}$ output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The $\overline{BUSRQ}$ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the $\overline{BUSAK}$ output LOW. After completing the transactions, the device must disable the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ into the high impedance state and stop driving the $\overline{BUSRQ}$ input LOW. The CPU will make $\overline{BUSAK}$ output HIGH sometime later and take back the bus control.
24	$\overline{BUSAK}$	O	Bus Acknowledge. LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.
13	$\overline{NMI}$	I	Non-Maskable Interrupt. A HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the Non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the $\overline{NMI}$ can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.
11	$\overline{VI}$	I	Vectored Interrupt. LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The VIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{VI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
12	$\overline{NVI}$	I	Non-Vectored Interrupt. LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{NVI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
7	$\overline{\mu I}$	I	Micro-In. This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
15	$\overline{\mu O}$	O	Micro-Out. This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
14	$\overline{RESET}$	I	Reset. LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.
30	CLK	I	Clock. All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.
6	$\overline{STOP}$	I	Stop. This active LOW input facilitates one instruction at a time operation. See the section on single stepping.

DETAILED DESCRIPTION

The following is a description of the Z8001 and Z8002 CPUs.

General Purpose Registers

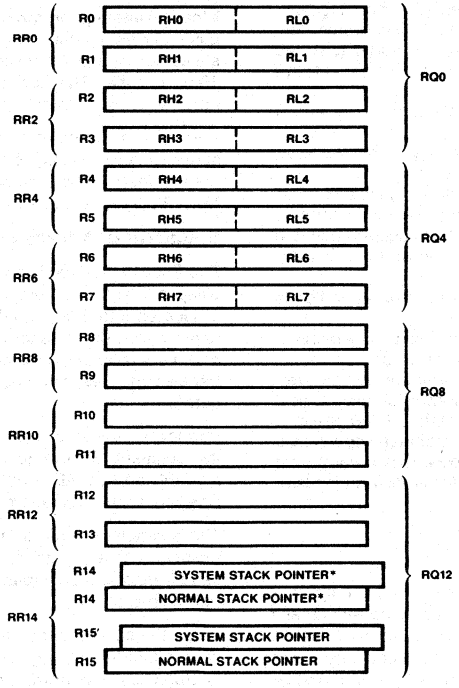
The CPU is organized around sixteen 16-bit general purpose registers R0 through R15 as shown in Figure 1. For byte operations, the first eight registers (R0 through R7) can also be addressed as sixteen 8-bit registers designated as RL0, RH0 and so on to RL7 and RH7. The sixteen registers can also be grouped in pairs RR0, RR2 and so on to RR14 to form eight long word (32-bit) registers. Similarly, the sixteen registers can be grouped in quadruples RQ0, RQ4, RQ8 and RQ12 to form four 64-bit registers.

Stack Pointer (Z8001)

The Z8001 architecture allows stacks to be maintained in memory. Any general-purpose register pair except RR0 can be used as a stack pointer in stack manipulating instructions, such as PUSH and POP. The designated register pair holds a 23-bit segmented address. Certain instructions (such as subroutine call and return) make implicit use of the register pair RR14 as the stack pointer. Two implicit stacks are allowed — normal stack using RR14 as the stack pointer and system stack using RR14′ as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, RR14 is active, and if the CPU is in System Mode, RR14′ will be used instead of RR14. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.

Stack Pointer (Z8002)

The Z8002 architecture allows stacks to be maintained in the memory. Any general purpose register except R0 can be used as a stack pointer in stack manipulating instructions, such as PUSH and POP. However, certain instructions such as subroutine call and return make implicit use of the register R15 as the stack pointer. Two implicit stacks are maintained — normal stack using R15 as the stack pointer and system stack using R15′ as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, R15 is active, and if the CPU is in System Mode, R15′ will be used instead of R15. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.



*AmZ8001 only

AF002640

Figure 1. CPU General Registers

5

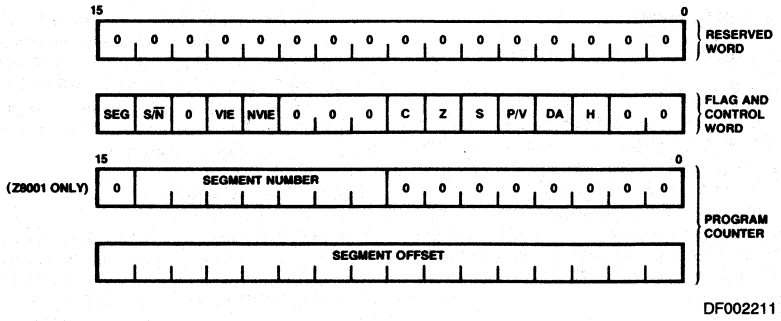


Figure 2. CPU Processor Status

Processor Status

The CPU status consists of the 16-bit flag and control word (FCW) register, and the 16- or 23-bit program counter (see Figure 2). A reserved word is also included for future expansion. The following is a brief description of the FCW bits.

SEG: Segmented/Non-Segmented Bit. Indicates whether the Z8001 is running in segmented or non-segmented mode. 1 indicates segmented; 0 indicates non-segmented. See the section on non-segmented mode, elsewhere in this document. This bit is always 0 in the Z8002.

S/$\overline{N}$: System/Normal – 1 indicates System Mode, and 0 indicates Normal Mode.

VIE: Vectored Interrupt Enable – 1 indicates that Vectored Interrupt requests will be honored.

NVIE: Non-Vectored Interrupt Enable – 1 indicates that Non-vectored interrupt requests will be honored.

C: Carry – 1 indicates that a carry has occurred from the most significant bit position when performing arithmetic operations.

Z: Zero – 1 indicates that the result of an operation is zero.

S: Sign – 1 indicates that the result of an operation is negative; i.e., most significant bit is one.

P/V: Parity/Overflow – 1 indicates that there was an overflow during arithmetic operations. For byte logical operations, this bit indicates parity of the result.

DA: Decimal Adjust – Records byte arithmetic operations.

H: Half Carry – 1 indicates that there was a carry from the most significant bit of the lower digit during byte arithmetic.

Data Types

The CPU instructions operate on bits, digits (4 bits), bytes (8 bits), words (16 bits), long words (32 bits), byte strings and word strings type operands. Bits can be set, reset or tested. Digits are used to facilitate BCD arithmetic operations. Bytes are used for characters and small integers. Words are used for integer values and addresses while long words are used for large integer values and addresses. All operands except strings can reside either in memory or general registers. Strings can reside in memory only.

Interrupt and Trap Structure

Interrupt is defined as an external asynchronous event requiring program interruption. For example, interruption is caused by a peripheral needing service. Traps are synchronous events resulting from execution of certain instructions under some defined circumstances. Both interrupts and traps are handled in a similar manner.

The CPU supports three types of interrupts in order of descending priority — non-maskable, vectored and non-vectored. The vectored and non-vectored interrupts can be disabled by appropriate control bits in the FCW. The CPU has four traps — system call, segment trap, unimplemented opcode and privileged instruction. The traps have higher priority than interrupts.

When an interrupt or trap occurs, the current program status is automatically pushed onto the system stack. The program status consists of processor status (i.e., PC and FCW) plus a 16-bit identifier. The identifier contains the reason, source and other coded information relating to the interrupt or trap.

After saving the current program status, the new processor status is automatically loaded from the new program status area located in the memory. This area is designated by the New Program Status Area Pointer (NPSAP) register.

Segmented Addressing (Z8001 Only)

The Z8001 can directly address up to 8MB of memory space, using a 23-bit segmented address. The memory space is divided up into 128 segments, each up to 64kB in size. The upper seven bits of address designate the segment number and are available on the SN0-SN6 outputs during a memory transaction. See the section on memory transactions for details.

The lower sixteen bits of address designate an offset within the segment, relative to the start of the segment, and are available on AD0-AD15 during part of the memory transaction. See the section on memory transactions for details.

The segmented address may be stored as a long word in memory or in a register pair. The segment number and offset can be manipulated separately or together by suitable use of the instruction set.

When the segmented address is contained in code space, a short offset format may be adopted. The segmented address is stored as one word, seven bits of segment number and eight bits of offset. Figure 3 shows the format for segmented addresses.

Addressing Modes

Information contained in the CPU instructions consists of the operation to be performed, the operand type and the location of the operands. Operand locations are designated by general register addresses, memory addresses or I/O addresses. The addressing mode of a given instruction defines the address space referenced and the method to compute the operand address. Addressing modes are explicitly specified or implied in an instruction. Figure 4 illustrates the eight explicit address-

ing modes: Register (R), Immediate (IM), Indirect Register (IR), Direct Address (DA), Indexed (X), Relative Address (RA), Base Address (BA) and Base Indexed (BX).

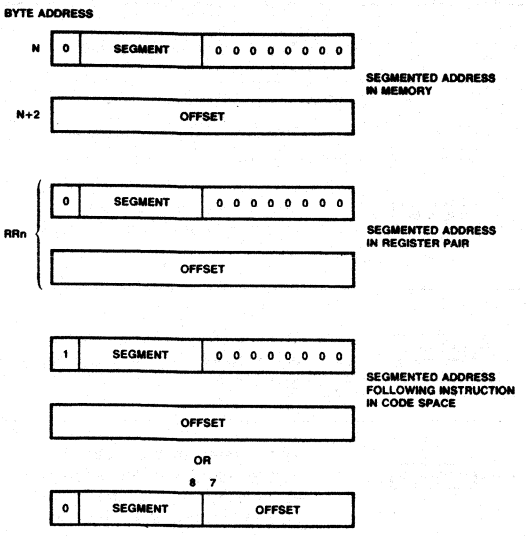


Figure 3. Segmented Address Formats

When an effective segmented address is being computed according to the designated addressing mode, the segment number is not affected by any carry from the 16-bit offset.

Non-Segmented Mode on the Z8001

The Z8001 can execute code designed to run on the non-segmented Z8002. This is achieved by changing the mode of execution of the Z8001 from segmented to non-segmented by writing a 0 to the SEG bit in the FCW. (See the section on processor status.) The change to non-segmented mode sets up a suitable environment for running non-segmented code. However, the environment only exists within the code segment that caused the change of mode from segmented to non-segmented.

SN0-SN6 will continue to indicate the code segment until a reset, interruption or return to segmented mode is encountered.

The effects of the non-segmented mode of operation on the Z8001 are described below:

a) The Z8001 will interpret instruction length like it was a non-segmented Z8002.

b) The Z8001 will implement address computation in an identical manner to the Z8002.

Other CPU functions, such as interrupt and trap handling, reset and stack pointer manipulation, are unaltered. These functions are characterized by the type of CPU, not by the state of the SEG bit in the FCW.

Input/Output

A set of I/O instructions are provided to accomplish byte or word transfers between the CPU and I/O devices. I/O devices are addressed using 16-bit I/O port addresses, and I/O address space is not a part of the memory address space. Two types of I/O instructions are provided, each with its own 16-bit address space. I/O instructions include a comprehensive set of In, Out and Block transfers.

CPU Timing

The CPU accomplishes instruction execution by stepping through a pre-determined sequence of machine cycles, such as memory read, memory write, etc. Each machine cycle requires between three and ten clock cycles. But Requests by DMA devices are granted at machine cycle boundaries. No machine cycle is longer than ten clock cycles, thus assuring fast response to a Bus Request (assuming no extra wait states). The start of a machine cycle is always marked by a LOW pulse on the $\overline{AS}$ output. The status output lines ST0-ST3 indicate the nature of the current cycle in a coded form.

Status Line Codes

Status line coding was listed in the table shown under Pin Description. The following is a detailed description of the status codes.

Internal Operation:

This status code indicates that the CPU is going through a machine cycle for its internal operation. Figure 5 depicts an internal operation cycle. It consists of three clock periods identified as T1, T2 and T3. The $\overline{AS}$ output will be activated with a LOW pulse by the Z8001 to mark the start of a machine cycle. The ST0-ST3 will reflect the code for the internal operation. The $\overline{MREQ}$, $\overline{DS}$ and R/$\overline{W}$ outputs will be HIGH. The N/$\overline{S}$ and SN0-SN6 outputs will remain at the same level as in the previous machine cycle. The CPU will ignore the $\overline{WAIT}$ input during the internal operation cycle. The CPU will drive the AD0-AD15 bus with unspecified information during T1. However, the bus will go into high-impedance during T2 and remain in that state for the remainder of the cycle. The B/$\overline{W}$ output is also activated by the CPU with unspecified information.

Memory Refresh:

This status code indicates that CPU is accessing the memory to refresh. The refresh cycle consists of three clock periods as depicted in Figure 6. The CPU will activate the $\overline{AS}$ output with a LOW pulse to mark the beginning of a machine cycle, and ST0-ST3 outputs will reflect the refresh cycle code. The least significant 9 lines of the AD0-AD15 bus contain the refresh address. Because the memory is word organized, the AD0 will always be LOW. The most significant 7 bus lines are not specified. The $\overline{DS}$ output will remain HIGH for the entire cycle, while R/$\overline{W}$, B/$\overline{W}$, SN0-SN6 and N/$\overline{S}$ outputs will remain at the same level as in the machine cycle prior to refresh. The AD0-AD15 bus will go into high-impedance state during T2 period and remain there for the remainder of the cycle. The CPU will activate the $\overline{MREQ}$ output LOW during the refresh cycle. It should be noted that $\overline{WAIT}$ input is ignored by the CPU for refresh operations.

I/O Transactions:

There are two status line codes used for I/O transaction cycles. The CPU provides two separate I/O spaces and two types of instructions called Normal I/O and Special I/O. Each I/O space is addressed by a 16-bit address called port address. The timing for both types of I/O transactions is essentially identical. A typical I/O cycle consists of four clock periods T1, T2, TWA and T3 as shown in Figure 7. The TWA is the wait state; insertion of one wait state for an I/O cycle is always automatic. Additional wait cycles can be inserted by LOW on the $\overline{WAIT}$ input. The $\overline{WAIT}$ input is sampled during every TW state. If this input is LOW, one more wait state will be inserted. Insertion of wait states continues until $\overline{WAIT}$ input is HIGH. T3 state will follow the last wait state to complete the I/O cycle.

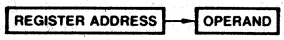
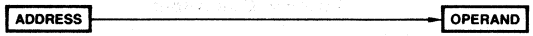
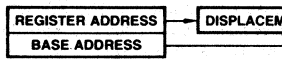
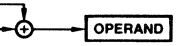
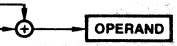
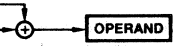
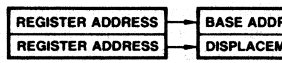
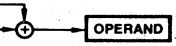
Mode	Operand Addressing			Operand Value
	In the Instruction	In a Register	In Memory	
Register				The content of the register.
Immediate				In the instruction
Indirect Register				The content of the location whose address is in the register.
Direct Address				The content of the locations whose address is in the instruction.
Index				The content of the location whose address is the address in the instruction, offset by the content of the working register.
Relative Address				The content of the locations whose address is the content of the program counter, offset by the displacement in the instruction.
Base Address				The content of the location whose address is the address in the register, offset by the displacement in the instruction.
Base Index				The content of the location whose address is the address in the register, offset by the displacement in the register.

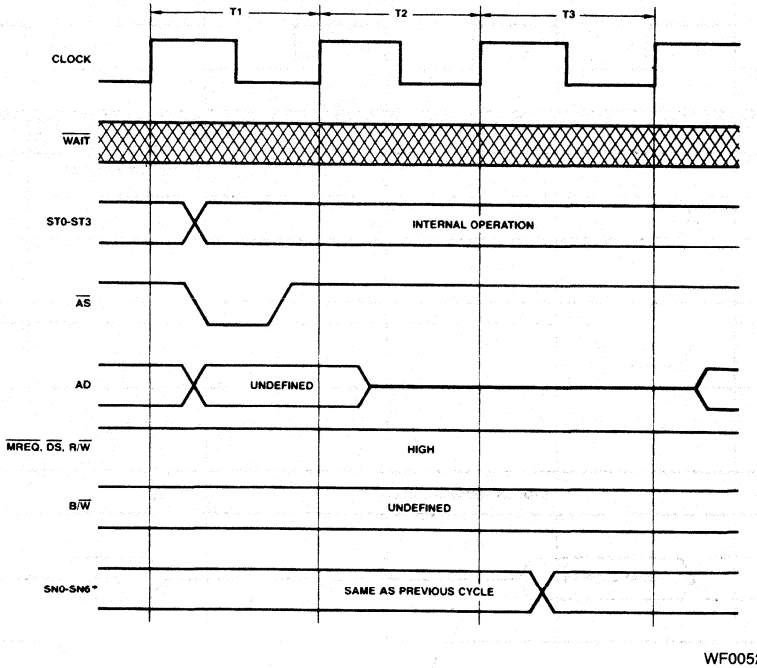
Figure 4. Addressing Modes

During I/O cycles the ST0-ST3 outputs will reflect the appropriate code depending on the type of instruction being executed (Normal I/O or Special I/O). $\overline{AS}$ output will be pulsed LOW to mark the beginning of the cycle. The CPU drives the AD0-AD15 bus with the 16-bit port address specified by the current instruction. The N/$\overline{S}$ output will be LOW indicating that the CPU is operating in the system mode. It should be recalled that the N/$\overline{S}$ output is derived from the appropriate bit in the FCW register. All I/O instructions are privileged instructions and will be allowed to execute only if the FCW specifies system mode operation. The $\overline{MREQ}$ output will be HIGH. The I/O instructions provide both word or byte transactions. The B/$\overline{W}$ output will be HIGH or LOW depending whether the instruction specifies a byte or word transfer. The SN0-SN6 output will remain at the same level as in the machine cycle prior to the I/O cycle.

Two kinds of I/O transfers should be considered: Data In means reading from the device, and Data Out means writing into the device. For In operations, the R/$\overline{W}$ output will be HIGH. The AD0-AD15 bus will go into high-impedance state during T2. During byte input instructions, the CPU reads either the even or odd half of the Data Bus dependent upon the port address. If the port address is even, the most significant half of the Data Bus is read. If the port address is odd, the least significant half of the Data Bus is read. During word input instructions, the CPU reads all 16 bits of the Data Bus. The CPU will drive the $\overline{DS}$ output LOW to signal to the device that data can be gated onto the bus. The CPU will accept the data during T3, and $\overline{DS}$ output will go HIGH signaling the end of an I/O transaction.

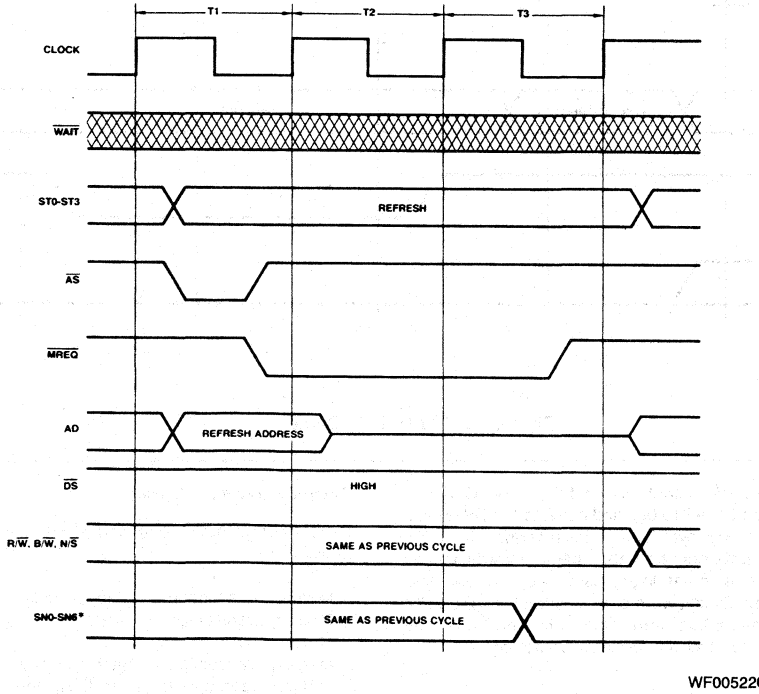
00971B

Refer to page 7-1 for Essential Information on Military Devices



*Z8001 only

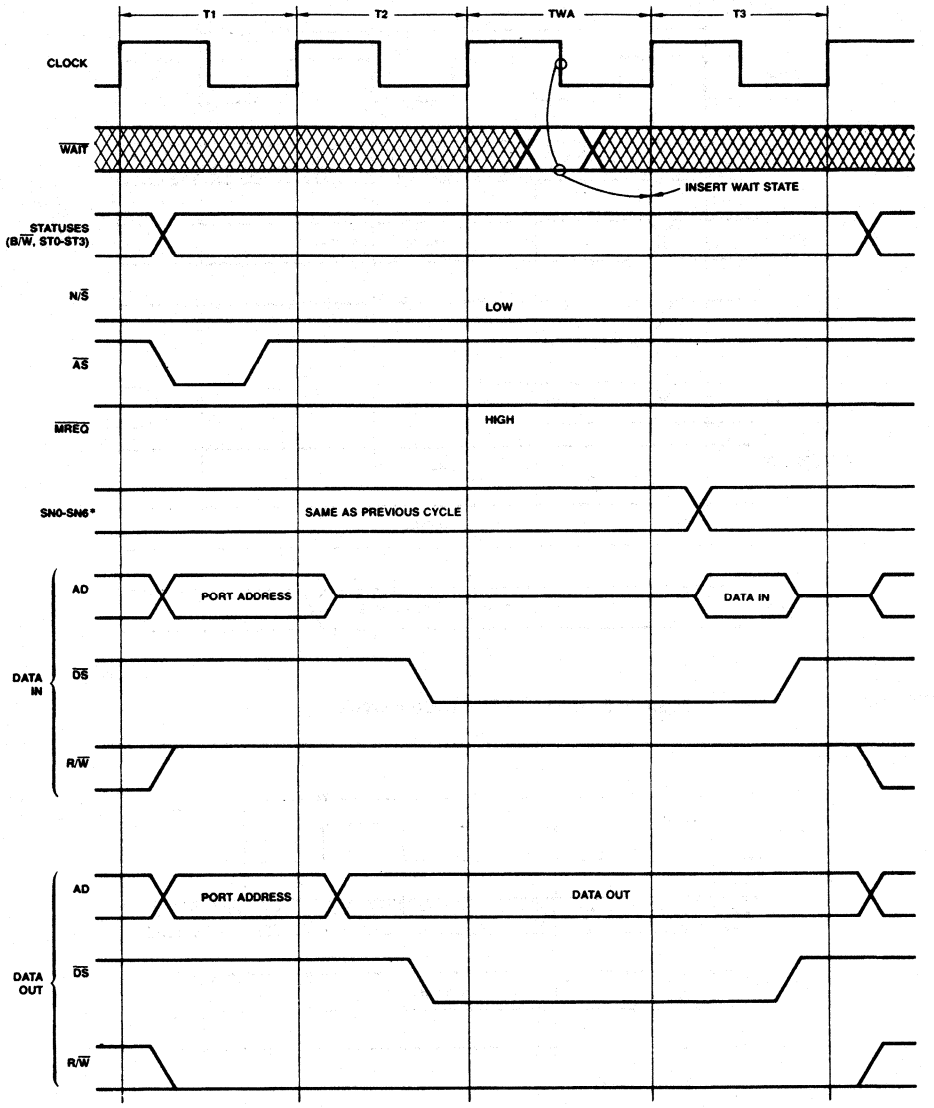
Figure 5. Internal Operation Cycle



*Z8001 only

Figure 6. Refresh Cycle

5



*Z8001 only

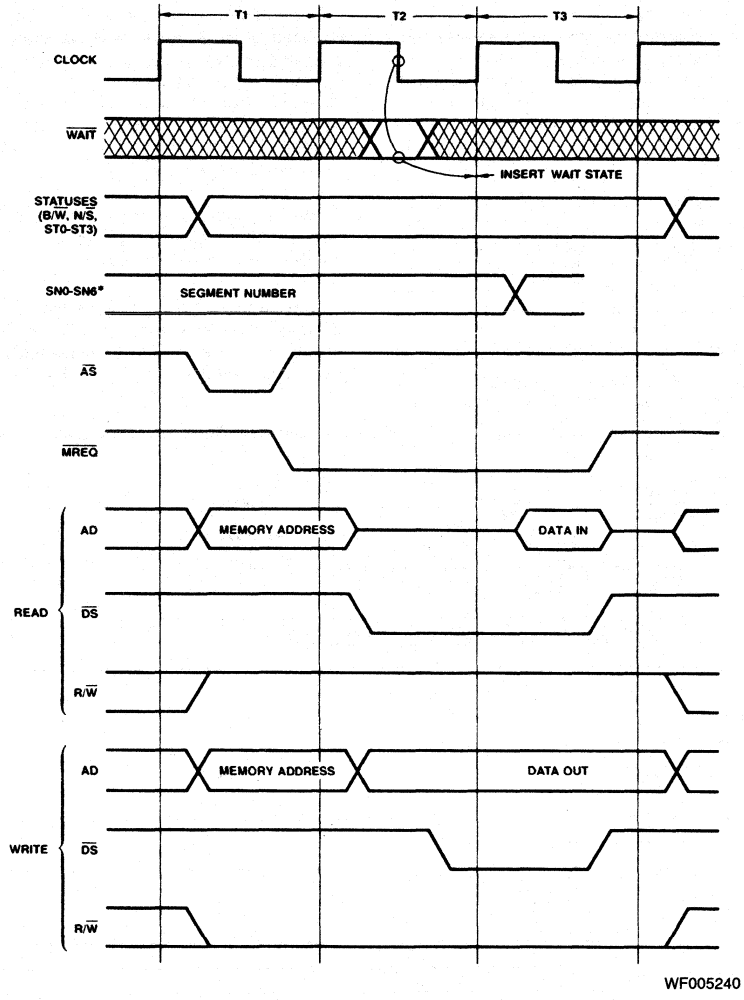
Figure 7. Z8001 I/O Cycle

For Data Out, the R/$\overline{W}$ output will be LOW. The CPU will provide data on the AD0-AD15 bus and activate the $\overline{DS}$ output LOW during T2. During byte output instructions, the CPU duplicates the byte data onto both the high and low halves of the Data Bus, and external logic, using A0, enables the appropriate byte port. During word output instructions the CPU outputs data onto all 16 bits of the Data Bus. The $\overline{DS}$ output goes HIGH during T3 and the cycle is complete.

Memory Transactions:

There are four status line codes that indicate a memory transaction:

a) Memory transaction to read or write an operand;
b) Memory transaction to read from or write into the stack;
c) Memory transaction to fetch the first word of an instruction (sometimes called IF1); and
d) Memory transaction to fetch the subsequent word of an instruction (sometimes called IFN).



*Z8001 only

Figure 8. Memory Transactions

It can be appreciated that all the above transactions essentially fall into two categories: memory read and memory write. In the case of IF1 and IFN cycles, the memory will be read at the address supplied by the program counter. All instructions are multiples of 16-bit words. Words are always addressed by an even address. Thus IF1 and IFN cycles involve performing a memory read for words. On the other hand, a memory transaction for operand and stack operation could be a read or write. Moreover, an operand could be a word or a byte. For stack operation involving the implied stack pointer, the address will be in the appropriate stack pointer register (R15, R15', RR14 or RR14'). For operand transactions, the memory address will come from several sources depending on the instruction and the addressing mode. Memory transaction cycle timing is shown in Figure 8. It typically consists of three clock periods: T1, T2 and T3. Wait states (TW) can be inserted between T2 and T3 by activating the $\overline{WAIT}$ input LOW. The $\overline{WAIT}$ input will be sampled during T2 and during every subsequent TW. The ST0-ST3 outputs will reflect the appropriate code for the current cycle early in T1, and the $\overline{AS}$ output will be pulsed LOW to mark the beginning of the cycle. The N/$\overline{S}$ output will indicate whether the normal or system address space will be used for the current cycle. As shown in the figure, the $\overline{MREQ}$ output will go LOW during T1 to indicate a memory operation.

The segment number becomes valid on the segment lines one clock period before the state of the memory operation and remains valid until the state of T3.

Consider a read operation first. The R/$\overline{W}$ output will be HIGH. The CPU will drive the AD0-AD15 with the appropriate address early in T1. During T2, the bus will go into high-impedance state, and $\overline{DS}$ output will be activated LOW by the CPU. The data can be gated onto the bus when $\overline{DS}$ is LOW. During T1 the B/$\overline{W}$ will also be activated to indicate which byte or word will be transacted. The memory is word organized, and words

are addressed by even addresses. However, when addressing bytes, the memory address may be odd or even, an even address for the most significant byte of a word and the next odd address for the least significant byte of that word. When reading a byte from the memory, the least significant address bit can be ignored and the whole word containing the desired byte is gated onto the bus. The CPU will pick the appropriate byte automatically and will drive the $\overline{DS}$ output HIGH indicating data acceptance.

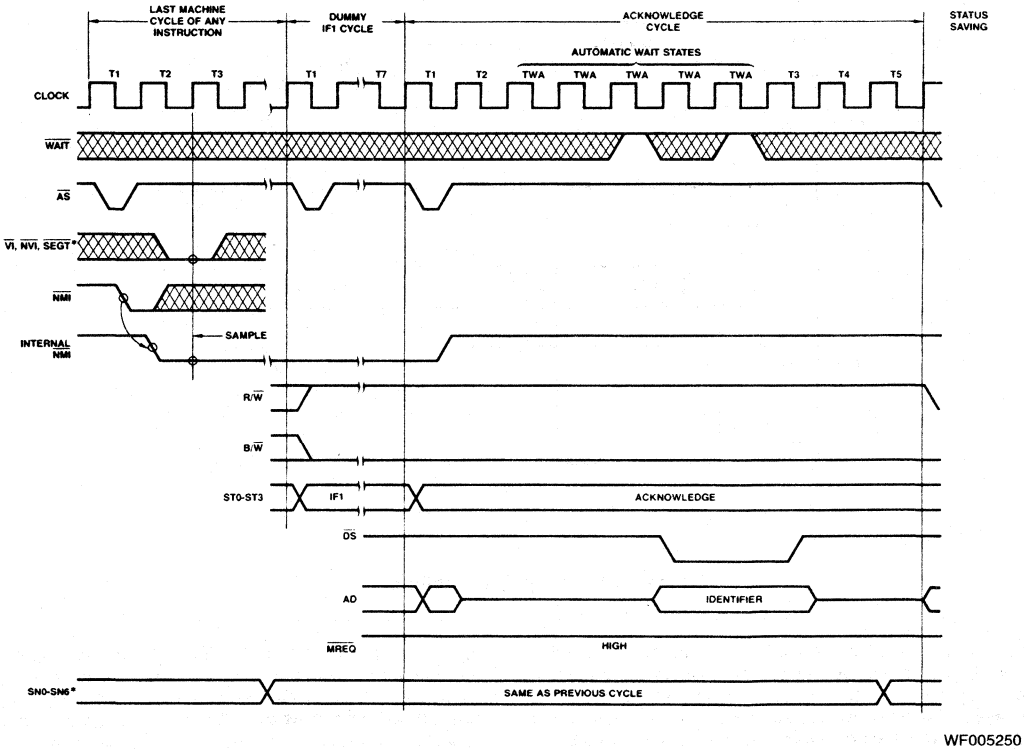
Consider the write operation next. The R/$\overline{W}$ output will be LOW. The CPU removes the address and gates out the data to be written on the bus and activates the $\overline{DS}$ output LOW during T2. If the data to be written is a byte, then the same byte will be on both halves of the bus. The $\overline{DS}$ output will go HIGH during T3, signifying completion of the cycle.

Interrupt and Segment Trap Acknowledge:

There are four status line codes devoted to interrupt and trap acknowledgement. These correspond to non-maskable, vectored and non-vectored interrupts, as well as segment trap. The Interrupt Acknowledge cycle is illustrated in Figure 9. The $\overline{NMI}$ input of the Z8001 is edge detected; i.e., a HIGH-to-LOW input level change is stored in an internal latch. Similar internal storage is not provided for the $\overline{VI}$, $\overline{NVI}$, and $\overline{SEGT}$ inputs. For $\overline{VI}$ and $\overline{NVI}$ inputs to cause an interruption, the corresponding interrupt enable bits in the FCW must be 1. For the following discussion, both the NVIE and VIE bits in the FCW are assumed to be 1.

As shown in the figure, the $\overline{VI}$, $\overline{NVI}$ and $\overline{SEGT}$ input and the internal $\overline{NMI}$ latch output are sampled during T3 of the last machine cycle of an instruction.

A LOW on these signals triggers the corresponding interrupt acknowledge sequence described below. The CPU executes a dummy IF1 cycle prior to entering the actual acknowledge cycle (see memory transactions for IF1 cycle description).



*Z8001 only

Figure 9. Interrupt Acknowledge Cycle

During this dummy IF1 cycle, the program counter is not updated; instead, the implied-system stack pointer (RR14') will be decremented. Following the dummy IF1 cycle is the actual interrupt/trap acknowledge cycle.

The interrupt acknowledge cycle typically consists of 10 clock periods: T1 through T5 and five automatic TW (wait states). As usual, the $\overline{AS}$ output will be pulsed LOW during T1 to mark the beginning of a cycle. The ST0-ST3 outputs will reflect the appropriate interrupt acknowledge code; the $\overline{MREQ}$ output will be HIGH; the N/$\overline{S}$ output remains the same as in the preceding cycle; the R/$\overline{W}$ output will be HIGH; and the B/$\overline{W}$ output will be LOW. The CPU will drive the AD0-AD15 bus with

unspecified information during T1, and the bus will go into the high-impedance state during T2. Three TWA states will automatically follow T2. The WAIT input will be sampled during the third TWA state.

If LOW, an extra TW state will be inserted and the WAIT will be sampled again during TW. Such insertion of TW states continues until the WAIT input is HIGH. After the last TW state, the DS output will go LOW and two more automatic wait states (TWA) follow. The interrupting device can gate up to a 16-bit identifier onto the bus when the DS output is LOW. The WAIT input will be sampled again during the last TWA state. If the WAIT input is LOW, one TW state will be inserted, and the WAIT will be sampled during TW. Such TW insertion continues until the WAIT input is HIGH. After completing the last TW state, T3 will be entered, and the DS output will go HIGH. The interrupting device should remove the identifier and cease driving the bus. T4 and T5 states will follow T3 to complete the cycle. Following the interrupt acknowledge cycle will be memory transaction cycles to save the status on the stack. Note that the N/S output will be automatically LOW during status saving. The SN0-SN6 outputs are undefined during the acknowledge cycle.

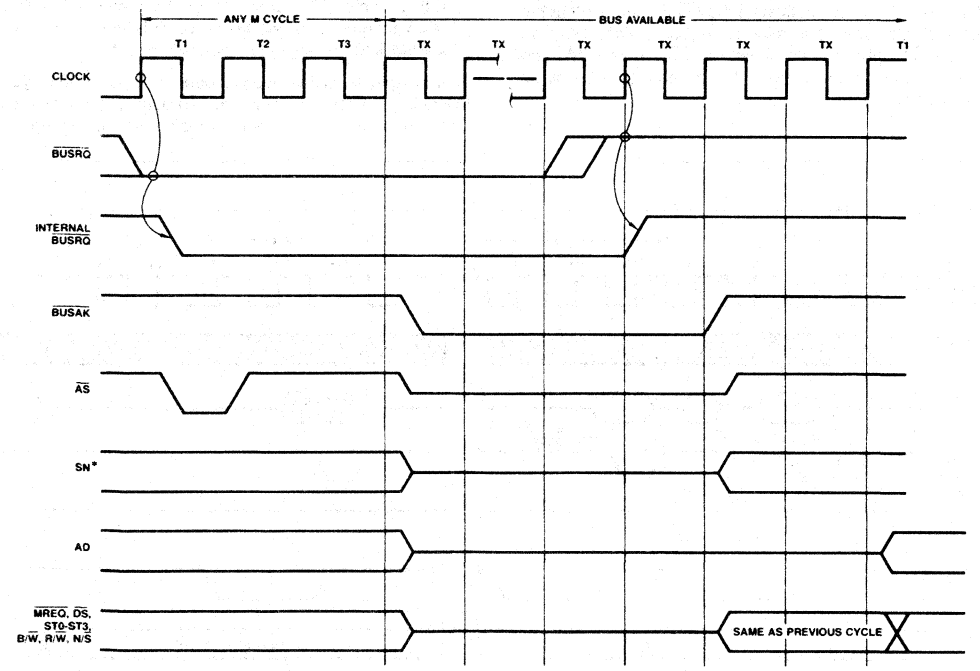
The internal NMI latch will be reset to the initial state as AS going HIGH in the interrupt acknowledge cycle. The VI, NVI and SEGT input should be kept LOW until this time also.

Status Saving Sequence:

The machine cycles following the interrupt acknowledge cycle push the old status information on the system stack in the following order: program counter, the flag and control word, and the interrupt/trap identifier. Subsequent machine cycles fetch the new program status from the new program status area and then branch to the interrupt/service routine.

Bus Request/Bus Acknowledge Timing:

A LOW on the BUSRQ input is an indication to the CPU that another device (such as DMA) is requesting control of the bus. The BUSRQ input is synchronized internally at T1 of any machine cycle. (See next paragraph for exception.) The BUSAK will go LOW after the last clock period of the machine cycle. The LOW on the BUSAK output indicates acknowledgement. When BUSAK is LOW, the following outputs will go into the high-impedance state: AD0-AD15, AS, DS, MREQ, ST0-ST3, B/W, R/W, SN0-SN6 and N/S. The BUSRQ must be held LOW until all transactions are completed. When BUSRQ goes HIGH, it is synchronized internally; the BUSAK output will go HIGH, and normal CPU operation will resume. Figure 10 illustrates the BUSRQ/BUSAK timing.



*Z8001 only

Figure 10. Bus Request/Acknowledge Cycle

It was mentioned that BUSRQ will be honored during any machine cycle with one exception. This exception is during the execution of TSET/TSETB instructions. BUSRQ will not be honored once execution of these instructions has started.

Single Stepping

The $\overline{STOP}$ input of the CPU facilitates one instruction at a time or single step operation. Figure 11 illustrates $\overline{STOP}$ input timing. The $\overline{STOP}$ input is sampled on the HIGH-to-LOW transition of the clock input that immediately precedes an IF1 cycle. If the $\overline{STOP}$ is found LOW, Z8001 introduces a memory refresh cycle after T3. Moreover, $\overline{STOP}$ input will be sampled again at T3 in the refresh cycle. If $\overline{STOP}$ is LOW, one more refresh cycle will follow the previous refresh cycle. The $\overline{STOP}$ will be sampled during T3 of the refresh cycle, also. One additional refresh cycle will be added every time $\overline{STOP}$ input is sampled LOW. After completing the last refresh cycle which will occur after $\overline{STOP}$ is HIGH, the CPU will insert two dummy states, T4 and T5, to complete the IF1 cycle and resume its normal operations for executing the instruction. See appropriate sections on memory transactions and memory refresh. It should be noted that refresh cycles will occur even if the refresh facility is disabled during single stepping.

Multimicroprocessor Facilities

The CPU is provided with hardware and software facilities to support multiple microprocessor systems. The $\overline{\mu O}$ and $\overline{\mu I}$ signals of the CPU are used in conjunction with the MBIT, $\overline{MREQ}$, MRES and MSET instructions for this purpose. The $\overline{\mu O}$ output can be activated LOW by using appropriate instruction to signal a request from the CPU for a resource. The $\overline{\mu I}$ input is tested by the CPU before activating the $\overline{\mu O}$ output. LOW at the $\overline{\mu I}$ input indicates that the resource is busy. The CPU can examine the $\overline{\mu I}$ input after activating the $\overline{\mu O}$ output LOW. The $\overline{\mu I}$ will be tested again to see if the requested resource became available.

Initialization

A LOW on the $\overline{Reset}$ input starts the CPU initialization. The initialization sequence is shown in Figure 12. Within five clock periods after the HIGH-to-LOW level change of the $\overline{Reset}$ input, the following will occur:

a) AD0-AD15 bus will be in the high-impedance state.

b) $\overline{AS}$, $\overline{DS}$, $\overline{MREQ}$, $\overline{BUSAK}$ and $\overline{\mu O}$ outputs will be HIGH.

c) ST0-ST3 outputs will be LOW.

d) Refresh will be disabled.

e) R/$\overline{W}$, B/$\overline{W}$ and N/$\overline{S}$ outputs are not affected. For a power-on reset, the state of these outputs is not specified.

f) SN0-SN6 outputs will be LOW.

After the $\overline{Reset}$ input returns HIGH and remains HIGH for three clock periods, two (three for the Z8001) 16-bit memory read operations will be performed as follows. Note that the N/$\overline{S}$ output will be LOW, and ST0-ST3 outputs will reflect IFN code.

a) The contents of the memory location 0002 (segment 0) will be read. This information will be loaded into the FCW of the CPU.
b) The contents of the memory location 0004 (segment 0) will be read. This information will be loaded into the program counter segment number.
c) (Z8001 only.) The contents of the memory location 0006 (segment 0) will be read. This information will be loaded into the program counter offset.

This completes initialization sequence, and an IF1 cycle will follow to fetch the first instruction to begin program execution. See the section on memory transactions for timing.

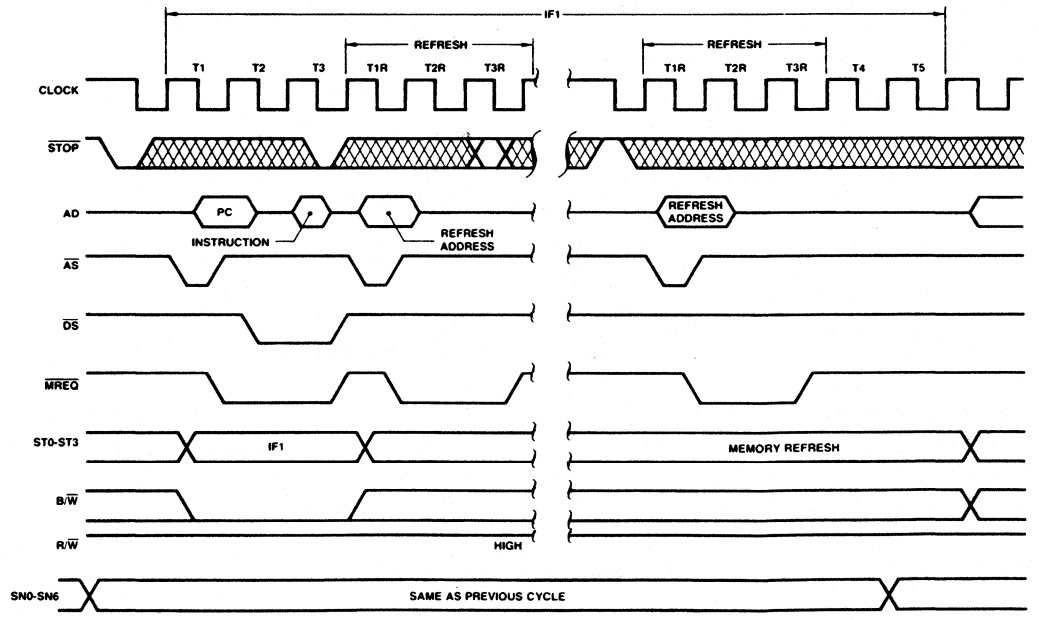
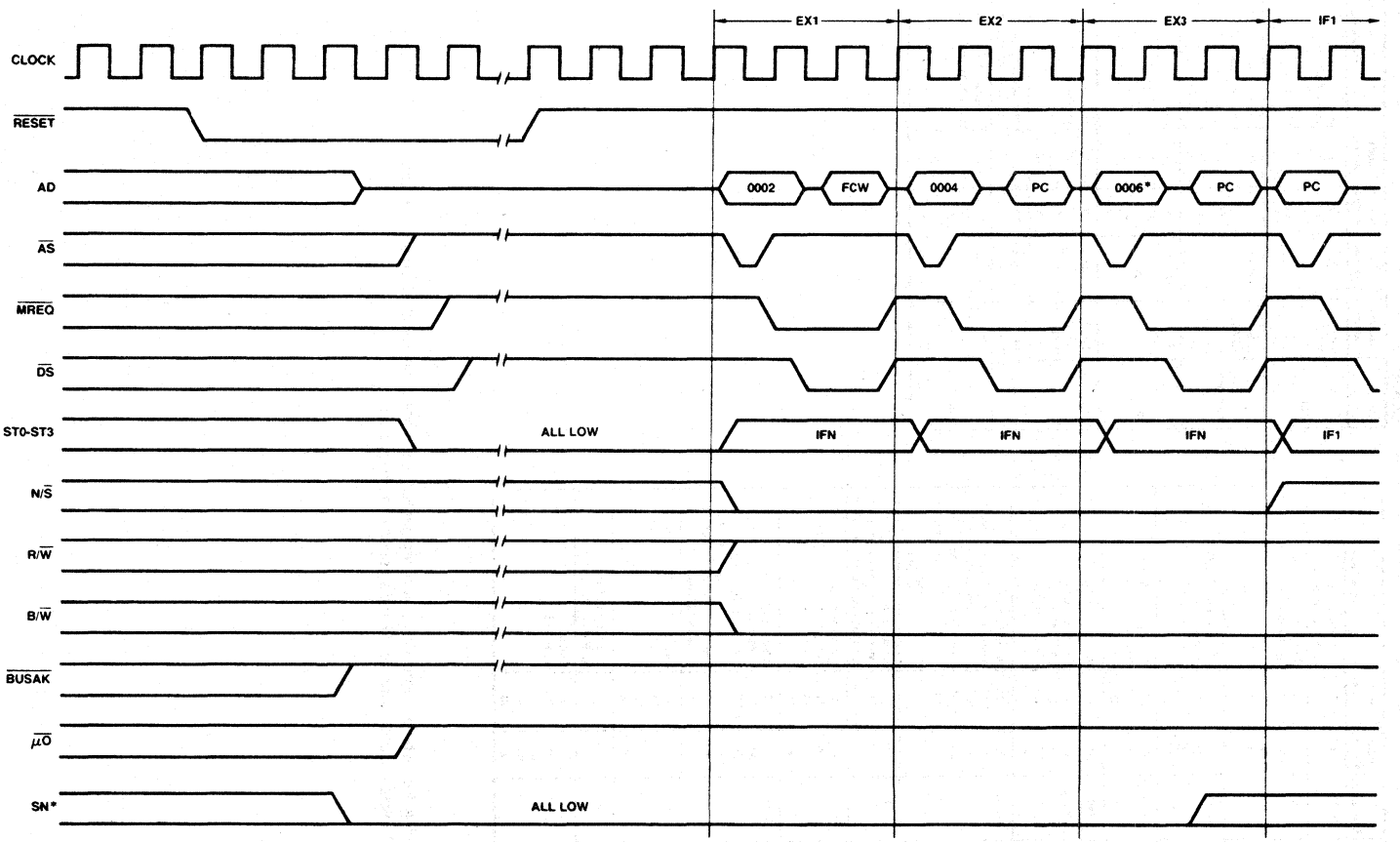


Figure 11. Single Step Timing



WF005280

*Z8001 only

Refer to page 7-1 for Essential Information on Military Devices

00971B

Z8001/2 CPU INSTRUCTION SET

ARITHMETIC

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
	ADC ADCB	R, src	R	5	Add with Carry R←R + src + carry
Add	ADD ADDB ADDL	R, src	R IM IR DA X	4 7 7 9 10	Add R←R + src
Compare	CP CPB CPL	R, src	R IM IR X	4 7 7 10	Compare with Register R – src
	CP CPB	dst, IM	IR DA X	11 14 15	Compare with Immediate dst – IM
	DAB	dst	R	5	Decimal Adjust
Decrement	DEC DECB	dst, n	R IR DA X	4 11 13 14	Decrement by n dst←dst – n (n = 1 . . . 16)
Divide	DIV DIVL	R, src	R IM IR DA X	107 107 107 108 109	Divide (signed) Word: $R_n + 1 \leftarrow R_{n,n} + 1 \div src$ $R_n \leftarrow$ remainder Long Word: $R_n + 2,n + 3$ $\leftarrow R_{n...n} + 3 \div src$ $R_{n,n} + 1$ ←remainder
Extend	EXTS EXTSB EXTSL	dst	R	11	Extend Sign Extend sign of low order half of st through high order half of dst
Increment	INC INCB	dst, n	R IR DA X	4 11 13 14	Increment by n dst←dst + n (n = 1 . . . 16)
Multiply	MULT MULTL	R, src	R IM IR DA X	70 70 70 71 72	Multiply (signed) Word: $R_{n,n} + 1 \leftarrow R_n + 1 \bullet src$ Long Word: $R_{n...n} + 3$ $\leftarrow R_n + 2, + 3 \bullet src$ *Plus seven cycles for each 1 in the multiplicand
Negate	NEG NEGB	dst	R R DA X	7 12 15 16	Negate dst←0 – dst
Subtract	SBC SBCB	R, src	R	5	Subtract with Carry R←R – src – carry
	SUB SUBB SUBL	R, src	R IM IR DA X	4 7 7 9 10	Subtract R←R – src

LOGICAL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
AND	AND ANDB	R, scr	R IM IR DA X	4 7 7 9 10	AND R←R AND src
Complement	COM COMB	dst	R IR DA X	7 12 15 16	Complement dst←NOT dst
OR	OR ORB	R, src	R IM IR DA X	4 7 7 9 10	OR R←R OR src
Test	TEST TESTB TESTL	dst	R IR DA X	7 8 11 12	TEST dst OR 0
	TCC TCCB	cc, dst	R	5	Test Condition Code Set LSB if cc is true
XOR	XOR XORB	R, src	R IM IR DA X	4 7 7 9 10	Exclusive OR R←R XOR src

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

LOAD AND EXCHANGE

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Clear	CLR CLRB	dst	R IR DA X	7 8 11 12	Clear dst←0
Exchange	EX EXB	R, src	R IR DA X	6 12 15 16	Exchange R←src
Load	LD LDB LDL	R, src	R IM IR DA X RA BA BX	3 7 7 9 10 14 14 14	Load into Register R←src
	LD LDB LDL	dst, R	IR DA X RA BA BX	8 11 12 14 14 14	Load into Memory (Store) dst←R
	LD LDB	dst, IM	R IR DA X	5 11 14 15	Load Immediate into Memory dst←IM
	LDA	R, src	DA X RA BA BX	12 13 15 15 15	Load Address R←source address
	LDAR	R, src	RA	15	Load Address Relative R←source address
	LDK	R, src	IM	5	Load Constant R←n (n = 0...15)
	LDM	R, src, n	IR DA X	11 + 3n 14 + 3n 15 + 3n	Load Multiple R←src (n consecutive words) (n = 1...16)
	LDM	dst, R, n	IR DA X	11 + 3n 14 + 3n 15 + 3n	Load Multiple (Store Multiple) dst←R (n consecutive words) (n = 1...16)
	LDR LDRB LDRL	R, src	RA	14	Load Relative R←src (range – 32768... + 32767)
	LDR LDRB LDRL	dst, R	RA	14	Load Relative (Store Relative) dst←R (range – 32768... + 32767)
Pop	POP POPL	dst, R	R IR DA X	8 12 16 16	Pop dst←IR Autoincrement contents of R
Push	PUSH PUSHL	IR, src	R IM IR DA X	9 12 13 14 14	Push Autodecrement contents of R IR←src

BIT MANIPULATION

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Test	BIT BITB	dst, b	R IR DA X	4 8 10 11	Test Bit Static Z flag←NOT dst bit specified by b
	BIT BITB	dst, R	R	10	Test Bit Dynamic Z flag←NOT dst bit specified by contents of R
Reset	RES RESB	dst, b	R IR DA X	4 11 13 14	Reset Bit Static Reset dst bit specified by b
	RES RESB	dst, R	R	10	Reset Bit Dynamic Reset dst bit specified by contents of R
Set	SET SETB	dst, b	R IR DA X	4 11 13 14	Set Bit Static Set dst bit specified by b
	SET SETB	dst, R	R	10	Set Bit Dynamic Set dst bit specified by contents of R
Test and Set	TSET TSETB	dst	R IR DA X	7 11 14 15	Test and Set S flag←MSB of dst dst←all 1s

ROTATE AND SHIFT

	Mnemonics	Operand	Addr. Modes	Clock Cycles†	Operation
Rotate	RLDB	R, src	R	9	Rotate Digit Left
	RRDB	R, src	R	9	Rotate Digit Right
	RL RLB	dst, n	R R	6	Rotate Left by n bits (n = 1, 2)
	RLC RLCB	dst, n	R R	6	Rotate Left through Carry by n bits (n = 1, 2)
	RR RRB	dst, n	R R	6	Rotate Right by n bits (n = 1, 2)
	RRC RRCB	dst, n	R R	6	Rotate Right through Carry by n bits (n = 1, 2)
Shift	SDA SDAB SDAL	dst, R	R	15 + 3n	Shift Dynamic Arithmetic Shift dst left or right by contents of R
	SDL SDLB SDLL	dst, R	R	15 + 3n	Shift Dynamic Logical Shift dst left or right by contents of R
	SLA SLAB SLAL	dst, n	R	13 + 3n	Shift Left Arithmetic by n bits
	SLL SLLB SLLL	dst, n	R	13 + 3n	Shift Left Logical by n bits
	SRA SRAB SRAL	dst, n	R	13 + 3n	Shift Right Arithmetic by n bits
	SRL SRLB SRLL	dst, n	R	13 + 3n	Shift Right Logical by n bits

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

BLOCK TRANSFER AND STRING MANIPULATION (AUTO INCREMENT/DECREMENT AND REPEAT)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Compare	CPD CPDB	R_X, src, R_Y, cc	IR	20	Compare and Decrement R_X – src Autodecrement src address $R_Y \leftarrow R_Y - 1$
	CPDR CPDRB	R_X, src, R_Y, cc	IR	11 + 9n	Compare, Decrement and Repeat R_X – src Autodecrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
	CPI CPIB	R_X, src, R_Y, cc	IR	20	Compare and Increment R_X – src Autoincrement src address $R_Y \leftarrow R_Y - 1$
	CPIR CPIRB	R_X, src, R_Y, cc	IR	11 + 9n	Compare, Increment and Repeat R_X – src Autoincrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
	CPSD CPSDB	dst, src, R, cc	IR	25	Compare String and Decrement dst – src Autodecrement dst and src addresses R ← R – 1
	CPSDR CPSDRB	dst, src, R, cc	IR	11 + 14n	Compare String, Decr. and Repeat dst – src Autodecrement dst and src addresses R ← R – 1 Repeat until cc is true or R = 0
	CPSI CPSIB	dst, src, R, cc	IR	25	Compare String and Increment dst – src Autoincrement dst and src addresses R ← R – 1
	CPSIR CPSIRB	dst, src, R, cc	IR	11 + 14n	Compare String, incr. and Repeat dst – src Autoincrement dst and src addresses R ← R – 1 Repeat until cc is true or R = 0
Load	LDD LDDB	dst, src, R	IR	20	Load and Decrement dst ← src Autodecrement dst and src addresses R ← R – 1
	LDDR LDDRB	dst, src, R	IR	11 + 9n	Load, Decrement and Repeat dst ← src Autodecrement dst and src addresses R ← R – 1 Repeat until R = 0

BLOCK TRANSFER AND STRING MANIPULATION (Cont.)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Load	LDI LDIB	dst, src, R	IR	20	Load and Increment dst ← src Autoincrement dst and src addresses R ← R – 1
	LDIR LDIRB	dst, src, R	IR	11 + 9n	Load, Increment and Repeat dst ← src Autoincrement dst and src addresses R ← R – 1 Repeat until R = 0
Translate	TRDB	dst, src, R	IR	25	Translate and Decrement dst ← src (dst) Autodecrement dst address R ← R – 1
	TRDRB	dst, src, R	IR	11 + 14n	Translate, Decrement and Repeat dst ← src (dst) Autodecrement dst address R ← R – 1 Repeat until R = 0
	TRIB	dst, src, R	IR	25	Translate and Increment dst ← src (dst) Autoincrement dst address R ← R – 1 0
	TRIRB	dst, src, R	IR	11 + 14n	Translate, Increment and Repeat dst ← src (dst) Autoincrement dst address R ← R – 1 Repeat until R = 0
Translate and Test	TRTDB	src 1, src 2, R	IR	25	Translate and Test, Decrement RH1 ← src 2 (src 1) Autodecrement src 1 address R ← R – 1
	TRTDRB	src 1, src 2, R	IR	11 + 14n	Translate and Test, Decrement and Repeat RH1 ← src 2 (src 1) Autodecrement src 1 address R ← R – 1 Repeat until R = 0 or RH1 = 0
	TRTIB	src 1, src 2, R	IR	25	Translate and Test, Increment RH1 ← src 2 (src 1) Autoincrement src 1 address R ← R – 1
	TRTIRB	src 1, src 2, R	IR	11 + 14n	Translate and Test, Increment and Repeat RH1 ← src 2 (src 1) Autoincrement src 1 address R ← R – 1 Repeat until R = 0 or RH1 = 0

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

INPUT/OUTPUT

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Input	IN* INB*	R, src	IR DA	10 12	Input R←src
	IND* INDB*	dst, src, R	IR	21	Input and Decrement dst←src Autodecrement dst address R←R−1
	INDR* INDRB*	dst, src, R	IR	11 + 10n	Input, Decrement and Repeat dst←src Autodecrement dst address R←R−1 Repeat until R = 0
	INI* INIB*	dst, src, R	IR	21	Input and Increment dst←src Autoincrement dst address R←R−1
	INIR* INIRB*	dst, src, R	IR	11 + 10n	Input, Increment and Repeat dst←src Autoincrement dst address R←R−1 Repeat until R = 0
Output	OUT* OUTB*	dst, R	IR DA	10 12	Output dst←R
	OUTD* OUTDB*	dst, src, R	IR	21	Output and Decrement dst←src Autodecrement src address R←R−1
	OTDR* OTDRB*	dst, src, R	IR	11 + 10n	Output and Decrement dst←src Autodecrement src address R←R−1 Repeat until R = 0
	OUTI* OUTIB*	dst, src, R	IR	21	Output and Increment dst←src Autoincrement src address R←R−1
	OTIR* OTIRB*	dst, src, R	IR	11 + 10n	Output, Increment and Repeat dst←src Autoincrement src address R←R−1 Repeat until R = 0
Special Input (identical to input but different status code)	SIN* SINB*	R, src	DA	12	Special Input R←src
	SIND* SINDB*	dst, src, R	IR	21	Special Input and Decrement dst←src Autodecrement dst address R←R−1
	SINDR* SINDRB*	dst, src, R	IR	11 + 10n	Special Input, Decr. and Repeat dst←src Autodecrement dst address R←R−1 Repeat until R = 0
	SINI* SINIB*	dst, src, R	IR	21	Special Input and Increment dst←src Autoincrement dst address R←R−1
	SINIR* SINIRB*	dst, src, R	IR	11 + 10n	Special Input, Incr. and Repeat dst←src Autoincrement dst address R←R−1 Repeat until R = 0

INPUT/OUTPUT (Cont.)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Special Output (identical to output, but different status code)	SOUT* SOUTB*	dst, src	DA	12	Special Output dst←src
	SOUTD* SOUTDB*	dst, src R	IR	21	Special Output and Decrement dst←src Autodecrement src address R←R−1
	SOTDR* SOTDRB*	dst, src, R	IR	11 + 10n	Special Output, Decr. and Repeat dst←src Autodecrement src address R←R−1 Repeat until R = 0
	SOUTI* SOUTIB*	dst, src R	IR	21	Special Output and Increment dst←src Autoincrement src address R←R−1
	SOTIR* SOTIRB*	dst, src R	R	11 + 10n	Special Output, Incr. and Repeat dst←src Autoincrement src address R←R−1 Repeat until R = 0

CPU CONTROL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Interrupts	DI*	int	–	7	Disable Interrupt (Any combination of NVI, VI)
	EI*	int	–	7	Enable Interrupt (Any combination of NVI, VI)
Halt	HALT*	–	–	8 + 3n	HALT
Control Words	LDCTL*	CTLR, src	R	7	Load into Control Register CTLR←src
	LDCTL*	dst, CTLR	R	7	Load into Control Register dst←CTLR
	LDCTLB	dst, FLGR	R	7	Load into Flag Byte Register FLGR←src
	LDPS*	src	IR DA X	12 16 17	Load Program Status PS←src
Multi Micro	MBIT*	–	–	7	Test Multi-Micro Bit Set S if μI is HIGH; reset S if μI is LOW
	MREQ*	dst	R	12 + 7n	Multi-Micro Request
	MRES*	–	–	5	Multi-Micro Reset
	MSET*	–		5	Multi-Micro Set
NOP	NOP			7	No Operation
Flags	RESFLG	flag		7 7	Reset Flag (Any combination of C, Z, S, P/V)
	SETFLG	flag		7	Set Flag (Any combination of C, Z, S, P/V)
	COMFLG	flags	–	7	Complement Flag (Any combination of C, Z, S, P/V)

*Privileged instructions. Executed in system mode only.

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

PROGRAM CONTROL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Call	CALL	dst	IR DA X	10 12 13	Call Subroutine Autodecrement SP @ SP←PC PC←dst
	CALR	dst	RA	10	Call Relative Autodecrement SP @SP←PC PC←PC + dst (range −4094 to +4096)
	SC	src	IM	33	System Call Autodecrement SP @ SP←old PS Push instruction PS←System Call PS
Jump	DJNZ DBJNZ	R, dst	RA	11	Decrement and Jump if Non-Zero R←R−1 IF R = 0: PC←PC + dst (range −254 to 0)
	IRET*	–	–	13	Interrupt Return PS←@SP Autoincrement SP
	JP	cc, dst	IR DA X	7 – 7 8	Jump Conditional If cc is true: PC←dst
	JR	cc, dst	RA	6	Jump Conditional Relative If cc is true: PC←PC + dst (range −256 to +254)
	RET	cc	–	10	Return Conditional If cc is true: PC←@SP Autodecrement SP
Return	IRET*	–	–	13	Interrupt Return PS←@SP Autoincrement SP

*Privileged instructions. Executed in system mode only.

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin
Relative to V_{SS} ..... −0.5 to +7.0V
Power Dissipation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	4MHz	6MHz	8MHz
	Z800X	**Z800XA**	**Z800XA–8**
Commercial Operating Range T_A = 0 to 70°C V_{CC} = 5V ±5%	Z800XDC[2] Z800XPC Z800XLC	Z800XADC Z800XAPC Z800XALC	Z800XA–8DC Z800XA–8PC Z800XA–8LC
Industrial Operating Range T_A = −40 to +85°C V_{CC} = 5V ±10%	Z800XDI Z800XLI	Z800XADI Z800XALI	Z800XA–8DI Z800XA–8LI

Notes: 1. T_A = Ambient Temperature
2. Add suffix B to indicate burn-in requirement.

Operating ranges define those limits over which the functionality of the device is guaranteed.

4, 6MHz Devices
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	V_{CC}−0.4	V_{CC}+0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	V_{CC}+0.3	Volts
V_{IH} $\overline{NMI}$, Reset	Input High Voltage		2.4	V_{CC}+0.3	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		Volts
V_{OL}	Output Low Voltage	I_{OL} = +2.0mA		0.4	Volts
I_{IL}	Input Leakage	$0.4 \leq V_{IN} \leq +2.4V$		±10	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4V$		±10	μA
ICC	V_{CC} Supply Current	Commercial		300	mA

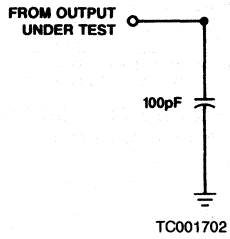
8MHz Devices
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	V_{CC}−0.4	V_{CC}+0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	V_{CC}+0.3	Volts
V_{IH} $\overline{NMI}$, Reset	Input High Voltage		2.4	V_{CC}+0.3	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	I_{OH} = −250μA	2.4		Volts
V_{OL}	Output Low Voltage	I_{OL} = +2.0mA		0.4	Volts
I_{IL}	Input Leakage	$0.4 \leq V_{IN} \leq +2.4V$		±10	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4V$		±10	μA
I_{CC}	V_{CC} Supply Current	Commercial		325	mA

Standard Test Conditions

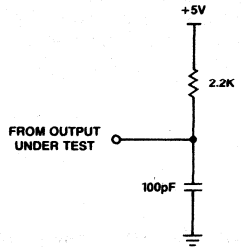
The characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

Standard Test Load



TC001702

Open Drain Test Load



TC001711

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Parameters	Description	4MHz Devices COML/IND Min	4MHz Devices COML/IND Max	6MHz Devices COML/IND Min	6MHz Devices COML/IND Max	Units
1	TcC	Clock Cycle Time	250	2000	165	2000	ns
2	TwCh	Clock Width (HIGH)	105	2000	70	2000	ns
3	TwCl	Clock Width (LOW)	105	2000	70	2000	ns
4	TfC	Clock Fall Time		20		10	ns
5	TrC	Clock Rise Time		20		10	ns
6†	TdC(SNv)	Clock ↑ to Segment Number Valid (50pF Load)		130		110	ns
7†	TdC(SNn)	Clock ↑ to Segment Number Not Valid	20		10		ns
8	TdC(Bz)	Clock ↑ to Bus Float		65		55	ns
9	TdC(A)	Clock ↑ to Address Valid		100		75	ns
10	TdC(Az)	Clock ↑ to Address Float		65		55	ns
11	TdA(Dl)*	Address Valid to Data In Required Valid		475		305	ns
12	TsDl(C)	Data In to Clock ↓ Set-up Time	30		20		ns
13	TdDS(A)*	$\overline{DS}$ ↑ to Address Active	80		45		ns
14	TdC(DO)	Clock ↑ to Data Out Valid		100		75	ns
15	ThDl(DS)	Data In to $\overline{DS}$ ↑ Hold Time	0		0		ns
16	TdDO(DS)*	Data Out Valid to $\overline{DS}$ ↑ Delay	295		195		ns
17	TdA(MR)	Address Valid to $\overline{MREQ}$ ↓ Delay	55		35		ns
18	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		80		70	ns
19	TwMRh*	$\overline{MREQ}$ Width (HIGH)	210		135		ns
20	TdMR(A)*	$\overline{MREQ}$ ↓ to Address Not Active	70		35		ns
21	TdDO(DSW)*	Data Out Valid to $\overline{DS}$ ↓ (Write) Delay	55		35		ns
22	TdMR(Dl)*	$\overline{MREQ}$ ↓ to Data In Required Valid	375		225		ns
23	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		80		60	ns
24	TdC(ASf)	Clock ↑ to $\overline{AS}$ ↓ Delay		80		60	ns
25	TdA(AS)*	Address Valid to $\overline{AS}$ ↑ Delay	55		35		ns
26	TdC(ASr)	Clock ↓ to $\overline{AS}$ ↑ Delay		90		80	ns
27	TdAS(Dl)*	$\overline{AS}$ ↑ to Data In Required Valid	360		215		ns
28	TdDS(AS)*	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	70		35		ns
29	TwAS*	$\overline{AS}$ Width (LOW)	85		55		ns
30	TdAS(A)*	$\overline{AS}$ ↑ to Address Not Active Delay	70		30		ns
31	TdAz(DSR)	Address Float to $\overline{DS}$ (Read) ↓ Delay	0		0		ns
32	TdAS(DSR)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Read) ↓ Delay	80		35		ns

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (Cont.)

Number	Parameters	Description	4MHz Devices COML/IND Min	4MHz Devices COML/IND Max	6MHz Devices COML/IND Min	6MHz Devices COML/IND Max	Units
33	TdDSR(DI)*	$\overline{DS}$ (Read) ↓ to Data In Required Valid	205		130		ns
34	TdC(DSr)	Clock ↓ to $\overline{DS}$ ↑ Delay		70		65	ns
35	TdDS(DO)*	$\overline{DS}$ ↑ to Data Out and STATUS Not Valid	75		45		ns
36	TdA(DSR)*	Address Valid to $\overline{DS}$ (Read) ↓ Delay	180		110		ns
37	TdC(DSR)	Clock ↑ to $\overline{DS}$ (Read) ↓ Delay		120		85	ns
38	TwDSR*	$\overline{DS}$ (Read) Width (LOW)	275		185		ns
39	TdC(DSW)	Clock ↓ to $\overline{DS}$ (Write) ↓ Delay		95		80	ns
40	TwDSW*	$\overline{DS}$ (Write) Width (LOW)	185		110		ns
41	TdDSI(DI)*	$\overline{DS}$ (Input) ↓ to Data In Required Valid	330		200		ns
42	TdC(DSI)	Clock ↓ to $\overline{DS}$ (I/O) ↓ Delay		120		100	ns
43	TwDS*	$\overline{DS}$ (I/O) Width (LOW)	410		255		ns
44	TdAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay	1065		690		ns
45	TdC(DSA)	Clock ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay		120		85	ns
46	TdDSA(DI)*	$\overline{DS}$ (Acknowledge) ↓ to Data In Required Delay	455		295		ns
47	TdC(S)	Clock ↑ to Status Valid Delay		110		85	ns
48	TdS(AS)*	Status Valid to AS ↑ Delay	50		30		ns
49	TsR(C)	$\overline{RESET}$ to Clock ↑ Set-up Time	180		70		ns
50	ThR(C)	$\overline{RESET}$ to Clock ↑ Hold Time	0		0		ns
51	TwNMI	$\overline{NMI}$ Width (LOW)	100		70		ns
52	TsNMI(C)	$\overline{NMI}$ to Clock ↑ Set-up Time	140		70		ns
53	TsVI(C)	$\overline{VI}$, $\overline{NVI}$ to Clock ↑ Set-up Time	110		50		ns
54	ThVI(C)	$\overline{VI}$, $\overline{NVI}$ to Clock ↑ Hold Time	20		20		ns
55†	TsSGT(C)	$\overline{SEGT}$ to Clock ↑ Set-up Time	70		55		ns
56†	ThSGT(C)	$\overline{SEGT}$ to Clock ↑ Hold Time	0		0		ns
57	TsMI(C)	$\overline{MI}$ to Clock ↑ Set-up Time	180		110		ns
58	ThMI(C)	$\overline{MI}$ to Clock ↑ Hold Time	0		0		ns
59	TdC(MO)	Clock ↑ to $\overline{MO}$ Delay		120		85	ns
60	TsSTP(C)	$\overline{STOP}$ to Clock ↓ Set-up Time	140		80		ns
61	ThSTP(C)	$\overline{STOP}$ to Clock ↓ Hold Time	0		0		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock ↓ Set-up Time	50		30		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock ↓ Hold Time	10		10		ns
64	TsBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Set-up Time	90		80		ns
65	ThBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Hold Time	10		10		ns
66	TdC(BAKr)	Clock ↑ to $\overline{BUSAK}$ ↑ Delay		100		75	ns
67	TdC(BAKf)	Clock ↑ to $\overline{BUSAK}$ ↓ Delay		100		75	ns
68	TwA	Address Valid Width	150		95		ns
69	TdDS(S)	$\overline{DS}$ ↑ to STATUS Not Valid	80		55		ns

*Clock-cycle-time-dependent characteristics. These numbers are computed assuming the clock characteristics are at the limits given in parameters 1 through 5. For other clock frequencies, these parameters can be derived from other specs and the clock characteristics. See tables on following pages.

†Z8001 and Z8001A only.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Parameters	Description	8MHz Devices COML/IND Min	8MHz Devices COML/IND Max	Units
1	TcC	Clock Cycle Time	125	2000	ns
2	TwCh	Clock Width (HIGH)	55	2000	ns
3	TwCl	Clock Width (LOW)	55	2000	ns
4	TfC	Clock Fall Time		10	ns
5	TrC	Clock Rise Time		10	ns
6†	TdC(SNv)	Clock ↑ to Segment Number Valid (50pF Load)		100	ns
7†	TdC(SNn)	Clock ↑ to Segment Number Not Valid	10		ns
8	TdC(Bz)	Clock ↑ to Bus Float		50	ns
9	TdC(A)	Clock ↑ to Address Valid		65	ns
10	TdC(Az)	Clock ↑ to Address Float		45	ns
11	TdA(DI)*	Address Valid to Data In Required Valid		225	ns
12	TsDI(C)	Data In to Clock ↓ Set-up Time	15		ns
13	TdDS(A)*	$\overline{DS}$ ↑ to Address Active	40		ns
14	TdC(DO)	Clock ↑ to Data Out Valid		65	ns
15	ThDI(DS)	Data In to $\overline{DS}$ ↑ Hold Time	0		ns
16	TdDO(DS)*	Data Out Valid to $\overline{DS}$ ↑ Delay	150		ns
17	TdA(MR)*	Address Valid to $\overline{MREQ}$ ↓ Delay	30		ns
18	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		55	ns
19	TwMRh*	$\overline{MREQ}$ Width (HIGH)	105		ns
20	TdMR(A)*	$\overline{MREQ}$ ↓ to Address Not Active	35		ns
21	TdDO(DSW)*	Data Out Valid to $\overline{DS}$ ↓ (Write) Delay	30		ns
22	TdMR(DI)*	$\overline{MREQ}$ ↓ to Data In Required Valid	175		ns
23	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		55	ns
24	TdC(ASf)	Clock ↑ to $\overline{AS}$ ↓ Delay		55	ns
25	TdA(AS)*	Address Valid to $\overline{AS}$ ↑ Delay	30		ns
26	TdC(ASr)	Clock ↓ to $\overline{AS}$ ↑ Delay		65	ns
27	TdAS(DI)*	$\overline{AS}$ ↑ to Data In Required Valid	170		ns
28	TdDS(AS)*	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	35		ns
29	TwAS*	$\overline{AS}$ Width (LOW)	45		ns
30	TdAS(A)*	$\overline{AS}$ ↑ to Address Not Active Delay	30		ns
31	TdAz(DSR)	Address Float to $\overline{DS}$ (Read) ↓ Delay	0		ns
32	TdAS(DSR)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Read) ↓ Delay	30		ns
33	TdDSR(DI)*	$\overline{DS}$ (Read) ↓ to Data In Required Valid	115		ns
34	TdC(DSr)	Clock ↓ to $\overline{DS}$ ↑ Delay		65	ns
35	TdDS(DO)*	$\overline{DS}$ ↑ to Data Out and STATUS Not Valid	40		ns
36	TdA(DSR)*	Address Valid to $\overline{DS}$ (Read) ↓ Delay	85		ns
37	TdC(DSR)	Clock ↑ to $\overline{DS}$ (Read) ↓ Delay		70	ns
38	TwDSR*	$\overline{DS}$ (Read) Width (LOW)	140		ns
39	TdC(DSW)	Clock ↓ to $\overline{DS}$ (Write) ↓ Delay		65	ns
40	TwDSW*	$\overline{DS}$ (Write) Width (LOW)	85		ns
41	TdDSI(DI)*	$\overline{DS}$ (Input) ↓ to Data In Required Valid	135		ns
42	TdC(DSf)	Clock ↓ to $\overline{DS}$ (I/O) ↓ Delay		85	ns
43	TwDS*	$\overline{DS}$ (I/O) Width (LOW)	200		ns
44	TdAS(DSA)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay	520		ns
45	TdC(DSA)	Clock ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay		65	ns

†Z8001A-8 only.

SWITCHING CHARACTERISTICS (Cont.)

Number	Parameters	Description	8MHz Devices COML/IND Min	8MHz Devices COML/IND Max	Units
46	TdDSA(DI)*	$\overline{DS}$ (Acknowledge) ↓ to Data In Required Delay	235		ns
47	TdC(S)	Clock ↑ to Status Valid Delay		75	ns
48	TdS(AS)*	Status Valid to $\overline{AS}$ ↑ Delay	25		ns
49	TsR(C)	$\overline{RESET}$ to Clock ↑ Set-up Time	70		ns
50	ThR(C)	$\overline{RESET}$ to Clock ↑ Hold Time	0		ns
51	TwNMI	$\overline{NMI}$ Width (LOW)	50		ns
52	TsNMI(C)	$\overline{NMI}$ to Clock ↑ Set-up Time	70		ns
53	TsVI(C)	$\overline{VI}$,$\overline{NVI}$, to Clock ↑ Set-up Time	50		ns
54	ThVI(C)	$\overline{VI}$,$\overline{NVI}$ to Clock ↑ Hold Time	20		ns
55†	TsSGT(C)	$\overline{SEGT}$ to Clock ↑ Set-up Time	45		ns
56†	ThSGT(C)	$\overline{SEGT}$ to Clock ↑ Hold Time	0		ns
57	TsMI(C)	$\overline{MI}$ to Clock ↑ Set-up Time	90		ns
58	ThMI(C)	$\overline{MI}$ to Clock ↑ Hold Time	0		ns
59	TdC(MO)	Clock ↑ to $\overline{MO}$ Delay		65	ns
60	TsSTP(C)	$\overline{STOP}$ to Clock ↓ Set-up Time	75		ns
61	ThSTP(C)	$\overline{STOP}$ to Clock ↓ Hold Time	0		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock ↓ Set-up Time	25		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock ↓ Hold Time	10		ns
64	TsBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Set-up Time	60		ns
65	ThBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Hold Time	10		ns
66	TdC(BAKr)	Clock ↑ to $\overline{BUSAK}$ ↑ Delay		60	ns
67	TdC(BAKf)	Clock ↑ to $\overline{BUSAK}$ ↓ Delay		60	ns
68	TwA*	Address Valid Width	90		ns
69	TdDS(S)*	$\overline{DS}$ ↓ to STATUS Not Valid	45		ns

*Clock-cycle-time-dependent characteristics. These numbers are computed assuming the clock characteristics are at the limits given in parameters 1 through 5. For other clock frequencies, three parameters can be derived from other specs and the clock characteristics. See following table.

†Z8001A-8 only.

5

00971B

Refer to page 7-1 for Essential Information on Military Devices

CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

The parameters listed below are also shown in the switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence, so the correct limit for these parameters may be determined for any system, regardless of the actual clock characteristics.

Number	Parameters	4MHz Devices	6MHz Devices
11	TdA(DI)	2TcC + TwCH – 125ns	2TcC + TwCh – 95ns
13	TdDS(C)	TwCl – 25ns	TWCl – 30ns
16	TdDO(DS)	TcC + TwCh – 60ns	TcC+ TwCh – 40ns
17	TdA(MR)	TwCh – 50ns	TwCh – 35ns
19	TwMRh	TcC – 40ns	TcC – 30ns
20	TdMR(A)	TwCl – 35ns	TwCl – 35ns
21	TdDO(DSW)	TwCh – 50ns	TwCh – 35ns
22	TdMR(DI)	2TcC – 125ns	2TcC – 105ns
25	TdA(AS)	TwCh – 50ns	TwCh – 35ns
27	TdAS(DI)	2TcC – 140ns	2TcC – 115ns
28	TdDS(AS)	TwCl – 35ns	TwCl – 35ns
29	TwAS	TwCh – 20ns	TwCh – 15ns
30	TdAS(A)	TwCl – 35ns	TwCl – 40ns
32	TdAS(DSR)	TwCl – 25ns	TwCl – 35ns
33	TdDSR(DI)	TcC + TwCh – 150ns	TcC+ TwCh – 105ns
35	TdDS(DO)	TwCl – 30ns	TwCl – 25ns
36	TdA(DSR)	TcC – 70ns	TcC – 55ns
38	TwDSR	TcC + TwCh – 80ns	TcC+ TwCh – 50ns
40	TwDSW	TcC – 65ns	TcC – 55ns
41	TdDSI(DI)	2TcC – 170ns	2TcC – 130ns
43	TwDS	2TcC – 90ns	2TcC – 75ns
44	TdAS(DSA)	4TcC + TwCh – 40ns	4TcC – TwCl – 40ns
46	TdDSA(DI)	2TcC + TwCh – 150ns	2TcC+ TwCh – 105ns
48	TdS(AS)	TwCh – 55ns	TwCh – 40ns
68	TwA	TcC – 90ns	TcC – 70ns
69	TdDS(S)	TwCl – 25ns	TwCl – 15ns

CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

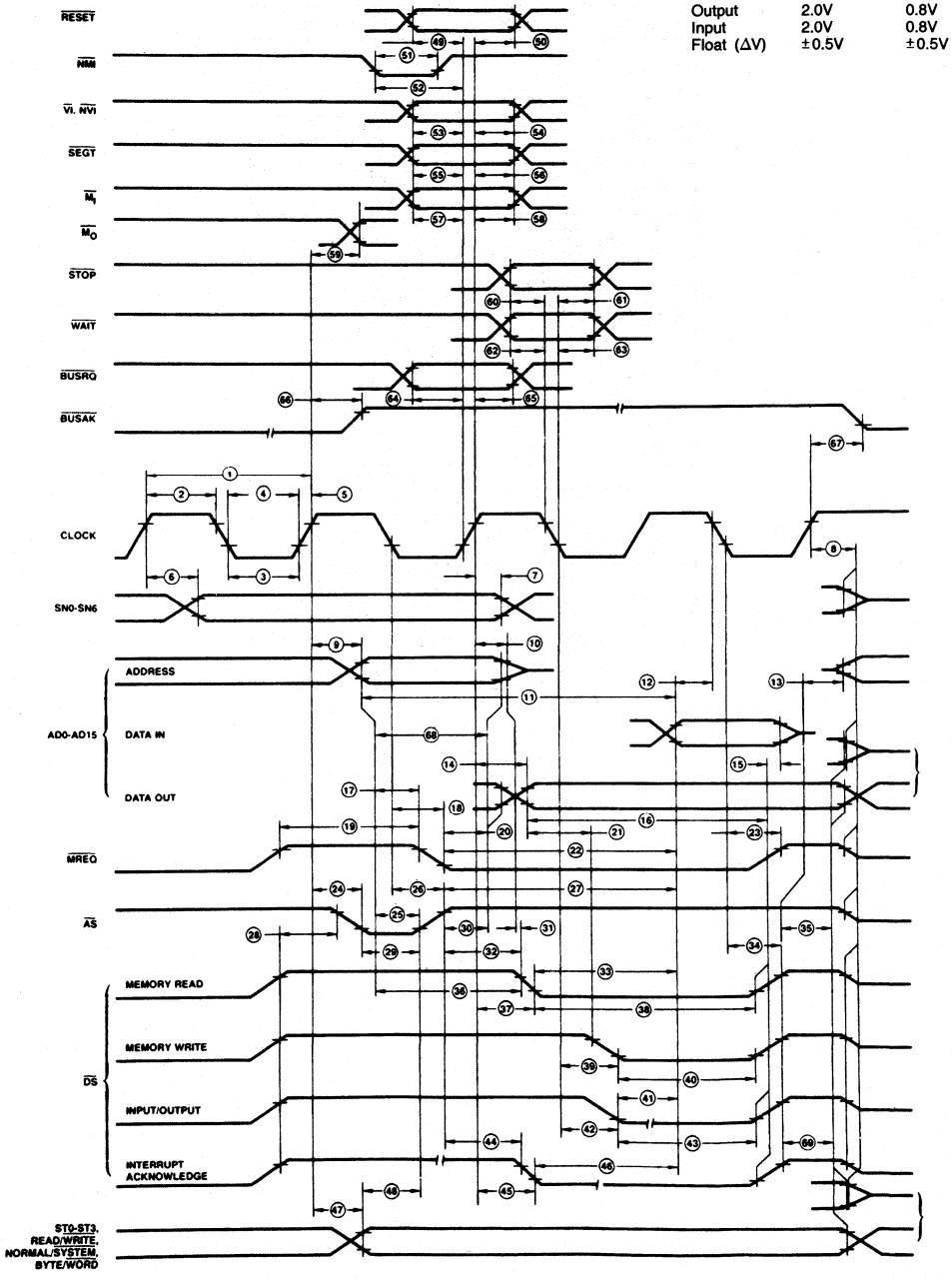
The parameters listed below are also shown in the switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence, so the correct limt for these parameters may be determined for any system, regardless of the actual clock characteristics.

Number	Parameters	8MHz Devices
11	TdA(DI)	2TcC + TwCh – 80ns
13	TdDS(A)	TwCl – 15ns
16	TdDO(DS)	TcC + TwCh – 30ns
17	TdA(MR)	TwCh – 25ns
19	TwMRh	TcC – 20ns
20	TdMR(A)	TwCl – 20ns
21	TdDO(DSW)	TwCh – 25ns
22	TdMR(DI)	2TcC – 75ns
25	TdA(AS)	TwCh – 25ns
27	TdAS(DI)	2TcC – 80ns
28	TdDS(AS)	TwCl – 20ns
29	TwAS	TwCh – 10ns
30	TdAS(A)	TwCl – 25ns
32	TdAS(DSR)	TwCl – 25ns
33	TdDSR(DI)	TcC + TwCh – 65ns
35	TdDS(DO)	TwCl – 15ns
36	TdA(DSR)	TcC – 40ns
38	TwDSR	TcC + TwCh – 40ns
40	TwDSW	TcC – 40ns
41	TdDSI(DI)	2TcC – 115ns
43	TwDS	2TcC – 50ns
44	TdAS(DSA)	4TcC + TwCl – 35ns
46	TdDSA(DI)	2TcC + TwCh – 70ns
48	TdS(AS)	TwCh – 30ns
68	TwA	Tcc – 35ns
69	TdDS(S)	TwCl – 10ns

Z8001 TIMING DIAGRAM

Timing Measurements are made at the following voltages.

	HIGH	LOW
Clock	4.0V	0.8V
Output	2.0V	0.8V
Input	2.0V	0.8V
Float (ΔV)	±0.5V	±0.5V



WF005290

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

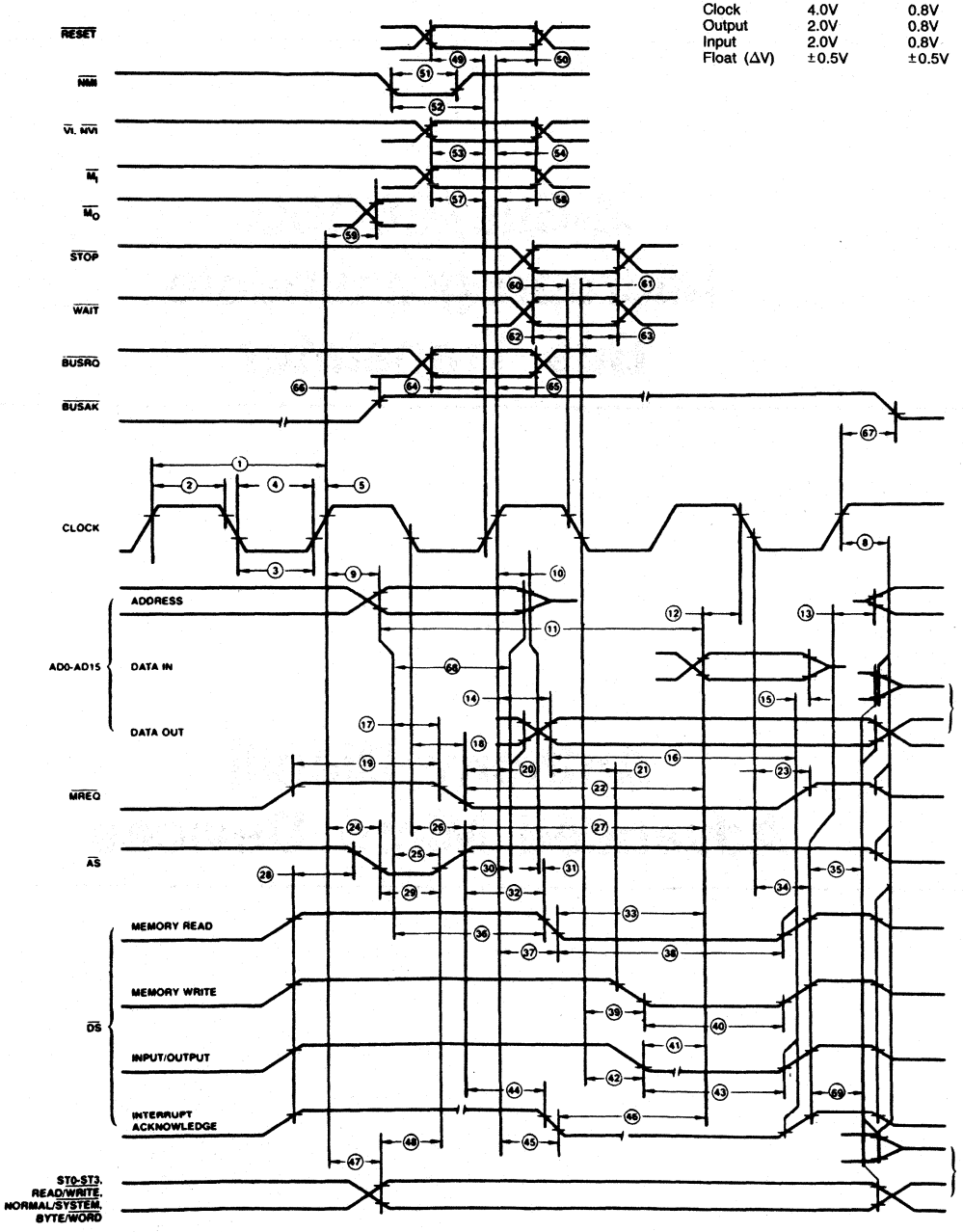
00971B

Refer to page 7-1 for Essential Information on Military Devices

Z8002 TIMING DIAGRAM

Timing Measurements are made at the following voltages.

	HIGH	LOW
Clock	4.0V	0.8V
Output	2.0V	0.8V
Input	2.0V	0.8V
Float (ΔV)	±0.5V	±0.5V



WF005300

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Z-Bus/68000 Microprogrammable Bus Translator

Anthony Dicolli
Advanced Micro Devices

00971B

Refer to page 7-1 for Essential Information on Military Devices

Bus Translator

ABSTRACT

This paper describes an interface technique that permits all speed versions of the 68000 CPU to communicate with all Z8000 peripherals. Further, the microprogrammable nature of this interface allows intermixing of various speed peripherals on the same Z-Bus by dynamically modifying the bus translator's timing characteristics on a cycle-by-cycle basis. Included are a circuit description, PROM programs, PAL equations and a discussion on a typical system's architecture.

BUS TRANSLATOR CIRCUIT DESCRIPTION

(Figures 1 and 2)

The bus multiplexer contains one 8-bit bus buffer (Am29827) and two bus transceivers (Am29863). These components accept the separate address and data buses of the 68000 and, using three-state techniques, multiplex them together to form the Z-Bus address/data path (AD Bus). Only six address lines are required to directly address all control and data registers in the Z8000 peripherals. The timing sequence of this multiplexing operation is derived from the state sequencer and the translator logic.

The translator logic contains hard-wired logic elements that accept 68000 bus control signals (CLK, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$) as inputs and, in conjunction with the state sequencer inputs (T_1, T_2, T_3, T_4, T_5), produce Z-Bus control signals ($\overline{AS}$, $\overline{DS}$, A_0, R/$\overline{W}$, B/$\overline{W}$), 68000 $\overline{DTACK}$, and the bus multiplexer control signals.

In addition, a Z8000 peripheral inhibit input (ZINH) is provided to prevent the bus translator from responding to 68000 bus operations when none of the Z8000 peripherals is being accessed (i.e., a 68000 instruction fetch). Also, a translator output enable ($\overline{TOE}$) is provided, so that control of the Z-Bus can be relinquished to another master, such as a DMA Controller, if required (see Figure 9). Both $\overline{TOE}$ and $\overline{ZINH}$ are generated by an external chip select decoder (see System Architecture section). This translator logic can easily be implemented utilizing a PAL (AmPAL16R4) as shown in Figure 3. The fuse map for this PAL is detailed in Figure 4 and the design equations are:

$\overline{AOE} = /T_2 * \overline{MAS}$
$\overline{ZAS} = /T_1 * \overline{MAS}$
$\overline{DOE}: = T_3 * \overline{MUDS} + T_3 * \overline{MLDS}$
$\overline{ZDS}: = T_4 * \overline{MUDS} + T_4 * \overline{MLDS}$
$ZB/\overline{W}: = /\overline{MUDS}: + :/\overline{MLDS}$
$\overline{MDTACK}: = T_5 * \overline{MUDS} + T_5 * \overline{MLDS}$
$/ZA_0 + \overline{MLDS}$
;when $\overline{TOE}$ is LOW

The state sequencer, shown in Figure 2, is a registered PROM (Am27S35) which provides microprogramming ability for the bus translator. The registered PROM contains a fusible-link PROM memory array, an output register, and an (user-programmable) initialize word (Figure 5). When $\overline{INIT}$ is pulled LOW, the contents of the initialize word are sent to the output, regardless of the state of the clock or the address inputs. In this application, the initialize word is programmed to OOH. Outputs O_0 through O_4 generate timing signals T_1 through T_5, which are used as gating inputs by the translator logic. These gating signals along with the 68000 control signals determine when the Z-Bus control signals will be activated and the duration of these signals. The program sequences stored in this PROM are user-definable and are a function of the CPU and peripheral types and speeds implemented in the system.

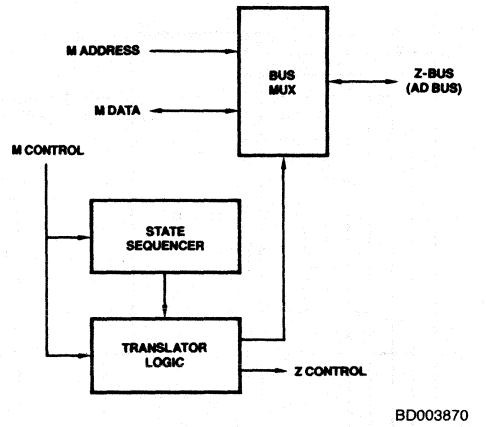


Figure 1.

The next address to be accessed is determined by the clock strobed data outputs, O_0 through O_7. For this to work properly, a unique and non-redundant output word must exist for every clock cycle. Since there will be instances where T_1 through T_5 will not change for many clock cycles, a "counter" function must be included via O_5, O_6 and O_7. This counter function allows up to 8 idle states (83.3ns per idle state) to be inserted between T_1 through T_5 transitions. Address lines A_8 and A_9 provide the ability to dynamically select up to 4 individual sequences. Figure 6 is a PROM program required to generate the read and write cycles shown in Figure 8. When A_8 is LOW, a read cycle is selected; with A_8 in the HIGH state, a write cycle is selected; with A_8 LOW and A_9 HIGH, a "read interrupt" cycle is selected. Other possible sequences would support intermixing of 4 and 6MHz peripherals on a cycle-by-cycle basis.

This state sequencer design assumes that a minimum chip count is preferred and that PROM space is inexpensive. In the example of Figure 6, only 29 locations out of 1,000 are used. If smaller memories are preferred and chip count is not critical, then the state sequencer in Figure 7 can be substituted. In this example, a hardware counter is provided so that no "counter" function is required in the PROM. Also, the PROM array need only be 64 words deep. This approach also allows more sequences to be added without drastically increasing the PROM array size. For instance, a separate read and write cycle for 4 and 6MHz peripherals, a separate read interrupt vector for 4 and 6MHz peripherals, and a two speed CPU cycle would require only 256 words of PROM space.

It should be noted that the address inputs, used to select the sequence to be enabled, are generated by the same external chip select decoder that generates $\overline{ZINH}$.

BUS TRANSLATOR TIMING ANALYSIS

Figure 8 illustrates the timing relationships between a 12MHz 68000 and all 4MHz Z8000 peripherals. The signals shown are the 68000 control, address, and data inputs to the bus translator of Figure 2, and the corresponding Z-Bus control, address, and data outputs. The following discussion assumes that the 68000 CPU, bus translator and Z8000 peripherals are physically located on the same circuit board. If a backplane is used, its propagation delays must also be considered; however, the set-up and hold times in this illustration will be sufficient for most system architectures.

5

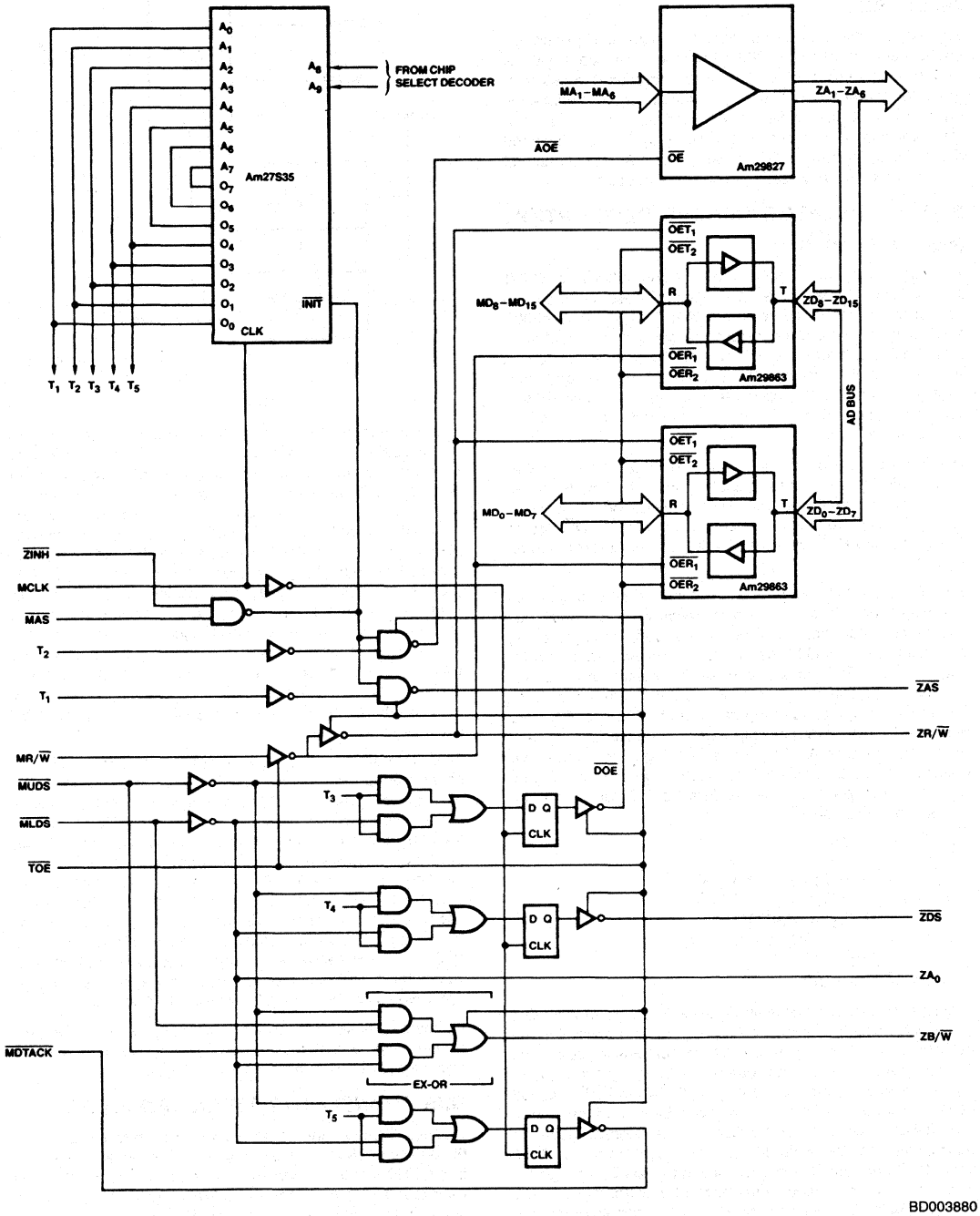
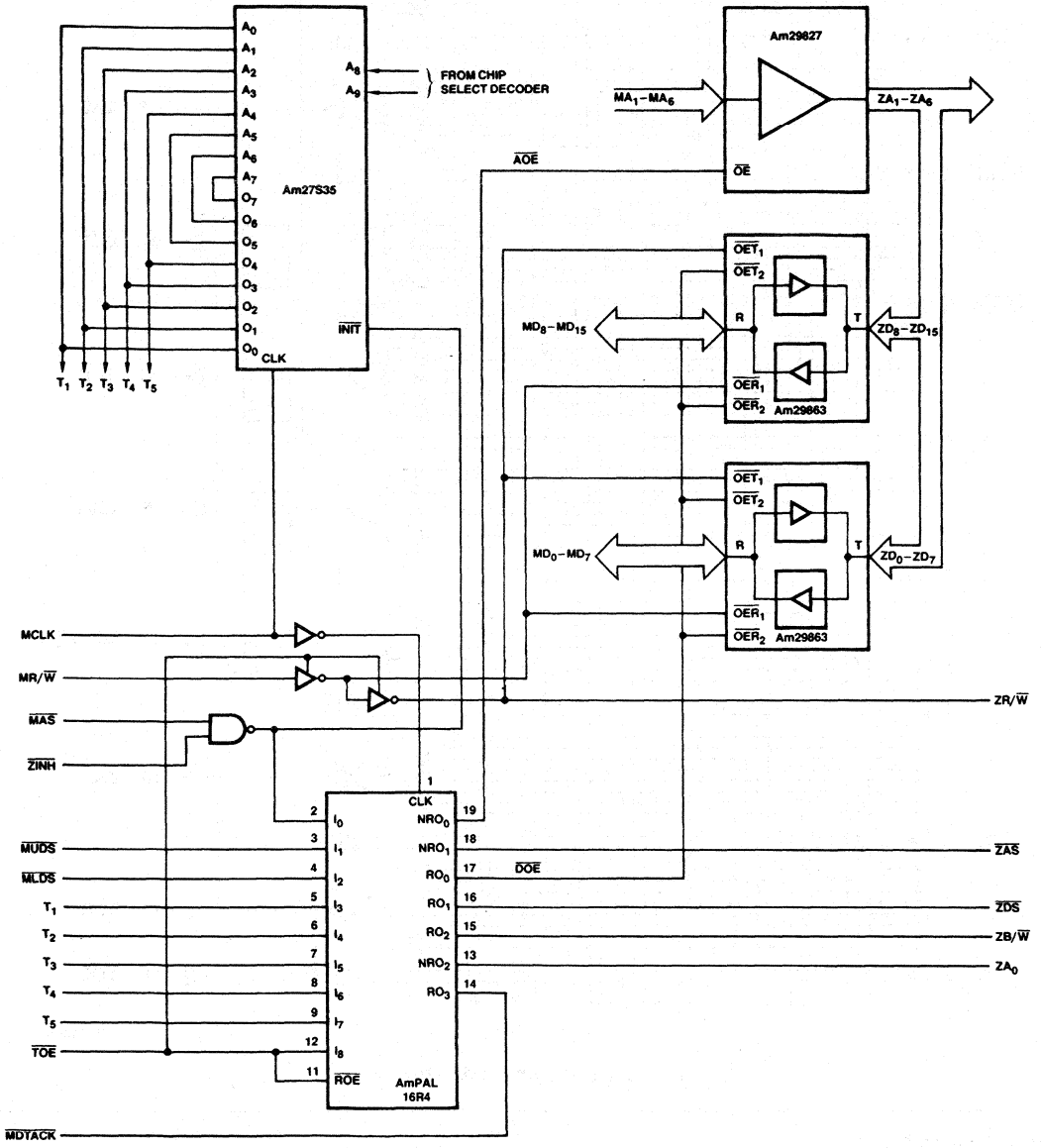

A0
A1
A2
A3
A4
A5
A6
A7
O7
O6
O5
O4
O3
O2
O1
O0
CLK
A8
A9
Am27S35
INIT
FROM CHIP SELECT DECODER
T1 T2 T3 T4 T5
MA1–MA6
ZA1–ZA6
AOE
OE
Am29827
OET1
OET2
R
T
OER1
OER2
Am29863
MD8–MD15
ZD8–ZD15
AD BUS
MD0–MD7
ZD0–ZD7
ZINH
MCLK
MAS
T2
T1
MR/W
MUDS
MLDS
TOE
T3
T4
T5
D Q
CLK
DOE
ZAS
ZR/W
ZDS
ZA0
ZB/W
MDTACK
EX-OR
BD003880

Figure 2.



BD003890

Figure 3.

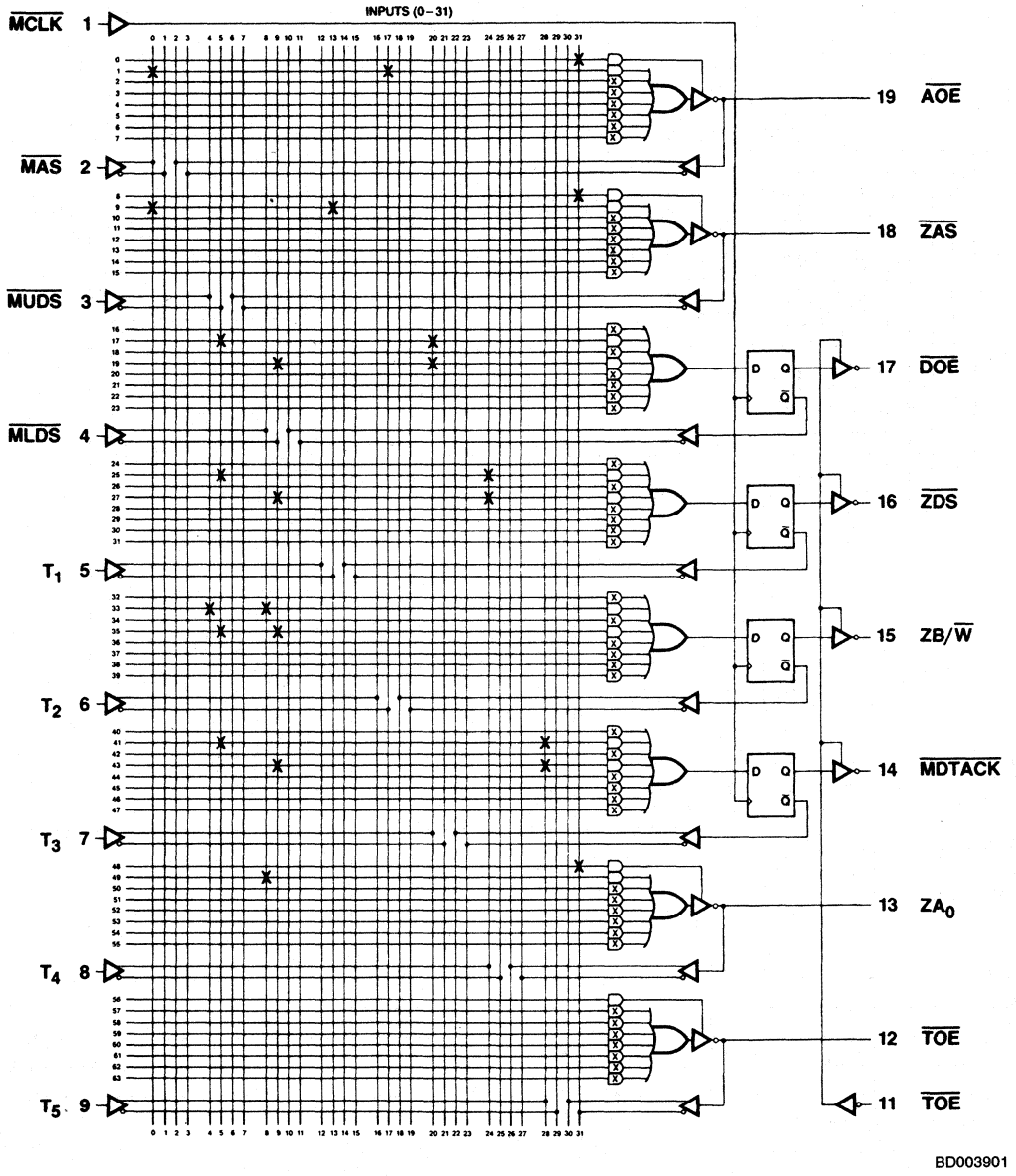


Figure 4. Logic Diagram PAL16R4

Bus Translator

During S_0 the read/write line from the 68K will be set HIGH to indicate a read operation. MR/$\overline{W}$ puts the transceivers in the receive mode and is used by the Z-Bus as ZR/$\overline{W}$. In a read operation, the 68K address will become valid sometime during S_1, and the external chip select decoder will read this address and determine whether a Z8000 I/O operation is being started. If it is a Z8000 I/O operation, $\overline{ZINH}$, A_8, and A_9 will be appropriately set. $\overline{MAS}$ will be asserted during S_2. This action: 1) releases the state sequencer via $\overline{INIT}$ and begins the Tx state on the next positive edge of $\overline{MCLK}$ (shown as 0_1), 2) causes $\overline{AOE}$ to go active, thus driving the AD bus with the address, and 3) asserts $\overline{ZAS}$. $\overline{MUDS}$ and/or $\overline{MLDS}$ are also asserted in S_2 and cause ZA_0 and ZB/$\overline{W}$ to be appropriately set. At 0_2 the state sequencer asserts T_1, causing $\overline{ZAS}$ to be negated. This phase was chosen to meet the required 70ns (min) $\overline{ZAS}$ pulse width (TwAS), and 30ns (min) address to $\overline{ZAS}$

set-up time (TsA (AS)). At 0_3 the state sequencer asserts T_2 and T_4. T_2 causes the $\overline{AOE}$ line to negate which removes the address and three-states the AD bus. This meets the required 50ns (min) address to $\overline{ZAS}$ hold time (ThA (AS)). T_4 causes the $\overline{ZDS}$ to be asserted on the negative edge of 0_3. This meets the $\overline{ZAS}$ to $\overline{ZDS}$ delay of 60ns (min).

On the falling edge of $\overline{ZDS}$, the peripheral will drive the AD bus, and data on the AD bus will be valid 250ns later (an exception to this occurs in the Z8030 which requires 520ns from rising edge of $\overline{ZAS}$ to valid read data). At 0_5 the sequencer sets T_5, which in turn causes $\overline{MDTACK}$ to be asserted on the negative edge of 0_5. The 68K samples this line on the negative edge of 0_6 and accepts the input data on the negative edge of 0_7. At 0_6 the T_3 line is asserted, which activates $\overline{DOE}$ on the negative edge of 0_6. This enables AD bus data to the 68K data bus. T_3 may be set to occur anytime before the negative edge of 0_7. During S_7 the 68K will negate $\overline{MAS}$, $\overline{MUDS}$, and $\overline{MLDS}$. This action causes the sequencer to reset (via $\overline{INIT}$), and on the negative edge of 0_8, $\overline{ZDS}$, $\overline{MDTACK}$, and $\overline{DOE}$ will be negated to end the read cycle. Note that the peripheral provides the zero data hold time required by the 68000. It should also be noted that $\overline{DTACK}$ is used to insert wait states (2 wait states = 1 MCLK). The assertion of $\overline{DTACK}$ via T_5 is a function of the minimum required $\overline{ZDS}$ pulse width (in this case 390ns), and the minimum time required for the peripheral to provide valid data to the AD bus. Therefore, in using the Z8030, T_5 would not occur (this is accomplished by inserting 2 idle states in the sequencer) until 0_7, and the rest of the read cycle would be proportionately extended.

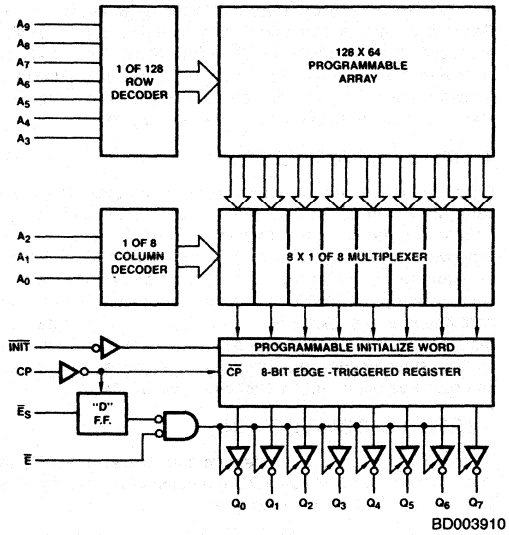


Figure 5. Am27S35 Block Diagram

			ADDR INPUT											PROM OUTPUT									REG. OUTPUT								
$\overline{INIT}$	CLOCK		A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	HEX	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	HEX	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	HEX
0	X		0	0	0	0	0	0	0	0	0	0	000	0	0	0	0	0	1	0	0	20	0	0	0	0	0	0	0	0	00
1	0_1		0	0	0	0	0	1	0	0	0	0	020	1	0	0	0	0	0	0	0	01	0	0	0	0	0	1	0	0	20
1	0_2	READ CYCLE	1	0	0	0	0	0	0	0	0	0	001	1	1	0	1	0	0	0	0	0B	1	0	0	0	0	0	0	0	01
1	0_3		1	1	0	1	0	0	0	0	0	0	00B	1	1	0	1	0	1	0	0	2B	1	1	0	1	0	0	0	0	0B
1	0_4		1	1	0	1	0	1	0	0	0	0	02B	1	1	0	1	1	0	0	0	1B	1	1	0	1	0	1	0	0	2B
1	0_5		1	1	0	1	1	0	0	0	0	0	01B	1	1	1	1	1	0	0	0	1F	1	1	0	1	1	0	0	0	1B
1	0_6		1	1	1	1	1	0	0	0	0	0	01F	1	1	1	1	1	0	0	0	1F	1	1	1	1	1	0	0	0	1F
0	X		0	0	0	0	0	0	0	0	1	0	100	0	0	0	0	0	1	0	0	20	0	0	0	0	0	0	0	0	00
1	0_1		0	0	0	0	0	1	0	0	1	0	120	1	0	0	0	0	0	0	0	01	0	0	0	0	0	1	0	0	20
1	0_2		1	0	0	0	0	0	0	0	1	0	101	1	1	1	0	0	0	0	0	07	1	0	0	0	0	0	0	0	01
1	0_3		1	1	1	0	0	0	0	0	1	0	107	1	1	1	1	0	0	0	0	0F	1	1	1	1	0	0	0	0	07
1	0_4		1	1	1	1	0	0	0	0	1	0	10F	1	1	1	1	0	1	0	0	2F	1	1	1	1	0	0	0	0	0F
1	0_5		1	1	1	1	0	1	0	0	1	0	12F	1	1	1	1	0	0	1	0	4F	1	1	1	1	0	1	0	0	2F
1	0_6	WRITE CYCLE	1	1	1	1	0	0	1	0	1	0	14F	1	1	1	1	0	1	1	0	6F	1	1	1	1	0	0	1	0	4F
1	0_7		1	1	1	1	0	1	1	0	1	0	16F	1	1	1	1	1	0	0	0	1F	1	1	1	1	0	1	1	0	6F
1	0_8		1	1	1	1	1	0	0	0	1	0	11F	1	1	1	0	1	0	0	0	17	1	1	1	1	1	0	0	0	1F
1	0_9		1	1	1	0	1	0	0	0	1	0	117	1	1	1	0	1	0	0	0	17	1	1	1	0	1	0	0	0	17
0	X		0	0	0	0	0	0	0	0	0	1	200	0	0	0	0	0	1	0	0	20	0	0	0	0	0	0	0	0	00
1	0_1		0	0	0	0	0	1	0	0	0	1	220	1	0	0	0	0	0	0	0	01	0	0	0	0	0	1	0	0	20
1	0_2		1	0	0	0	0	0	0	0	0	1	201	1	1	0	0	0	0	0	0	03	1	0	0	0	0	0	0	0	01
1	0_3		1	1	0	0	0	0	0	0	0	1	203	1	1	0	0	0	1	0	0	23	1	1	0	0	0	0	0	0	03
1	0_4		1	1	0	0	0	1	0	0	0	1	223	1	1	0	0	0	0	1	0	43	1	1	0	0	0	0	1	0	23
1	0_5		1	1	0	0	0	0	1	0	0	1	243	1	1	0	0	0	0	1	1	63	1	1	0	0	0	0	1	0	43
1	0_6		1	1	0	0	0	1	1	0	0	1	263	1	1	0	0	0	0	0	1	83	1	1	0	0	0	1	1	0	63
1	0_7		1	1	0	0	0	0	0	1	0	1	283	1	1	0	0	0	1	0	1	A3	1	1	0	0	0	0	0	1	83
1	0_8	READ INTERRUPT VECTOR CYCLE	1	1	0	0	0	1	0	1	0	1	2A3	1	1	0	0	0	0	1	1	C3	1	1	0	0	0	1	0	1	A3
1	0_9		1	1	0	0	0	0	1	1	0	1	2C3	1	1	0	0	0	1	1	1	E3	1	1	0	0	0	0	1	1	C3
1	0_{10}		1	1	0	0	0	1	1	1	0	1	2E3	1	1	0	1	0	0	0	0	0B	1	1	0	0	0	1	1	1	E3
1	0_{11}		1	1	0	1	0	0	0	0	0	1	20B	1	1	0	1	0	1	0	0	2B	1	1	0	1	0	0	0	0	0B
1	0_{12}		1	1	0	1	0	1	0	0	0	1	22B	1	1	0	1	1	0	0	0	1B	1	1	0	1	0	1	0	0	2B
1	0_{13}		1	1	0	1	1	0	0	0	0	1	21B	1	1	1	1	1	0	0	0	1F	1	1	0	1	1	0	0	0	1B

Figure 6.

The write cycle operates in a similar manner with two exceptions. First, there is a required data set-up time of 30ns with respect to the falling edge of $\overline{ZDS}$ (TsDW(DSF)). Therefore, one sequencer idle state is required between T_2 and T_4. Secondly, there is a required data hold time of 30ns with respect to the rising edge of $\overline{ZDS}$ (ThDW(DS)). Therefore, T_5 is asserted at 0_8, while T_4 is negated at 0_9. If bidirectional registers are substituted for the data transceivers, four wait states can be eliminated from the write cycle.

TIMING PARAMETERS

During a read interrupt vector cycle, timing parameters are the same as a normal read operation with the exception of the $\overline{ZAS}$ to $\overline{ZDS}$ delay. In a normal read, this parameter is 60ns. However, during an interrupt, this parameter (TdDCST) must be equal to or greater than the interrupt daisy chain settling time. When five Z8000 peripherals are in this chain, this parameter is approximately 710ns. To accommodate this requirement, 8 idle states must be inserted between T_2 and T_4 by the state sequencer. This state sequencer detects an interrupt operation via A_9 (reference Figure 6).

SYSTEM ARCHITECTURE

Figure 9 illustrates an architecture that allows the 68K to perform other tasks while an I/O operation is in progress. The ZI/O chip select decoder accepts:

1) CPU Status — indicates an interrupt read cycle is required. This activates RIV (Read Interrupt Vector).

2) R/W — indicates that the current cycle is a read or a write. This activates I/OR/$\overline{W}$.

3) MAn — A user-definable number of address lines (6 lines minimum) that identifies which peripheral is being addressed. This activates the appropriate $\overline{CS}$ line, ZI/O REQ, $\overline{ZINH}$ and 4/6MHz.

The ZI/O bus arbiter is used to determine who has I/O bus control and is only required if a DMA controller is part of the Z peripheral chain.

It accepts:

1) ZI/O REQ — indicates the CPU is requesting the I/O bus.

2) DMA BUS REQ — indicates the DMA controller is requesting the I/O bus.

3) M BUS REQ — allows the 68K to request the I/O bus before the I/O cycle is started.

and generates:

1) ZBUS ERR — flags the 68K when the CPU starts a ZI/O operation and the I/O bus is busy.

2) ZI/O BUS BUSY — A Z-bus status line that can be polled by the CPU.

3) DMA BUS ACQ — grants control of the ZI/O bus to the DMA controller.

4) $\overline{TOE}$ — grants control of the ZI/O bus to the Bus Translator.

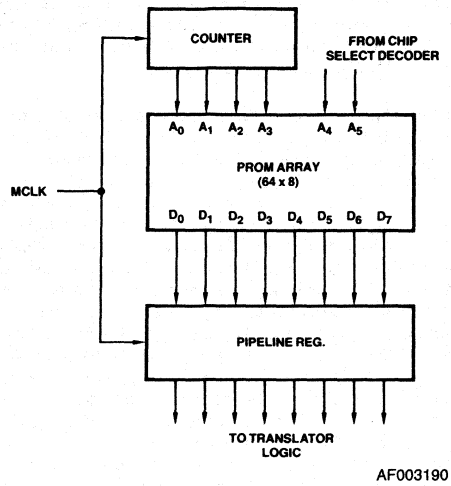


Figure 7.

This type of architecture is useful in I/O intensive applications where the data must be operated on concurrently by the CPU (i.e., PBX systems, disk controllers, etc.). Data is received and transmitted by this system via the DMA controller. Transmitted data is loaded into the Buffer Memory by the CPU and is then transmitted via a DMA operation. Received data is loaded into the Buffer Memory via a DMA operation and is then read by the CPU. In extreme cases this Buffer Memory can be dual ported so that true concurrent operation between CPU and I/O is achieved. In single-ported Buffer Memory architectures, a Bus Arbiter is required to grant ZI/O bus control to CPU or DMA controller. In this example, the CPU can poll the arbiter for ZI/O bus control and lock out the DMA controller until the CPU has completed its I/O operation. Alternately, the CPU can start a ZI/O bus operation, and if the bus is busy, a bus error is generated. The operation of this arbiter is straightforward and can be constructed using a minimal number of gates and flip-flops.

The I/O chip select decoder is a combination chip selector and memory mapper. Note that the 4MHz/6MHz signal is used by the bus translator to allow intermixing of slow and fast peripheral chips.

The DMA controller is an AmZ8016, which utilizes "link-lists" to update its control registers. This further reduces the need for CPU intervention, thus increasing system performance.

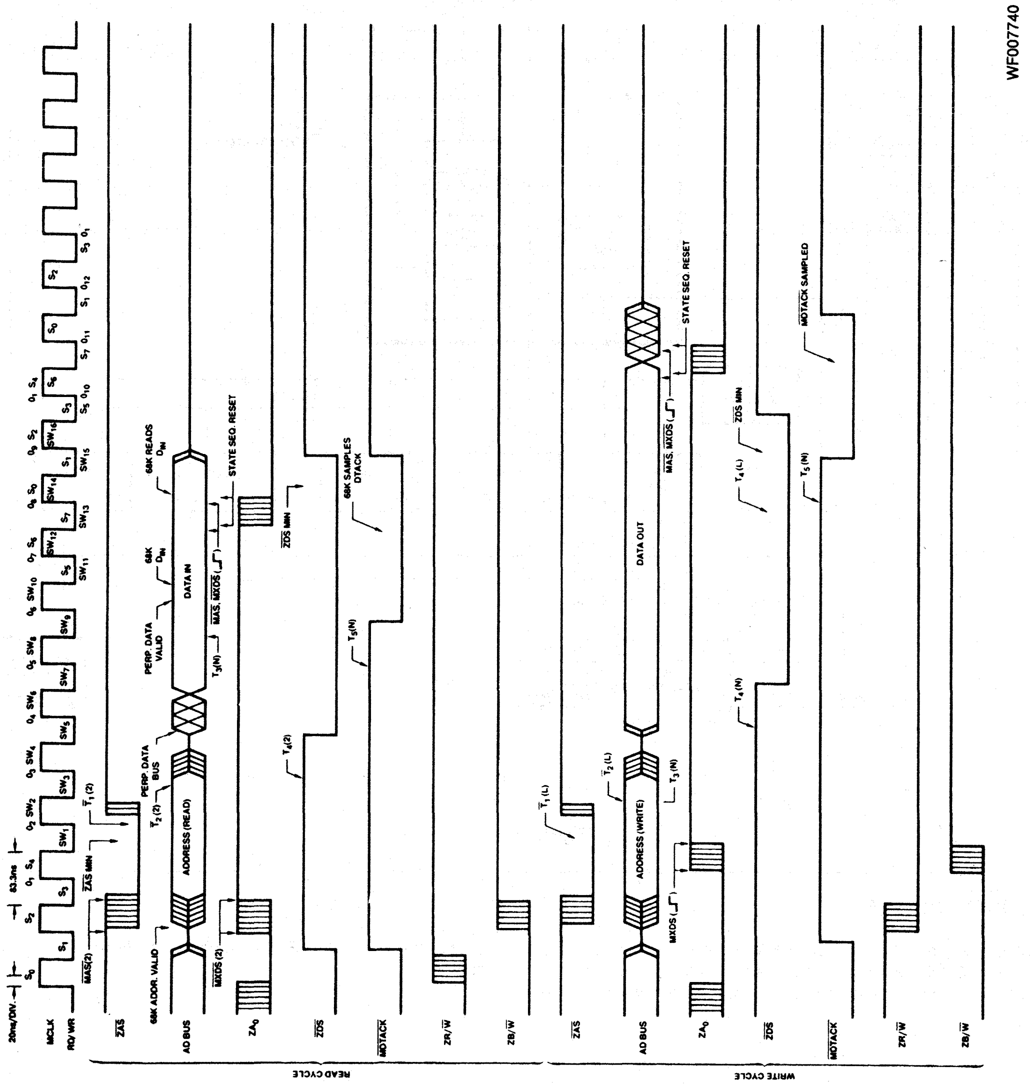


Figure 8. 12MHz 68000 to 4MHz Peripherals Timing Relationships

5

00971B

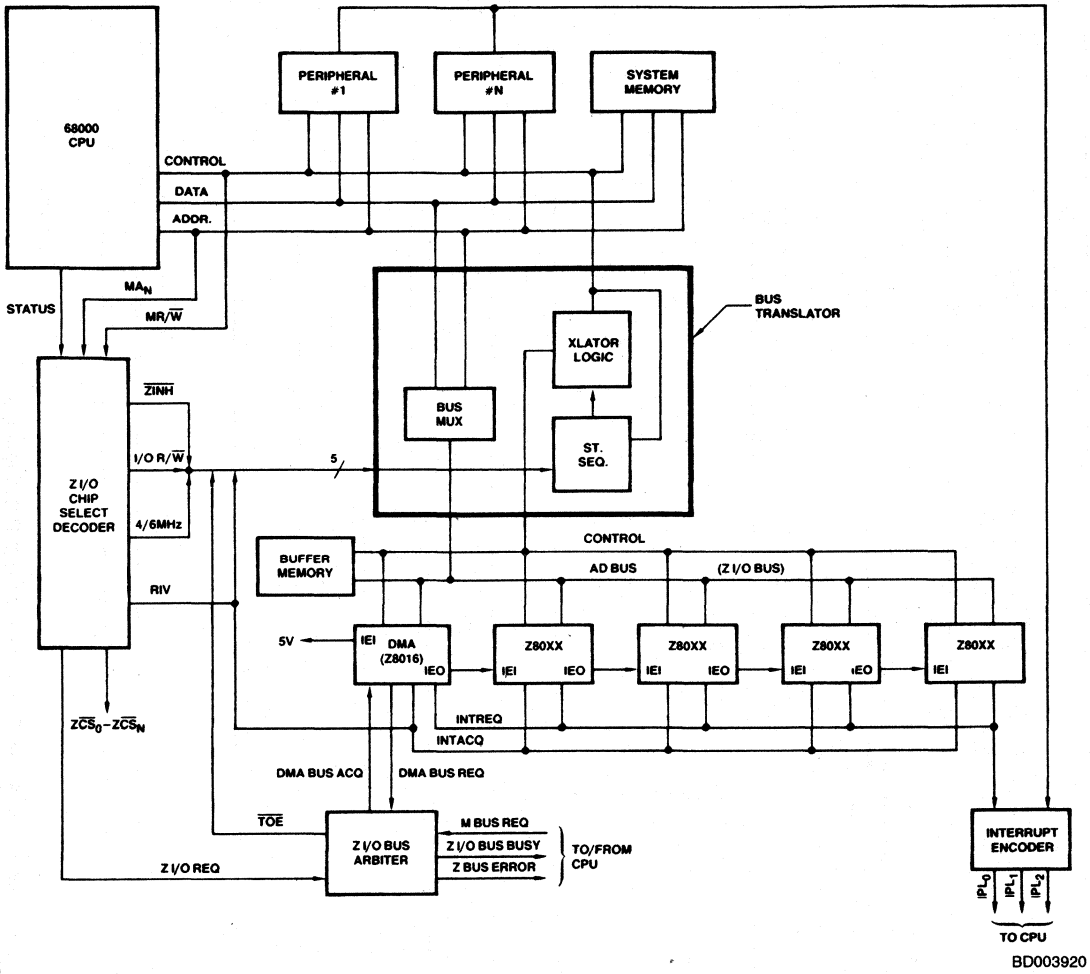


Figure 9.

CONCLUSIONS

This circuit is a highly efficient and flexible bus translator that allows Z8000 peripherals and a 68000 CPU to intercommunicate in a uniform manner; the same technique is also applicable to other CPU's, such as iAPX86, LSI-11 and N16000.

8-Bit Microprocessors Index

8080A/Am9080A

8-Bit Microprocessor

DISTINCTIVE CHARACTERISTICS

- High-speed version with 1μsec instruction cycle
- Military temperature range operation to 1.5μsec
- Ion-implanted, n-channel, silicon-gate MOS technology
- 3.2mA of output drive at 0.4V (two full TTL loads)
- 700mV of high, 400mV of low level noise immunity
- 820mW maximum power dissipation at ±5% power

GENERAL DESCRIPTION

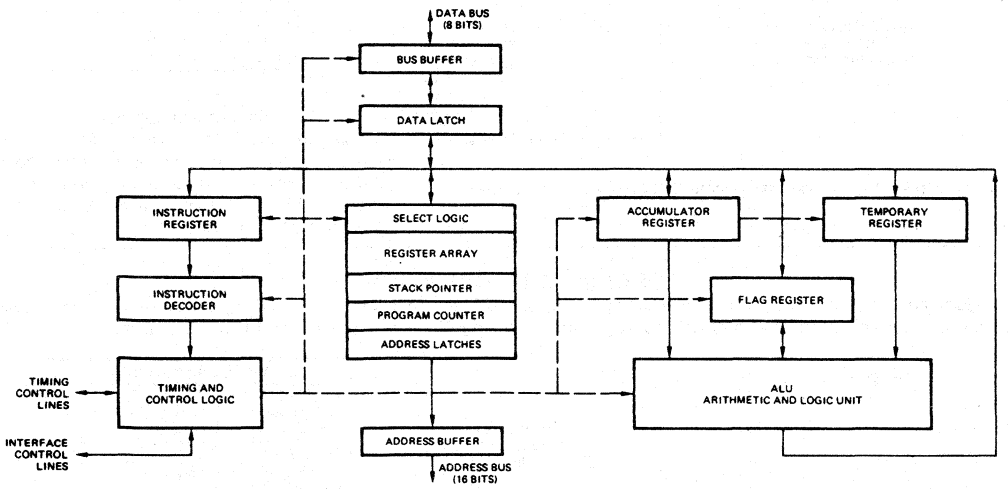
The 8080A products are complete, general-purpose, single-chip digital processors. They are fixed instruction set, parallel, 8-bit units fabricated with Advanced N-Channel Silicon Gate MOS technology. When combined with external memory and peripheral devices, powerful microcomputer systems are formed. The 8080A may be used to perform a wide variety of operations, ranging from complex arithmetic calculations to character handling to bit control. Several versions are available offering a range of performance options.

The processor has a 16-bit address bus that may be used to directly address up to 64K bytes of memory. The memory may be any combination of read/write and read-only. Data are transferred into or out of the processor on a bi-directional 8-bit data bus that is separate from the address lines. The data bus transfers instructions, data and status information between system devices. All transfers are handled using asynchronous handshaking controls so that any speed memory or I/O device is easily accommodated.

An accumulator plus six general registers are available to the programmer. The six registers are each 8 bits long and may be used singly or in pairs for both 8- and 16-bit operations. The accumulator forms the primary working register and is the destination for many of the arithmetic and logic operations.

A general purpose push-down stack is an important part of the processor architecture. The contents of the stack reside in R/W memory and the control logic, including a 16-bit stack pointer, is located on the processor chip. Subroutine call and return instructions automatically use the stack to store and retrieve the contents of the program counter. Push and Pop instructions allow direct use of the stack for storing operands, passing parameters and saving the machine state.

BLOCK DIAGRAM

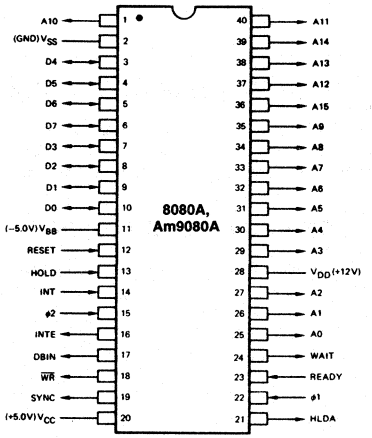


BD003800

04123A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40, P-40



CD005572

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

8080A

- Temperature Range: Blank = Commercial 0°C to 70°C; I = Industrial −40°C to 85°C; M = Military −55°C to 125°C
- 40-pin Package: P = Plastic Dip; D = Ceramic Dip
- Additional Processing: Blank = Std. Processing; B = Burn-in
- Speed (Will vary by device): Blank = 2MHz; −2 = 2.6MHz; −1 = 3.1MHz
- DEVICE*

*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations	
8080A-1 8080A-2 8080A 8080A-1B 8080A-2B 8080AB	P, D, ID
8080A 8080A-2 8080A-1	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am9080A

- Device Type
- Speed: Blank = 2MHz; −2 = 2.6MHz; −1 = 3.1MHz
- Screening Option: Blank = Std. Processing; B = Burn In
- Temperature: C = Commercial 0°C to 70°C; I = Industrial −40°C to 85°C; M = Military −55°C to 125°C
- 40-pin Package: D = Cerdip; P = Plastic

Valid Combinations	
Am9080A-1 Am9080A-2 Am9080A	PC, DC PCB, DCB DI, DIB
Am9080A Am9080A-2 Am9080A-1	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

TYPE	PINS	ABBREVIATION	SIGNAL
INPUT	1	V_{SS}	Ground
INPUT	3	V_{DD}, V_{CC}, V_{BB}	+12V, +5V, −5V Supplies
INPUT	2	ϕ_1, ϕ_2	Clocks
INPUT	1	RESET	Reset
INPUT	1	HOLD	Hold
INPUT	1	INT	Interrupt
INPUT	1	READY	Ready
IN/OUT	8	D_0-D_7	Data Bus
OUTPUT	16	A_0-A_{15}	Address
OUTPUT	1	INTE	Interrupt Enable
OUTPUT	1	DBIN	Data Bus In Control
OUTPUT	1	$\overline{WR}$	Write Not
OUTPUT	1	SYNC	Cycle Synchronization
OUTPUT	1	HLDA	Hold Acknowledge
OUTPUT	1	WAIT	Wait

Pin No.	Names	I/O	Description
22, 15	ϕ_1, ϕ_2	I	The Clock inputs provide basic timing generation for all internal operations. They are non-overlapping two phase, high level signals. All other inputs to the processor are TTL compatible.
12	RESET	I	The Reset input initializes the processor by clearing the program counter, the instruction register, the interrupt enable flip-flop and the hold acknowledge flip-flop. The Reset signal should be active for at least three clock periods. The general registers are not cleared.
13	HOLD	I	The Hold input allows an external signal to cause the processer to relinquish control over the address lines and the data bus. When Hold goes active, the processor completes its current operation, activates the HLDA output, and puts the 3-state address and data lines into their high-impedance state. The Holding device can then utilize the address and data busses without interference.
23	READY	I	The Ready input synchronizes the processor with external units. When Ready is absent, indicating the external operation is not complete, the processor will enter the Wait state. It will remain in the Wait state until the clock cycle, following the appearance of Ready.
14	INT	I	The Interrupt input signal provides a mechanism for external devices to modify the instruction flow of the program in progress. Interrupt requests are handled efficiently with the vectored interrupt procedure and the general purpose stack. Interrupt processing is described in more detail on the next page.
10-7, 3-6	D_0-D_7	I/O	The Data Bus is comprised of 8 bidirectional signal lines for transferring data, instructions and status information between the processor and all external units.
25-27, 29-35, 1, 40, 37-39, 36	A_0-A_{15}	O	The Address Bus is comprised of 16 output signal lines used to address memory and peripheral devices.
19	SYNC	O	The Sync output indicates the start of each processor cycle and the presence of processor status information on the data bus.
17	DBIN	O	The Data Bus In output signal indicates that the bidirectional data bus is in the input mode and incoming data may be gated onto the Data Bus.
24	WAIT	O	The Wait output indicates that the processor has entered the Wait state and is prepared to accept a Ready from the current external operation.
18	$\overline{WR}$	O	The Write output indicates the validity of output on the data bus during a write operation.
21	HLDA	O	The Hold Acknowledge output signal is a response to a Hold input. It indicates that processor activity has been suspended and the Address and Data Bus signals will enter their high-impedance state.
16	INTE	O	The Interrupt Enable output signal shows the status of the interrupt enable flip-flop, indicating whether or not the processor will accept interrupts.

8080A/Am9080A INSTRUCTION SET

The instructions executed by the 8080A are variable length and may be one, two or three bytes long. The length is determined by the nature of the operation being performed and the addressing mode being used.

The instruction summary shows the number of successive memory bytes occupied by each instruction, the number of clock cycles required for the execution of the instruction, the binary coding of the first byte of each instruction, the mnemonic coding used by assemblers and a brief description of each operation. Some branch-type instructions have two execution times depending on whether the conditional branch is taken or not. Some fields in the binary code are labeled with alphabetic abbreviations. That shown as vvv is the address pointer used in the one-byte Call instruction (RST). Those shown as ddd or sss designate destination and source register fields that may be filled as follows:

111 A register
000 B register
001 C register
010 D register
011 E register
100 H register
101 L register
110 Memory

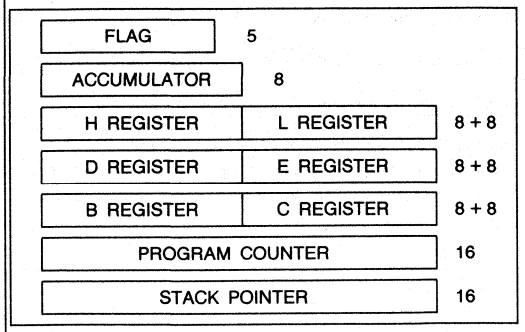
The register diagram shows the internal registers that are directly available to the programmer. The accumulator is the primary working register for the processor and is a specified or implied operand in many instructions. All I/O operations take place via the accumulator. Registers H, L, D, E, B and C may be used singly or in the indicated pairs. The H and L pair is the implied address pointer for many instructions.

The Flag register stores the program status bits used by the conditional branch instructions: carry, zero, sign and parity. The fifth flag bit is the intermediate carry bit. The flags and the accumulator can be stored on or retrieved from the stack with a single instruction. Bit positions in the flag register when pushed onto the stack (PUSH PSW) are:

7	6	5	4	3	2	1	0
S	Z	0	CY1	0	P	1	CY2

Where S = sign, Z = zero, CY1 = intermediate carry, P = parity, CY2 = carry.

REGISTER DIAGRAM



During Sync time at the beginning of each instruction cycle, the data bus contains operation status information that describes the machine cycle being executed. Positions for the status bits are:

7	6	5	4	3	2	1	0
MEMR	INP	M1	OUT	HLTA	STK	$\overline{WO}$	INTA

STATUS DEFINITION:

INTA Interrupt Acknowledge. Occurs in response to an Interrupt input and indicates that the processor will be ready for an interrupt instruction on the data bus when DBIN goes true.

$\overline{WO}$ Write or Output indicated when signal is LOW. When HIGH, a Read or Input will occur.

STK Stack indicates that the content of the stack pointer is on the address bus.

HLTA Halt Acknowledge.

OUT Output instruction is being executed.

M1 First instruction byte is being fetched.

INP Input instruction is being executed.

MEMR Memory Read operation.

INTERRUPT PROCESSING

When the processor interrupt mechanism is enabled (INTE = 1), interrupt signals from external devices will be recognized unless the processor is in the Hold State. In handling an interrupt, the processor will complete the execution of the current instruction, disable further interrupts and respond with INTA status instead of executing the next sequential instruction in the interrupted program.

The interrupting device should supply an instruction opcode to the processor during the next DBIN time after INTA status appears.

Any opcode may be used except XTHL. If the instruction supplied is a single byte instruction, it will be executed. (The usual single byte instruction utilized is RST.) If the interrupt instruction is two or three bytes long, the next one or two processor cycles, as indicated by the DBIN signal, should be used by the external device to supply the succeeding byte(s) of the interrupt instruction. Note that INTA status from the processor is not present during these operations.

If the interrupt instruction is not some form of CALL, it is executed normally by the processor except that the Program Counter is not incremented. The next instruction in the interrupted program is then fetched and executed. Notice that the interrupt mechanism must be re-enabled by the processor before another interrupt can occur.

If the interrupt instruction is some form of CALL, it is executed normally. The Program Counter is stored and control transferred to the interrupt service subroutine. The routine has responsibility for saving and restoring the machine state and for re-enabling interrupts if desired. When the interrupt service is complete, a RETURN instruction will transfer control back to the interrupted program.

INSTRUCTION SET SUMMARY

DATA TRANSFER

Op Code 7|6|5|4|3|2|1|0	No. of Bytes	Clock Cycles	Assembly Mnemonic	Instruction Description
0 1 d d d s s s	1	5	MOVr, r	Move register to register
0 1 1 1 0 s s s	1	7	MOVm, r	Move register to memory
0 1 d d d 1 1 0	1	7	MOVr, m	Move memory to register
0 0 d d d 1 1 0	2	7	MVI, r	Move to register, immediate
0 0 1 1 0 1 1 0	2	10	MVI, m	Move to memory, immediate
0 0 1 1 1 0 1 0	3	13	LDA	Load Acc, direct
0 0 0 0 1 0 1 0	1	7	LDAX B	Load Acc, indirect via B & C
0 0 0 1 1 0 1 0	1	7	LDAX D	Load Acc, indirect via D & E
0 0 1 0 1 0 1 0	3	16	LHLD	Load H & L, direct
0 0 1 0 0 0 0 1	3	10	LXI H	Load H & L, immediate
0 0 0 1 0 0 0 1	3	10	LXI D	Load D & E, immediate
0 0 0 0 0 0 0 1	3	10	LXI B	Load B & C, immediate
0 0 1 1 0 0 0 1	3	10	LXI SP	Load stack pointer, immediate
0 0 1 0 0 0 1 0	3	16	SHLD	Store H&L, direct
0 0 1 1 0 0 1 0	3	13	STA	Store Acc, direct
0 0 0 0 0 0 1 0	1	7	STAX B	Store Acc, indirect via B & C
0 0 0 1 0 0 1 0	1	7	STAX D	Store Acc, indirect via D & E
1 1 1 1 1 0 0 1	1	5	SPHL	Transfer H & L to stack pointer
1 1 1 0 1 0 1 1	1	4	XCHG	Exchange D & E with H & L
1 1 1 0 0 0 1 1	1	18	XTHL	Exchange top of stack with H & L
1 1 0 1 1 0 1 1	2	10	IN	Input to Acc
1 1 0 1 0 0 1 1	2	10	OUT	Output from Acc

CONTROL

0 1 1 1 0 1 1 0	1	7	HLT	Halt and enter wait state
0 0 1 1 0 1 1 1	1	4	STC	Set carry flag
0 0 1 1 1 1 1 1	1	4	CMC	Complement carry flag
1 1 1 1 1 0 1 1	1	4	EI	Enable interrupts
1 1 1 1 0 0 1 1	1	4	DI	Disable interrupts
0 0 0 0 0 0 0 0	1	4	NOP	No operation

BRANCH

1 1 0 0 0 0 1 1	3	10	JMP	Jump unconditionally
1 1 0 1 1 0 1 0	3	10	JC	Jump on carry
1 1 0 1 0 0 1 0	3	10	JNC	Jump on no carry
1 1 0 0 1 0 1 0	3	10	JZ	Jump on zero
1 1 0 0 0 0 1 0	3	10	JNZ	Jump on not zero
1 1 1 1 0 0 1 0	3	10	JP	Jump on positive
1 1 1 1 1 0 1 0	3	10	JM	Jump on minus
1 1 1 0 1 0 1 0	3	10	JPE	Jump on parity even
1 1 1 0 0 0 1 0	3	10	JPO	Jump on parity odd
1 1 0 0 1 1 0 1	3	17	CALL	Call unconditionally
1 1 0 1 1 1 0 0	3	17-11	CC	Call on carry
1 1 0 1 0 1 0 0	3	17-11	CNC	Call on no carry
1 1 0 0 1 1 0 0	3	17-11	CZ	Call on zero
1 1 0 0 0 1 0 0	3	17-11	CNZ	Call on not zero
1 1 1 1 0 1 0 0	3	17-11	CP	Call on positive
1 1 1 1 1 1 0 0	3	17-11	CM	Call on minus
1 1 1 0 1 1 0 0	3	17-11	CPE	Call on parity even
1 1 1 0 0 1 0 0	3	17-11	CPO	Call on parity odd
1 1 0 0 1 0 0 1	1	10	RET	Return unconditionally
1 1 0 1 1 0 0 0	1	11-5	RC	Return on carry
1 1 0 1 0 0 0 0	1	11-5	RNC	Return on no carry
1 1 0 0 1 0 0 0	1	11-5	RZ	Return on zero
1 1 0 0 0 0 0 0	1	11-5	RNZ	Return on not zero
1 1 1 1 0 0 0 0	1	11-5	RP	Return on positive
1 1 1 1 1 0 0 0	1	11-5	RM	Return on minus
1 1 1 0 1 0 0 0	1	11-5	RPE	Return on parity even
1 1 1 0 0 0 0 0	1	11-5	RPO	Return on parity odd
1 1 1 0 1 0 0 1	1	5	PCHL	Jump unconditionally, indirect via H & L
1 1 V V V 1 1 1	1	11	RST	Restart

ARITHMETIC

1 0 0 0 0 s s s	1	4	ADDr	Add register to Acc
1 0 0 0 1 s s s	1	4	ADCr	Add with carry register to Acc
1 0 0 0 0 1 1 0	1	7	ADDm	Add memory to Acc
1 0 0 0 1 1 1 0	1	7	ADCm	Add with carry memory to Acc
1 1 0 0 0 1 1 0	2	7	ADI	Add to Acc, immediate
1 1 0 0 1 1 1 0	2	7	ACI	Add with carry to Acc, immediate
0 0 0 0 1 0 0 1	1	10	DAD B	Double add B & C to H & L
0 0 0 1 1 0 0 1	1	10	DAD D	Double add D & E to H & L
0 0 1 0 1 0 0 1	1	10	DAD H	Double add H & L to H & L
0 0 1 1 1 0 0 1	1	10	DAD SP	Double add stack pointer to H & L
1 0 0 1 0 s s s	1	4	SUBr	Subtract register from Acc
1 0 0 1 1 s s s	1	4	SBBr	Subtract with borrow register from Acc
1 0 0 1 0 1 1 0	1	7	SUBm	Subtract memory from Acc
1 0 0 1 1 1 1 0	1	7	SBBm	Subtract with borrow memory from Acc
1 1 0 1 0 1 1 0	2	7	SUI	Subtract from Acc, immediate
1 1 0 1 1 1 1 0	2	7	SBI	Subtract with borrow from Acc, immediate
0 0 1 0 0 1 1 1	1	4	DAA	Decimal adjust Acc

STACK OPERATIONS

1 1 0 0 0 1 0 1	1	11	PUSH B	Push registers B & C on stack
1 1 0 1 0 1 0 1	1	11	PUSH D	Push registers D & E on stack
1 1 1 0 0 1 0 1	1	11	PUSH H	Push registers H & L on stack
1 1 1 1 0 1 0 1	1	11	PUSH PSW	Push Acc and flags on stack
1 1 0 0 0 0 0 1	1	10	POP B	Pop registers B & C off stack
1 1 0 1 0 0 0 1	1	10	POP D	Pop registers D & E off stack
1 1 1 0 0 0 0 1	1	10	POP H	Pop registers H & L off stack
1 1 1 1 0 0 0 1	1	10	POP PSW	Pop Acc and flags off stack

LOGICAL

1 0 1 0 0 s s s	1	4	ANA r	And register with Acc
1 0 1 0 0 1 1 0	1	7	ANA m	And memory with Acc
1 1 1 0 0 1 1 0	2	7	ANI	And with Acc, immediate
1 0 1 0 1 s s s	1	4	XRA r	Exclusive or register with Acc
1 0 1 0 1 1 1 0	1	7	XRA m	Exclusive Or memory with Acc
1 1 1 0 1 1 1 0	2	7	XRI	Exclusive Or with Acc, immediate
1 0 1 1 0 s s s	1	4	ORA r	Inclusive Or register with Acc
1 0 1 1 0 1 1 0	1	7	ORA m	Inclusive Or memory with Acc
1 1 1 1 0 1 1 0	2	7	ORI	Inclusive Or with Acc, immediate
1 0 1 1 1 s s s	1	4	CMP r	Compare register with Acc
1 0 1 1 1 1 1 0	1	7	CMP m	Compare memory with Acc
1 1 1 1 1 1 1 0	2	7	CPI	Compare with Acc, immediate
0 0 1 0 1 1 1 1	1	4	CMA	Compliment Acc
0 0 0 0 0 1 1 1	1	4	RLC	Rotate Acc left
0 0 0 0 1 1 1 1	1	4	RRC	Rotate Acc right
0 0 0 1 0 1 1 1	1	4	RAL	Rotate Acc left through carry
0 0 0 1 1 1 1 1	1	4	RAR	Rotate Acc right through carry

INCREMENT/DECREMENT

0 0 d d d 1 0 0	1	5	INR r	Increment register
0 0 1 1 0 1 0 0	1	10	INR m	Increment memory
0 0 0 0 0 0 1 1	1	5	INX B	Increment extended B & C
0 0 0 1 0 0 1 1	1	5	INX D	Increment extended D & E
0 0 1 0 0 0 1 1	1	5	INX H	Increment extended H & L
0 0 1 1 0 0 1 1	1	5	INX SP	Increment stack pointer
0 0 d d d 1 0 1	1	5	DCR r	Decrement register
0 0 1 1 0 1 0 1	1	10	DCR m	Decrement memory
0 0 0 0 1 0 1 1	1	5	DCX B	Decrement extended B & C
0 0 0 1 1 0 1 1	1	5	DCX D	Decrement extended D & E
0 0 1 0 1 0 1 1	1	5	DCX H	Decrement extended H & L
0 0 1 1 1 0 1 1	1	5	DCX SP	Decrement stack pointer

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
All Input or Output Voltages
With Respect to V_{BB} ..... −0.3V to +20V
V_{CC}, V_{DD} and V_{SS} With
Respect to V_{BB} ..... −0.3V to +20V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	V_{BB}	V_{DD}
8080A 8080A-2, 8080A-1 Am9080A Am9080A-2 Am9080A-1	0° to 70°C	5V ±5%	−5V ±5%	12V ±5%

Operating ranges define those limits over which the functionality of the device is guaranteed.

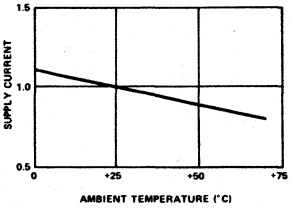
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{ILC}	Clock Input Low Voltage		$V_{SS}-1$		$V_{SS}+0.8$	V
V_{IHC}	Clock Input High Voltage		9.0		$V_{DD}+1$	V
V_{IL}	Input Low Voltage		$V_{SS}-1$		$V_{SS}+0.8$	V
V_{IH}	Input High Voltage		3.3		$V_{CC}+1$	V
V_{OL}	Output Low Voltage	I_{OL} = 1.9mA on all outputs,			0.45	V
V_{OH}	Output High Voltage	I_{OH} = −150μA.	3.7			V
$I_{DD(AV)}$	Avg. Power Supply Current (V_{DD})	Operation		40	70	mA
$I_{CC(AV)}$	Avg. Power Supply Current (V_{CC})	T_{CY} = .48μsec		60	80	mA
$I_{BB(AV)}$	Avg. Power Supply Current (V_{BB})			.01	1	mA
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$			±10	μA
I_{CL}	Clock Leakage	$V_{SS} \leq V_{CLOCK} \leq V_{DD}$			±10	μA
I_{DL} [2]	Data Bus Leakage in Input Mode	$V_{SS} \leq V_{IN} \leq V_{SS}+0.8V$ $V_{SS}+0.8V \leq V_{IN} \leq V_{CC}$			−100 −2.0	μA mA
I_{FL}	Address and Data Bus Leakage During HOLD	$V_{ADDR/DATA} = V_{CC}$ $V_{ADDR/DATA} = V_{SS}+0.45V$			+10 −100	μA

CAPACITANCE (T_A = 25°C, $V_{CC} = V_{DD} = V_{SS}$ = 0V, V_{BB} = −5V)

Parameters	Description	Test Conditions	Typ.	Max.	Units
C_ϕ	Clock Capacitance	f_c = 1 MHz	17	25	pf
C_{IN}	Input Capacitance	Unmeasured Pins	6	10	pf
C_{OUT}	Output Capacitance	Returned to V_{SS}	10	20	pf

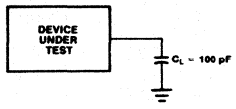
Notes: 1. The RESET signal must be active for a minimum of 3 clock cycles .
2. Δl supply / ΔT_A = −0.45%/°C.



OP001690

Typical Supply Current vs. Temperature, Normalized [3]

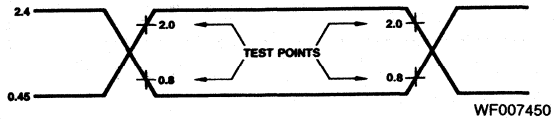
SWITCHING TEST LOAD CIRCUIT



TC001840

C_L = 100pF
C_L INCLUDES JIG CAPACITANCE

SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF007450

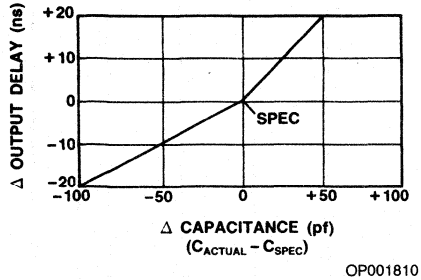
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	-1 Min	-1 Max	-2 Min	-2 Max	Unit
t_{CY} [3]	Clock Period		0.48	2.0	0.32	2.0	0.38	2.0	μsec
t_r, t_f	Clock Rise and Fall Time		0	50	0	25	0	50	nsec
$t_{\phi 1}$	ϕ_1 Pulse Width		60		50		60		nsec
$t_{\phi 2}$	ϕ_2 Pulse Width		220		145		175		nsec
t_{D1}	Delay ϕ_1 to ϕ_2		0		0		0		nsec
t_{D2}	Delay ϕ_2 to ϕ_1		70		60		70		nsec
t_{D3}	Delay ϕ_1 to ϕ_2 Leading Edges		80		60		70		nsec
t_{DA}	Address Output Delay From ϕ_2	C_L = 100pF		200		150		175	nsec
t_{DD}	Data Output Delay From ϕ_2	C_L = 100pF		220		180		200	nsec
t_{DC}	Signal Output Delay From ϕ_1 or ϕ_2 (SYNC, WR, WAIT, HLDA)	C_L = 50pF		120		110		120	nsec
t_{DF}	DBIN Delay From ϕ_2	C_L = 50pF	25	140	25	130	25	140	nsec
t_{DI} [1]	Delay for Input Bus to Enter Input Mode			t_{DF}		t_{DF}		t_{DF}	nsec
t_{DS1}	Data Set-up Time During ϕ_1 and DBIN		30		10		20		nsec
t_{DS2}	Data Set-up Time to ϕ_2 During DBIN		150		120		130		nsec
t_{DH} [1]	Data Hold time From $\phi 2$ During DBIN		[1]		[1]		[1]		nsec
t_{IE}	INTE Output Delay From ϕ_2	C_L = 50 pF		200		200		200	nsec
t_{RS}	READY Set-up Time During ϕ_2		120		90		90		nsec
t_{HS}	HOLD Set-up Time to ϕ_2		140		120		120		nsec
t_{IS}	INT Set-up Time During ϕ_2		120		100		100		nsec
t_H	Hold Time From ϕ_2 (READY, INT, HOLD)		0		0		0		nsec
t_{FD}	Delay to Float During Hold (Address and Data Bus)			120		120		120	nsec
t_{AW}	Address Stable Prior to WR	C_L = 100pF: Address, Data C_L = 50pF: $\overline{WR}$, HLDA, DBIN	[5]		[5]		[5]		nsec
t_{DW}	Output Data Stable Prior to WR		[6]		[6]		[6]		nsec
t_{WD}	Output Data Stable From WR		[7]		[7]		[7]		nsec
t_{WA}	Address Stable From WR		[7]		[7]		[7]		nsec
t_{HF}	HLDA to Float Delay		[8]		[8]		[8]		nsec
t_{WF}	WR to Float Delay		[9]		[9]		[9]		nsec
t_{AH}	Address Hold Time After DBIN during HLDA		-20		-20		-20		nsec

Notes: (Parenthesis gives -1, -2 specifications, respectively)

1. Data input should be enabled with DBIN status. No bus conflict can then occur and data hold time is assured. t_{DH} = 50 ns or t_{DF}, whichever is less.
2. $t_{CY} = t_{D3} + t_{r\phi 2} + t_{\phi 2} + t_{f\phi 2} + t_{D2} + t_{r\phi 1} \geq 480$ ns (-1:320 ns, -2:380 ns).

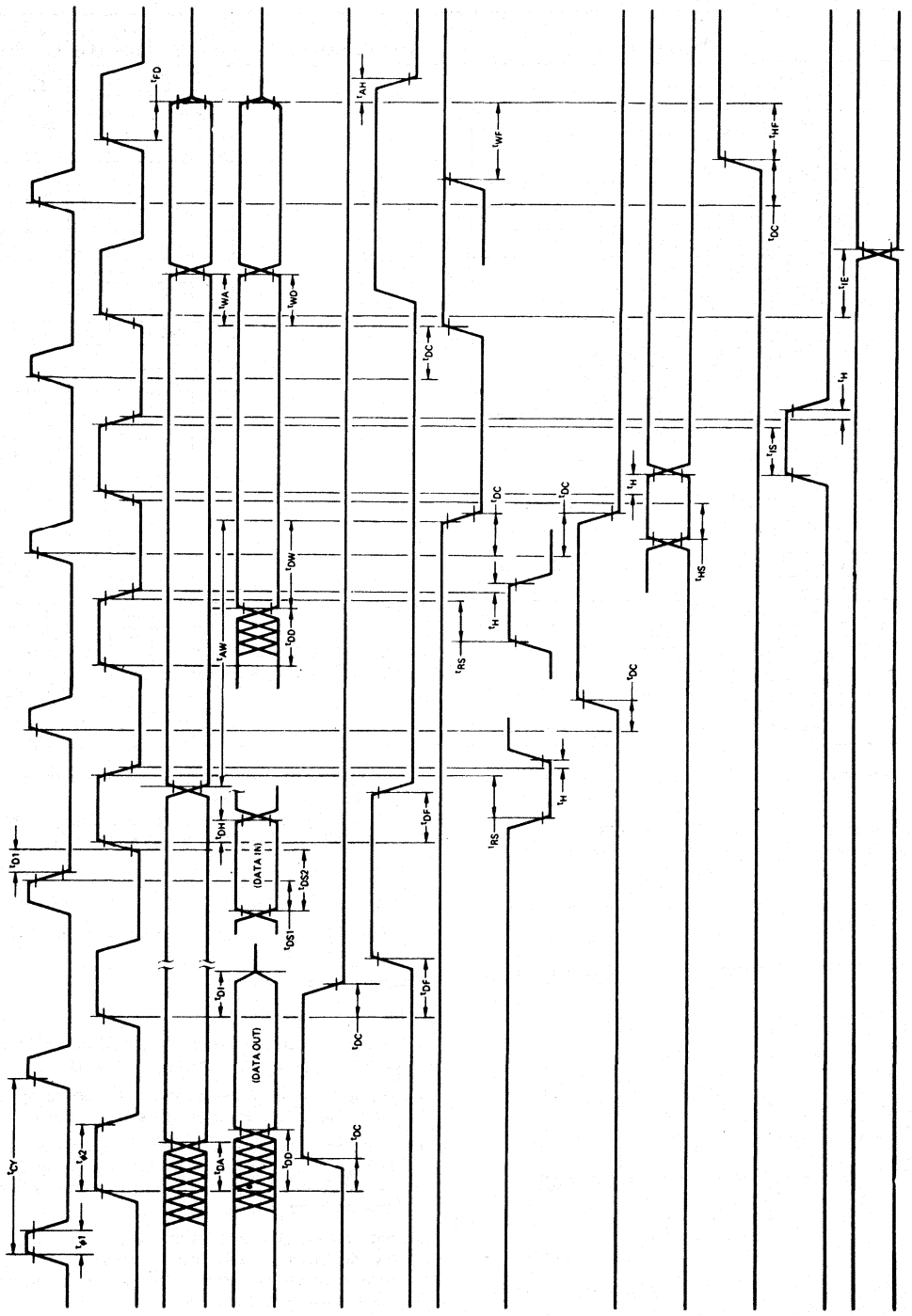
TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE



OP001810

3. The following are relevant when interfacing the 8080A to devices having V_{IH} = 3.3V:
   a) Maximum output rise time from .8V to 3.3V = 100ns @ C_L = SPEC.
   b) Output delay when measured to 3.0V = SPEC + 60ns @ C_L = SPEC.
   c) If C_L = SPEC, add .6ns/pF if $C_L > C_{SPEC}$, subtract .3ns/pF (from modified delay) if $C_L < C_{SPEC}$.
4. $t_{AW} = 2t_{CY} - t_{D3} - t_{r\phi 2} - 140$ ns (-1:110 ns, -2:130 ns).
5. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi 2} - 170$ ns (-1:150 ns, -2:170 ns).
6. If not HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi 2} + 10$ ns. If HLDA, $t_{WD} = t_{WA} = t_{WF}$.
7. $t_{HF} = t_{D3} + t_{r\phi 2} - 50$ ns).
8. $t_{WF} = t_{D3} + t_{r\phi 2} - 10$ ns.
9. Data in must be stable for this period during DBIN T_3. Both t_{DS1} and t_{DS2} must be satisfied.
10. Ready signal must be stable for this period during T_2 or T_W. (Must be externally synchronized.)
11. Hold signal must be stable for this period during T_2 or T_W when entering hold mode, and during T_3, T_4, T_5 and T_{WH} when in hold mode. (External synchronization is not required.)
12. Interrupt signal must be stable during this period of the last clock cycle of any instruction in order to be recognized on the following instruction. (External synchronization is not required.)
13. This timing diagram shows timing relationships only; it does not represent any specific machine cycle.

SWITCHING WAVEFORMS



WF007240

This chart presents relative timing waveform relationships and does not show actual processor operating cycles.

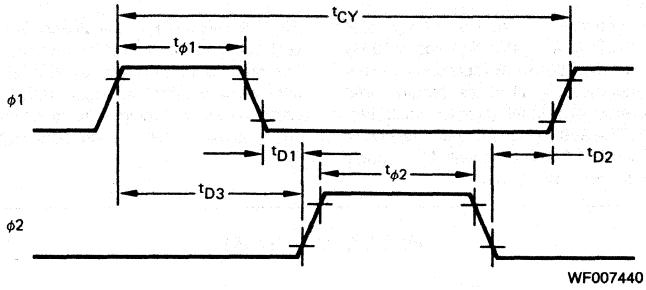
04123A

Refer to page 7-1 for Essential Information on Military Devices

CLOCK SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Am9080A-1, 8080A-1		Am9080A-2, 8080A-2		Am9080A, 8080A		Units
		Min	Max	Min	Max	Min	Max	
t_{CY}	Clock Period	320	2000	380	2000	480	2000	ns
t_r, t_f	Clock Transition Times	0	25	0	50	0	50	ns
$t_{\phi 1}$	Clock $\phi 1$ Pulse Width	50		60		60		ns
$t_{\phi 2}$	Clock $\phi 2$ Pulse Width	145		175		220		ns
t_{D1}	$\phi 1$ to $\phi 2$ Offset	0		0		0		ns
t_{D2}	$\phi 2$ to $\phi 1$ Offset	60		70		70		ns
t_{D3}	$\phi 1$ to $\phi 2$ Delay	60		70		80		ns

CLOCK WAVEFORM DETAIL



$$t_{CY} = t_{D3} + t_{r\phi 2} + t_{\phi 2} + t_{f\phi 2} + t_{D2} + t_{r\phi 1}$$

8085AH

8-Bit Microprocessor

DISTINCTIVE CHARACTERISTICS

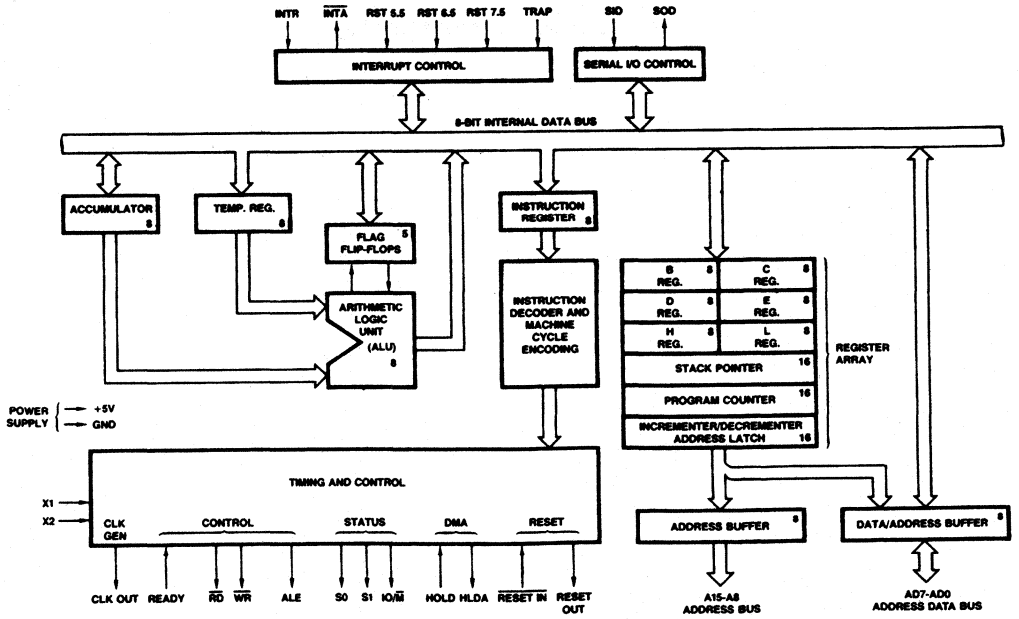
- Complete 8-bit parallel CPU
- On-chip system controller; advanced cycle status information available for large system control
- Four vectored interrupts (one is non-maskable)
- On-chip clock generator (with external crystal, LC or R/C network)
- Serial in/serial out port
- Decimal, binary and double precision arithmetic
- Direct addressing capability to 64K bytes of memory
- 1.3μs instruction cycle (8085AH)
- 0.8μs instruction cycle (8085AH-2)
- 100% software compatible with 8080A
- Single +5V power supply

GENERAL DESCRIPTION

The 8085AH is a new generation, complete 8-bit parallel central processing unit (CPU). Its instruction set is 100% software compatible with the 8080A microprocessor. Specifically, the 8085AH incorporates all of the features that the 8224 (clock generator) and 8228 (system controller) provided for the 8080A. The 8085AH-2 is a faster version of the 8085AH. The 8085AH is a 3MHz CPU with 10% supply tolerances and lower power consumption.

The 8085AH uses a multiplexed Data Bus. The address is split between the 8-bit address bus and the 8-bit data bus. The on-chip address latches of 8155H/56H memory products allow a direct interface with 8085AH. The 8085AH components, including various timing compatible support chips, allow system speed optimization.

BLOCK DIAGRAM

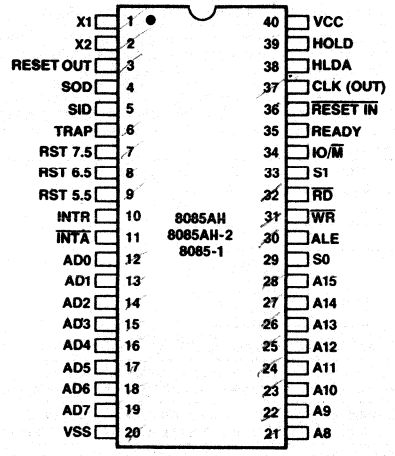


BD003790

04125A

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40, P-40



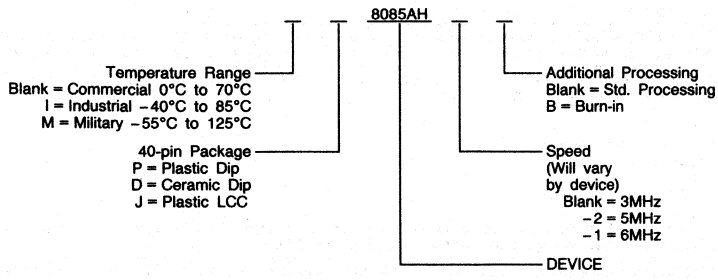
CD005563

Note: Pin 1 is marked for orientation

Figure 1.

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
8085A 8085AB 8085AH 8085AHB	P, D, ID
8085A-2 8085A-2B 8085AH-2 8085AH-2B 8085AH-1 8085AH-1B	P, D
8085AH	/BQA

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
21-28	A8-A15	O	Address Bus. The most significant eight bits of the memory address or the eight bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.
12-19	AD0-AD7	I/O	Multiplexed Address/Data Bus. Lower eight bits of the memory address (or I/O address), appears on the bus during the first clock cycle of a machine cycle. It then becomes the data bus during the second and third clock cycles. Three-stated during Hold and Halt modes.
30	ALE	O	Address Latch Enable. It occurs during the first clock cycle of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee set-up and hold times for the address information. The falling edge ALE can also be used to strobe the status information. ALE is never 3-stated.
29, 33	S0, S1	O	Data Bus Status. Encoded status of the bus cycle. S1 S0 0 0 HALT 0 1 WRITE 1 0 READ 1 1 FETCH S1 can be used as an advanced R/$\overline{W}$ status.
32	$\overline{RD}$	O	READ. A low level on $\overline{RD}$ indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer. Three-stated during Hold and Halt and during RESET.
31	$\overline{WR}$	O	WRITE. A low level on $\overline{WR}$ indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of $\overline{WR}$. Three-stated during Hold and Halt modes.
35	READY	I	If READY is HIGH during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is LOW, the CPU will wait an integral number of clock cycles for READY to go HIGH before completing the read or write cycle.
39	HOLD	I	HOLD. Indicates that another Master is requesting the use of the Address and Data Buses. The CPU, upon receiving the Hold request, will relinquish the use of buses as soon as the completion of the current machine cycle occurs. Internal processing can continue. The processor can regain the buses only after the Hold is removed. When the Hold is acknowledged, the Address, Data, $\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$ lines are three-stated.
38	HLDA	O	HOLD ACKNOWLEDGE. Indicates that the CPU has received the Hold request and that it will relinquish the buses in the next clock cycle. HLDA goes LOW after the Hold request is removed. The CPU takes the buses one half clock cycle after HLDA goes LOW.
10	INTR	I	INTERRUPT REQUEST. Is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of the instruction. If it is active, the Program Counter (PC) will be inhibited from incrementing and an $\overline{INTA}$ will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.
11	$\overline{INTA}$	O	INTERRUPT ACKNOWLEDGE. Is used instead of (and has the same timing as) $\overline{RD}$ during the Instruction cycle after an INTR is accepted. It can be used to activate the Am9519A Interrupt chip or some other interrupt port.
7-9	RST 7.5 RST 6.5 RST 5.5	I	RESTART INTERRUPTS. These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted. RST 7.5 → Highest Priority RST 6.5 RST 5.5 → Lowest Priority The priority of these interrupts is ordered as shown above. These interrupts have a higher priority than the INTR. However, they may be individually masked out using the SIM instructions.
6	TRAP	I	Trap interrupt is a non-maskable restart interrupt. It is recognized at the same time as INTR. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt.
36	$\overline{RESET\ IN}$	I	Reset sets the Program Counter to zero and resets the Interrupt Enable and HLDA flip-flops. None of the other flags or registers (except the instruction register) is affected. The CPU is held in the reset condition as long as $\overline{RESET}$ is applied.
3	RESET OUT	O	Indicates CPU is being reset. Can be used as a system RESET. The signal is synchronized to the processor clock.
1, 2	X1, X2	I	Crystal, LC or R/C network connections to set the internal clock generator. X1 can also be an external clock input instead of a crystal. The input frequency is divided by 2 to give the internal operating frequency.
37	CLK	O	Clock Output for use as a system clock when a crystal or R/C network is used as an input to the CPU. The period of CLK is twice the X1, X2 input period.
34	IO/$\overline{M}$	O	IO/$\overline{M}$ indicates whether the Read/Write is to memory or I/O. 3-stated during Hold and Halt Modes.
5	SID	I	Serial input data line. The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.
4	SOD	O	Serial output data line. The output SOD is set or reset as specified by the SIM instruction.
40	VCC		+5 volt supply.
20	VSS		Ground reference.

04125A

Refer to page 7-1 for Essential Information on Military Devices

DETAILED DESCRIPTION

The 8085AH is a complete 8-bit parallel central processor. It is designed with N-channel depletion loads and requires a single +5 volt supply. Its basic clock speed is 3MHz (5MHz: 8085AH-2/6MHz: 8085AH-1), thus improving on the present Am9080's performance with higher system speed. Also, it is designed to fit into a minimum system of three ICs: the CPU, a RAM/IO, and a ROM or PROM/IO chip.

The 8085AH uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first cycle, the address is sent out. The lower eight bits are latched into the peripherals by the Address Latch Enable (ALE). During the rest of the machine cycle, the Data Bus is used for memory or I/O data.

The 8085AH provides $\overline{RD}$, $\overline{WR}$ and IO/$\overline{Memory}$ signals for bus control. An Interrupt Acknowledge signal ($\overline{INTA}$) is also provided. Hold, Ready and all Interrupts are synchronized. The 8085AH also provides serial input data (SID) and serial output data (SOD) lines for simple serial interface.

In addition to these features, the 8085AH has three maskable, restart interrupts and one non-maskable trap interrupt.

8085AH vs. 8080A

The 8085AH includes the following features on-chip in addition to all of the Am9080A functions:

a. Internal clock generator
b. Clock output
c. Fully synchronized Ready
d. Schmitt action on $\overline{RESET\ IN}$
e. RESET OUT pin
f. $\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$ Bus Control Signals
g. Encoded Status information
h. Multiplexed Address and Data
i. Direct Restarts and non-maskable interrupt
j. Serial Input/Output lines

The internal clock generator requires an external crystal or R/C network. It will oscillate at twice the basic CPU operating frequency. A 50% duty cycle, two-phase, non-overlapping clock is generated from this oscillator internally, and one phase of the clock ($\phi 2$) is available as an external clock. The 8085AH directly provides the external RDY synchronization previously provided by the 8224. The $\overline{RESET\ IN}$ input is provided with a Schmitt action input so that power-on reset only requires a resistor and capacitor. RESET OUT is provided for System RESET.

The 8085AH provides $\overline{RD}$, $\overline{WR}$ and IO/$\overline{M}$ signals for Bus control. An $\overline{INTA}$ which was previously provided by the 8228 in Am9080A systems is also included in 8085AH.

Status Information

Status information is directly available from the 8085AH. ALE serves as a status strobe. The status is partially encoded and provides the user with advanced timing of the type of bus transfer being done. IO/$\overline{M}$ cycle status signal is provided directly also. Decoded S0, S1 carries the following status information:

MACHINE CYCLE STATUS

IO/$\overline{M}$	S1	S0	STATUS
0	0	1	Memory write
0	1	0	Memory read
1	0	1	I/O write
1	1	0	I/O read
0	1	1	Opcode fetch
1	1	1	Interrupt Acknowledge
*	0	0	Halt
*	X	X	Hold
*	X	X	Reset

* = 3-state (high-impedance)
X = unspecified

S1 can be interpreted as R/$\overline{W}$ in all bus transfers.

In the 8085AH the eight LSB of address are multiplexed with the data instead of status. The ALE line is used as a strobe to enter the lower half of the address into the memory or peripheral address latch. This also frees extra pins for expanded interrupt capability.

Interrupt and Serial I/O

The 8085AH/8085AH-2 has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5 and TRAP. INTR is identical in function to the 8080A INT. Each of three RESTART inputs, 5.5, 6.5, 7.5, has a programmable mask. TRAP is also a RESTART interrupt except it is non-maskable.

The three RESTART interrupts cause the internal execution of RST (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The nonmaskable TRAP causes the internal execution of a RST independent of the state of the interrupt enable or masks.

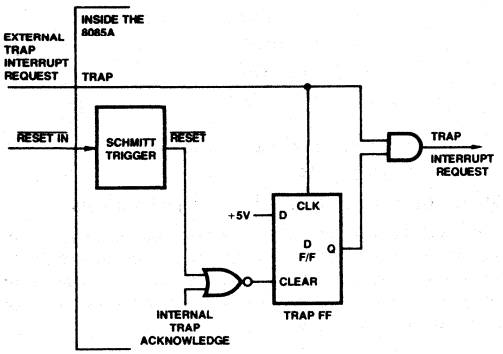
Name	RESTART Address (Hex)
TRAP	24_{16}
RST 5.5	$2C_{16}$
RST 6.5	34_{16}
RST 7.5	$3C_{16}$

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are high level-sensitive like INTR (and INT on the 8080A) and are recognized with the same timing as INTR. RST 7.5 is rising edge-sensitive. For RST 7.5, only a pulse is required to set an internal flip-flop which generates the internal interrupt request. The RST 7.5 request flip-flop remains set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a RESET IN to the 8085AH. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The status of the three RST interrupt masks can only be affected by the SIM instruction and RESET IN.

The interrupts are arranged in a fixed priority (that determines which interrupt is to be recognized if more than one is pending) as follows: TRAP - highest priority, RST 7.5, RST 6.5, RST 5.5, INTR – lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt a RST 7.5 routine if the interrupts were re-enabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic errors, such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority. It is not affected by any flag or mask. The TRAP input is both edge- and level-sensitive. The TRAP input must go HIGH and remain HIGH to be acknowledged, but will not be recognized again until it goes LOW, then HIGH again. This avoids any false triggering due to noise or logic glitches. The following diagram illustrates the TRAP interrupt request circuitry within the 8085AH.



AF003070

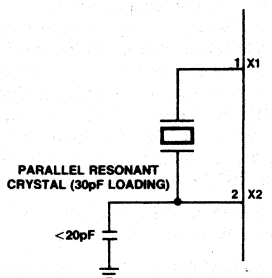
Note that the servicing of any interrupt (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR) disables all future interrupts (except TRAPs) until an EI instruction is executed.

The TRAP interrupt is special in that it preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instructions provide current interrupt enable status.

The serial I/O system is also controlled by the RIM and SIM instructions. SID is read by RIM, and SIM sets the SOD data.

Driving the X1 and X2 Inputs

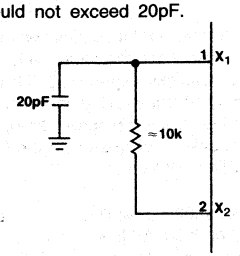
The user may drive the X1 and X2 inputs of the 8085AH or 8085AH-2 with a crystal, an external clock source or an R/C network as shown below. The driving frequency must be twice the desired internal operating frequency (the 8085AH would require a 6MHz crystal for 3MHz internal operation).



TC001960

1-6MHz Input Frequency

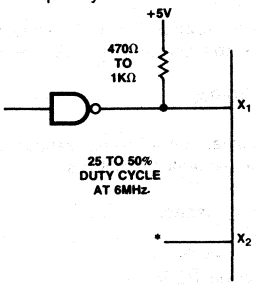
The 20pF capacitor is required to guarantee oscillation at the proper frequency during system start-up. Capacitance from X_2 to Ground should not exceed 20pF.



TC001980

≈3 MHz Input Frequency

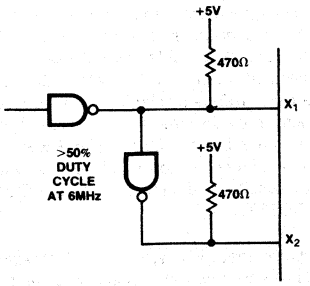
RC Mode causes a large drift in clock frequency because of the variation in on-chip timing generation parameters. Use of RC Mode should be limited to an application which can tolerate a wide frequency variation.



*X_2 Left Floating

TC001970

1-6 MHz Input Frequency



TC001990

≈6 MHz Input Frequency

Note: Duty cycle refers to the percentage of the clock input cycle when X_1 is high.

Figure 2. Driving the Clock Inputs (X1 and X2) of 8085AH

Generating 8085AH Wait State

The following circuit may be used to insert one WAIT state in each 8085AH machine cycle.

The D flip-flops should be chosen such that
- CLK is rising edge-triggered and
- CLEAR is low-level active.

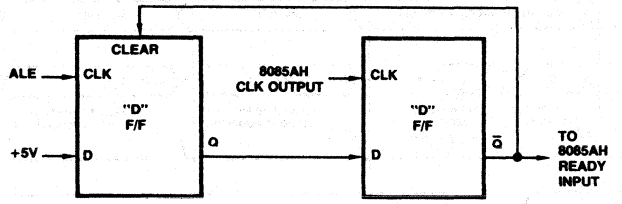


Figure 3. Generation of a Wait State for 8085AH CPU

Basic System Timing

The 8085AH has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8 bits of address on the Data Bus. Figure 4 shows an instruction fetch, memory read and I/O write cycle (OUT). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address.

As in the Am9080A, the READY line is used to extend the read and write pulse lengths so that the 8085AH can be used with slow memory. Hold causes the CPU to relinquish the bus when it is through with it by floating the Address and Data Buses.

Table 1. 8085AH Machine Cycle Chart

MACHINE CYCLE		STATUS			CONTROL		
		IO/$\overline{M}$	S1	S0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
OPCODE FETCH	(OF)	0	1	1	0	1	1
MEMORY READ	(MR)	0	1	0	0	1	1
MEMORY WRITE	(MW)	0	0	1	1	0	1
I/O READ	(IOR)	1	1	0	0	1	1
I/O WRITE	(IOW)	1	0	1	1	0	1
ACKNOWLEDGE OF INTR	(INA)	1	1	1	1	1	0
BUS IDLE	(BI): DAD	0	1	0	1	1	1
	ACK. OF RST, TRAP	1	1	1	1	1	1
	HALT	TS	0	0	TS	TS	1

Table 2. 8085AH Machine State Chart

Machine State	Status & Buses				Control		
	S1, S0	IO/$\overline{M}$	A_8–A_{15}	AD_0–AD_7	$\overline{RD}$, $\overline{WR}$	$\overline{INTA}$	ALE
T_1	X	X	X	X	1	1	1*
T_2	X	X	X	X	X	X	0
T_{WAIT}	X	X	X	X	X	X	0
T_3	X	X	X	X	X	X	0
T_4	1	0†	X	TS	1	1	0
T_5	1	0†	X	TS	1	1	0
T_6	1	0†	X	TS	1	1	0
T_{RESET}	X	TS	TS	TS	TS	1	0
T_{HALT}	0	TS	TS	TS	TS	1	0
T_{HOLD}	X	TS	TS	TS	TS	1	0

0 = Logic "0" TS = High Impedance
1 = Logic "1" X = Unspecified

*ALE not generated during 2nd and 3rd machine cycles of DAD instruction.
† IO/$\overline{M}$ = 1 during $T_4 - T_6$ of INA machine cycle.

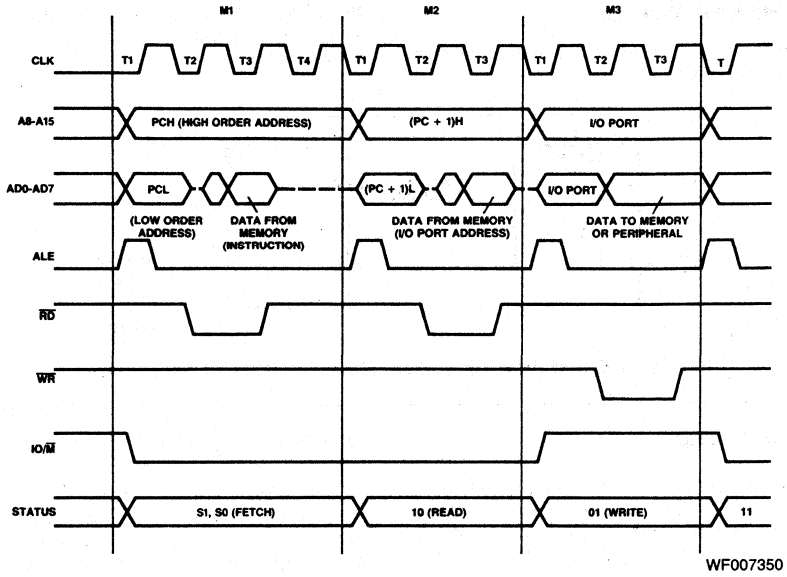


Figure 4. 8085AH Basic System Timing

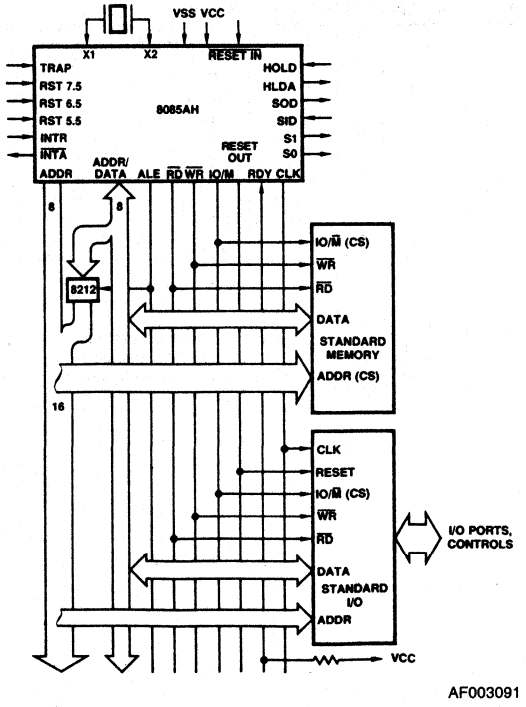


Figure 5. System Using Standard Memories

8085AH INSTRUCTION SET SUMMARY

Mnemonic*	Description	Instruction Code (Note 1) D7	D6	D5	D4	D3	D2	D1	D0	Clock Cycles (Note 2)
MOVE, LOAD AND STORE										
MOVr1r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV Mr	Move register to memory	0	1	1	1	0	S	S	S	7
MOV rM	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10

8085AH INSTRUCTION SET SUMMARY (Cont.)

Mnemonic*	Description	Instruction Code (Note 1) D7	D6	D5	D4	D3	D2	D1	D0	Clock Cycles (Note 2)
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or Immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
CONTROL										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupts	1	1	1	1	0	0	1	1	4
NOP	No operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
NEW 8085AH INSTRUCTIONS										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

Notes: 1. DDD or SSS: 8 = 000, C = 001, D = 010, E = 011, H = 100, L = 101, Memory = 110, A = 111.
2. Two possible cycle times (6/12) indicate instruction cycles dependent on condition flags.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... −0.5V to +7V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	I_{CC}
8085A 8085A-2	0°C to 70°C	5V ±5%	170mA
8085AH 8085AH-2	0°C to 70°C	5V ±10%	135mA
8085AH-1	0°C to 70°C	5V ±5%	200mA

Operating ranges define those limits over which the functionality of the device is guaranteed.

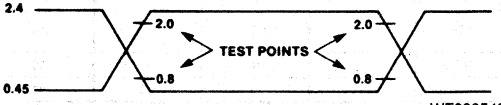
DC CHARACTERISTICS (8085A, 8085A-2) over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	V
V_{IH}	Input High Voltage	Except Pins 1 and 2	2.0	V_{CC} +0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current			170	mA
I_{IL}	Input Leakage	$0 \leq V_{IN} \leq V_{CC}$		±10	μA
I_{LO}	Output Leakage	$0.45V \leq V_{out} \leq V_{CC}$		±10	μA
V_{ILR}	Input Low Level, RESET		−0.5	+0.8	V
V_{IHR}	Input High Level, RESET		2.4	V_{CC} +0.5	V
V_{HY}	Hysteresis, RESET		0.25		V

DC CHARACTERISTICS (8085AH, 8085AH-2, 8085AH-1) over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage		−0.5	+0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} +0.5	V
V_{OL}	Output Low Voltage	I_{OL} = 2mA		0.45	V
V_{OH}	Output High Voltage	I_{OH} = −400μA	2.4		V
I_{CC}	Power Supply Current	8085AH, 8085AH-2		135	mA
		8085AH-1		200	mA
I_{IL}	Input Leakage	$0 \leq V_{IN} \leq V_{CC}$		±10	μA
I_{LO}	Output Leakage	$0.45V \leq V_{OUT} \leq V_{CC}$		±10	μA
V_{ILR}	Input Low Level, RESET		−0.5	+0.8	V
V_{IHR}	Input High Level, RESET		2.4	V_{CC} +0.5	V
V_{HY}	Hysteresis, RESET		0.25		V

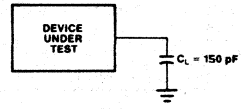
SWITCHING TEST INPUT/OUTPUT WAVEFORM



WF009540

A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0." TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0."

SWITCHING TEST LOAD CIRCUIT



TC001841

C_L = 150pF
C_L INCLUDES JIG CAPACITANCE

SWITCHING CHARACTERISTICS

Parameters	Description	8085A[2]		8085A-2[2]		Units
		Min	Max	Min	Max	
t_{CYC}	CLK Cycle Period	320	2000	200	2000	ns
t_1	CLK Low Time (Standard CLK Loading)	80		40		ns
t_2	CLK High Time (Standard CLK Loading)	120		70		ns
t_r, t_f	CLK Rise and Fall Time		30		30	ns
t_{XKR}	X_1 Rising to CLK Rising	30	120	30	100	ns
t_{XKF}	X_1 Rising to CLK Falling	30	150	30	110	ns
t_{AC}	A_{8-15} Valid to Leading Edge of Control[1]	270		115		ns
t_{ACL}	A_{0-7} Valid to Leading Edge of Control	240		115		ns
t_{AD}	A_{0-15} Valid to Valid Data In		575		350	ns
t_{AFR}	Address Float After Leading Edge of $\overline{READ}$ ($\overline{INTA}$)		0		0	ns
t_{AL}	A_{8-15} Valid Before Trailing Edge of ALE[1]	115		50		ns
t_{ALL}	A_{0-7} Valid Before Trailing Edge of ALE	90		50		ns
t_{ARY}	READY Valid from Address Valid		220		100	ns
t_{CA}	Address (A_{8-15}) Valid After Control	120		60		ns
t_{CC}	Width of Control Low ($\overline{RD}$, $\overline{WR}$, $\overline{INTA}$) Edge of ALE	400		230		ns
t_{CL}	Trailing Edge of Control to Leading Edge of ALE	50		25		ns
t_{DW}	Data Valid to Trailing Edge of $\overline{WRITE}$	420		230		ns
t_{HABE}	HLDA to Bus Enable		210		150	ns
t_{HABF}	Bus Float After HLDA		210		150	ns
t_{HACK}	HLDA Valid to Trailing Edge of CLK	110		40		ns
t_{HDH}	HOLD Hold Time	0		0		ns
t_{HDS}	HOLD Set-up Time to Trailing Edge of CLK	170		120		ns
t_{INH}	INTR Hold Time	0		0		ns
t_{INS}	INTR, RST, and TRAP Set-up Time to Falling Edge of CLK	160		150		ns
t_{LA}	Address Hold Time After ALE	100		50		ns
t_{LC}	Trailing Edge of ALE to Leading Edge of Control	130		60		ns
t_{LCK}	ALE Low During CLK High	100		50		ns
t_{LDR}	ALE to Valid Data During Read		460		270	ns
t_{LDW}	ALE to Valid Data During Write		200		120	ns
t_{LL}	ALE Width	140		80		ns
t_{LRY}	ALE to READY Stable		110		30	ns
t_{RAE}	Trailing Edge of $\overline{READ}$ to Re-Enabling of Address	150		90		ns
t_{RD}	$\overline{READ}$ (or $\overline{INTA}$) to Valid Data		300		150	ns
t_{RV}	Control Trailing Edge to Leading Edge of Next Control	400		220		ns
t_{RDH}	Data Hold Time After $\overline{READ}$ $\overline{INTA}$[7]	0		0		ns
t_{RYH}	READY Hold Time	0		0		ns
t_{RYS}	READY Set-up Time to Leading Edge of CLK	110		100		ns
t_{WD}	Data Valid After Trailing Edge of $\overline{WRITE}$	100		60		ns
t_{WDL}	LEADING Edge of WRITE to Data Valid		40		20	ns

Notes:
1. $A_8 - A_{15}$ address Specs apply to IO/$\overline{M}$, S_0, and S_1, except $A_8 - A_{15}$ are undefined during $T_4 - T_6$ of OF cycle; whereas, IO/$\overline{M}$, S_0, and S_1 are stable.
2. Test conditions: t_{CYC} = 320ns (8085A)/200ns (8085A-2); C_L = 150pF.
3. For all output timing where C_L = 150pF use the following correction factors:
   25pF $\leq C_L <$ 150pF: −0.10ns/pF
   150pF $< C_L \leq$ 300pF: +0.30ns/pF
4. Output timings are measured with purely capacitive load.
5. All timings are measured at output voltage V_L = 0.8V, V_H = 2.0V, and 1.5V with 20ns rise and fall time on inputs.
6. To calculate timing specifications at other values of t_{CYC} use Table 7.
7. Data hold time is guaranteed under all loading conditions.

SWITCHING CHARACTERISTICS

Parameter	Description	8085AH[2]		8085AH-2[2]		8085AH-1		Units
		Min	Max	Min	Max	Min	Max	
t_{CYC}	CLK Cycle Period	320	2000	200	2000	167	2000	ns
t_1	CLK Low Time (Standard CLK Loading)	80		40		20		ns
t_2	CLK High Time (Standard CLK Loading)	120		70		50		50
t_r, t_f	CLK Rise and Fall Time		30		30		30	ns
t_{XKR}	X_1 Rising to CLK Rising	20	120	20	100	20	100	ns
t_{XKF}	X_1 Rising to CLK Falling	20	150	20	110	20	110	ns
t_{AC}	A_{8-15} Valid to Leading Edge of Control[1]	270		115		70		ns
t_{ACL}	A_{0-7} Valid to Leading Edge of Control	240		115		60		ns
t_{AD}	A_{0-15} Valid to Valid Data In		575		350		225	ns
t_{AFR}	Address Float After Leading Edge of $\overline{READ}$ ($\overline{INTA}$)		0		0		0	ns
t_{AL}	A_{8-15} Valid Before Trailing Edge of ALE[1]	115		50		25		ns
t_{ALL}	A_{0-7} Valid Before Trailing Edge of ALE	90		50		25		ns
t_{ARY}	READY Valid from Address Valid		220		100		40	ns
t_{CA}	Address (A_{8-15}) Valid After Control	120		60		30		ns
t_{CC}	Width of Control Low ($\overline{RD}$, $\overline{WR}$, $\overline{INTA}$) Edge of ALE	400		230		150		ns
t_{CL}	Trailing Edge of Control to Leading Edge of ALE	50		25		0		ns
t_{DW}	Data Valid to Trailing Edge of $\overline{WRITE}$	420		230		140		ns
t_{HABE}	HLDA to Bus Enable		210		150		150	ns
t_{HABF}	Bus Float After HLDA		210		150		150	ns
t_{HACK}	HLDA Valid to Trailing Edge of CLK	110		40		0		ns
t_{HDH}	HOLD Hold Time	0		0		0		ns
t_{HDS}	HOLD Set-up Time to Trailing Edge of CLK	170		120		120		ns
t_{INH}	INTR Hold Time	0		0		0		ns
t_{INS}	INTR, RST, and TRAP Set-up Time to Falling Edge of CLK	160		150		150		ns
t_{LA}	Address Hold Time After ALE	100		50		20		ns
t_{LC}	Trailing Edge of ALE to Leading Edge of Control	130		60		25		ns
t_{LCK}	ALE Low During CLK High	100		50		15		ns
t_{LDR}	ALE to Valid Data During Read		460		270		175	ns
t_{LDW}	ALE to Valid Data During Write		200		140		110	ns
t_{LL}	ALE Width	140		80		50		ns
t_{LRY}	ALE to READY Stable		110		30		10	ns
t_{RAE}	Trailing Edge of $\overline{READ}$ to Re-Enabling of Address	150		90		50		ns
t_{RD}	$\overline{READ}$ (or $\overline{INTA}$) to Valid Data		300		150		75	ns
t_{RV}	Control Trailing Edge to Leading Edge of Next Control	400		220		160		ns
t_{RDH}	Data Hold Time After $\overline{READ}$ $\overline{INTA}$	0		0		0		ns
t_{RYH}	READY Hold Time	0		0		5		ns
t_{RYS}	READY Set-up Time to Leading Edge of CLK	110		100		100		ns
t_{WD}	Data Valid After Trailing Edge of $\overline{WRITE}$	100		60		30		ns
t_{WDL}	LEADING Edge of WRITE to Data Valid		40		20		30	ns

Notes: 1. A_8-A_{15} Address Specs apply to IO/$\overline{M}$, S_0, and S_1, except A8-A15 are undefined during T_4-T_6 OF cycle; whereas, IO/$\overline{M}$, S_0, and S_1 are stable.

2. *Test Conditions:* t_{CYC} = 320ns (8085AH)/200ns (8085AH-2);/167ns (8085AH-1); C_L = 150pF.

3. For all output timing where $C_L \neq 150$pF use the following correction factors:
   25pF $\leq C_L <$ 150pF: −0.10ns/pF
   150pF $< C_L \leq$ 300pF: +0.30ns/pF

4. Output timings are measured with purely capacitive load.

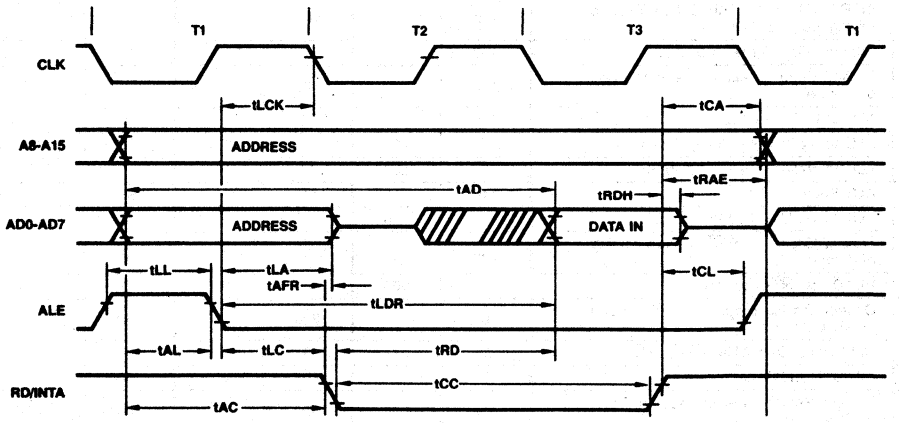
5. To calculate timing specifications at other values of t_{CYC} use Table 3.

Table 3. Bus Timing Specification as a T_{CYC} Dependent

Symbol	8085AH, 8085A	8085AH-2, 8085A-2	8085AH-1	
t_{AL}	(1/2) T – 45	(1/2) T – 50	(1/2) T – 58	Minimum
t_{LA}	(1/2) T – 60	(1/2) T – 50	(1/2) T – 63	Minimum
t_{LL}	(1/2) T – 20	(1/2) T – 20	(1/2) T – 33	Minimum
t_{LCK}	(1/2) T – 60	(1/2) T – 50	(1/2) T – 68	Minimum
t_{LC}	(1/2) T – 30	(1/2) T – 40	(1/2) T – 58	Minimum
t_{AD}	(5/2 + N) T – 225	(5/2 + N) T – 150	(5/2 + N) T – 192	Maximum
t_{RD}	(3/2 + N) T – 180	(3/2 + N) T – 150	(3/2 + N) T – 175	Maximum
t_{RAE}	(1/2) T – 10	(1/2) T – 10	(1/2) T – 33	Minimum
t_{CA}	(1/2) T – 40	(1/2) T – 40	(1/2) T – 53	Minimum
t_{DW}	(3/2 + N) T – 60	(3/2 + N) T – 70	(3/2 + N) T – 110	Minimum
t_{WD}	(1/2) T – 60	(1/2) T – 40	(1/2) T – 53	Minimum
t_{CC}	(3/2 + N) T – 80	(3/2 + N) T – 70	(3/2 + N) T – 100	Minimum
t_{CL}	(1/2) T – 110	(1/2) T – 75	(1/2) T – 83	Minimum
t_{ARY}	(3/2) T – 260	(3/2) T – 200	(3/2) T – 210	Maximum
t_{HACK}	(1/2) T – 50	(1/2) T – 60	(1/2) T – 83	Minimum
t_{HABF}	(1/2) T + 50	(1/2) T + 50	(1/2) T + 67	Maximum
t_{HABE}	(1/2) T + 50	(1/2) T + 50	(1/2) T + 67	Maximum
t_{AC}	(2/2) T – 50	(2/2) T – 85	(2/2) T – 97	Minimum
t_1	(1/2) T – 80	(1/2) T – 60	(1/2) T – 63	Minimum
t_2	(1/2) T – 40	(1/2) T – 30	(1/2) T – 33	Minimum
t_{RV}	(3/2) T – 80	(3/2) T – 80	(3/2) T – 90	Minimum
t_{LDR}	(4/2) T – 180	(4/2) T – 130	(4/2) T – 159	Maximum

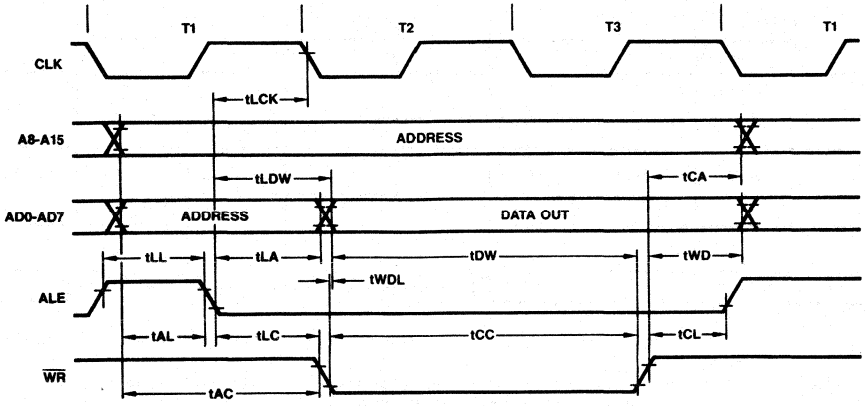
Note: N is equal to the total WAIT states. $T = t_{CYC}$.

READ OPERATION

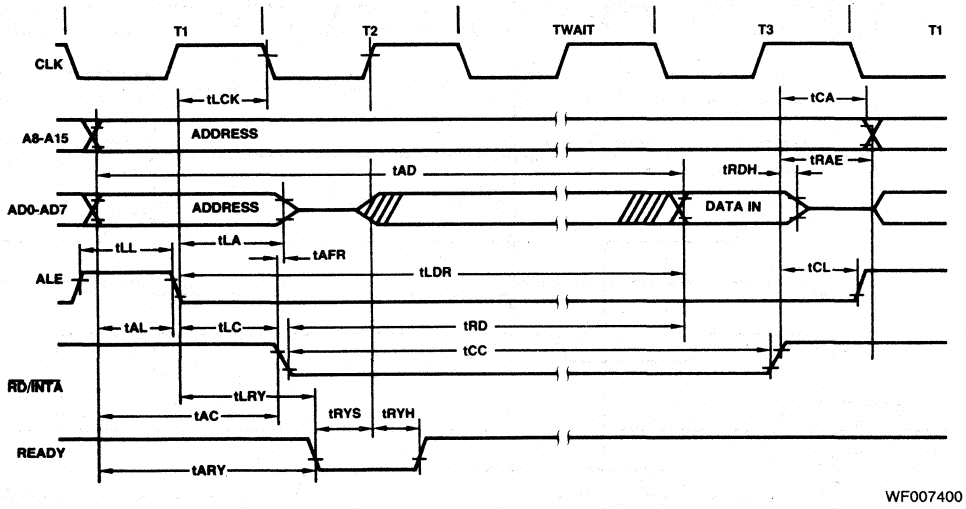


WF007380

WRITE OPERATION



TYPICAL READ OPERATION WITH WAIT CYCLE



Same READY timing applies to WRITE operation.

Figure 6. 8085AH/8085AH-2 Bus Timing

HOLD OPERATION

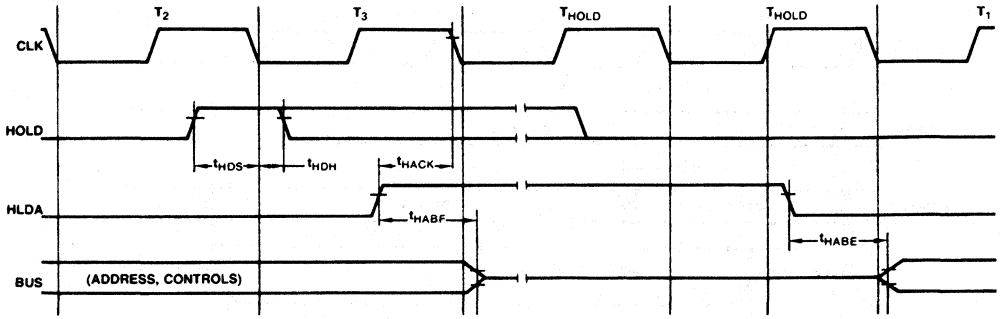


Figure 7. 8085AH Hold Timing

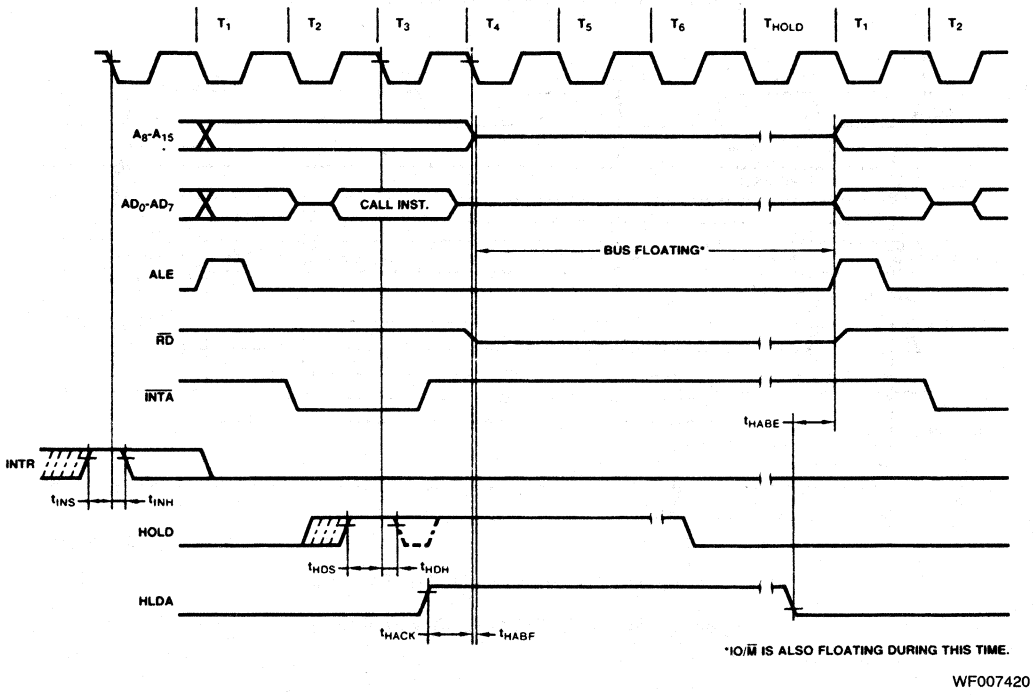


Figure 8. 8085AH Interrupt and Hold Timing

8155H/8156H

2048-Bit Static MOS RAM With I/O Ports and Timer

DISTINCTIVE CHARACTERISTICS

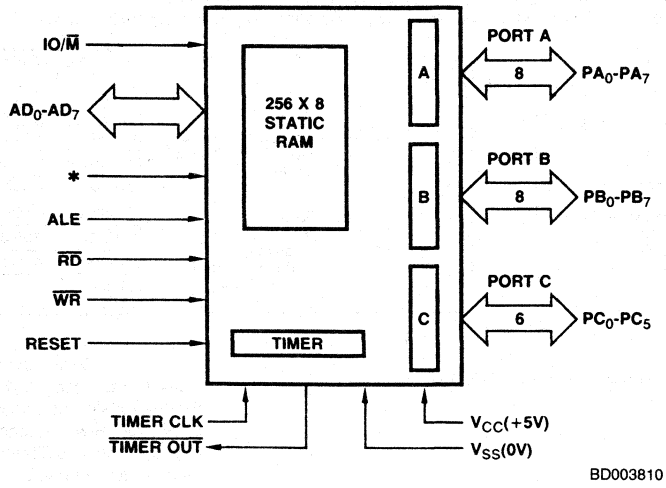
- 256 word x 8-bits
- Single +5V power supply
- Completely static operation
- Internal address latch
- 2 programmable 8-bit I/O ports
- 1 programmable 6-bit I/O port
- Programmable 14-bit binary counter/timer
- Multiplexed address and data bus

GENERAL DESCRIPTION

The 8155H and 8156H are RAM and I/O chips to be used in the 8085AH MPU system. The RAM portion is designed with 2K bit static cells organized as 256 x 8. They have a maximum access time of 400ns to permit use with no wait states in 8085AH CPU. The 8155H-2 and 8156H-2 have maximum access times of 330ns for use with the 8085AH. The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.

BLOCK DIAGRAM



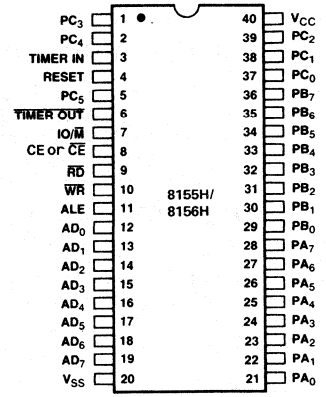
BD003810

*8155H = $\overline{CE}$, 8156H = CE

00934B

Refer to page 7-1 for Essential Information on Military Devices

CONNECTION DIAGRAM
Top View
D-40-1
P-40-1

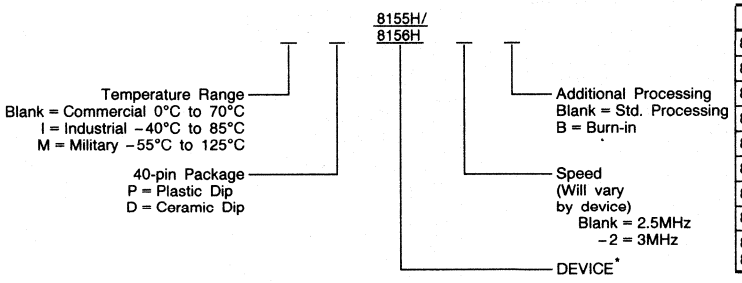


CD005581

Note: Pin 1 is marked for orientation

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations		
8155	8156	P, D, ID
8155H	8156H	
8155B	8156B	
8155HB	8156HB	
8155-2	8156-2	
8155H-2	8156H-2	
8155-2B	8156-2B	
8155H-2B	8156H-2B	
8155 8156		/BQA

*A "C" in the middle of the device type denotes CMOS version of the product.

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

00934B

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	RESET	I	The Reset signal is a pulse provided by the 8085AH to initialize the system. Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be 600ns. (Two 8085AH clock cycle times).
12-19	AD_0-AD_7	I/O	These are 3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch on the falling edge of the ALE. The address can be either for the memory section or the I/O section depending on the polarity of the IO/$\overline{M}$ input signal. The 8-bit data is either written into the chip or read from the chip depending on the status of $\overline{WRITE}$ or $\overline{READ}$ input signal.
8	CE OR $\overline{CE}$	I	Chip Enable: On the 8155H, this pin is $\overline{CE}$ and is active low. On the 8156H, this pin is CE and is active high.
9	$\overline{RD}$	I	Input low on this line with the Chip Enable active enables the AD_{0-7} buffers. If IO/$\overline{M}$ pin is LOW, the RAM content will be read out to the AD bus. Otherwise, the content of the selected I/O port will be read to the AD bus.
10	$\overline{WR}$	I	Input low on this line with the Chip Enable active causes the data on the AD lines to be written to the RAM or I/O ports, depending on the polarity of IO/$\overline{M}$.
11	ALE	I	Address Latch Enable: This control signal latches the address on the AD_{0-7} lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.
7	IO/$\overline{M}$	I	IO/$\overline{Memory}$ Select: This line selects the memory if LOW and selects the IO if HIGH.
21-28	PA_0-PA_7	I/O	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command/ Status Register.
29-36	PB_0-PB_7	I/O	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command/ Status Register.
37-39, 1, 2, 5	PC_0-PC_5	I/O	These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the C/S Register. When PC_{0-5} are used as control signals, they will provide the following: PC_0-A INTR (Port A Interrupt) PC_1-A $\overline{BF}$ (Port A Buffer Full) PC_2-$\overline{A\ STB}$ (Port A Strobe) PC_3-B INTR (Port B Interrupt) PC_4-$\overline{B\ BF}$ (Port B Buffer Full) PC_5-$\overline{B\ STB}$ (Port B Strobe)
3	TIMER IN	I	This is the input to the counter timer.
6	$\overline{TIMER\ OUT}$	O	This pin is the timer output. This output can be either a square wave or a pulse depending on the timer mode.
40	V_{CC}		+5 volt supply.
20	V_{SS}		Ground reference.

00934B

Refer to page 7-1 for Essential Information on Military Devices

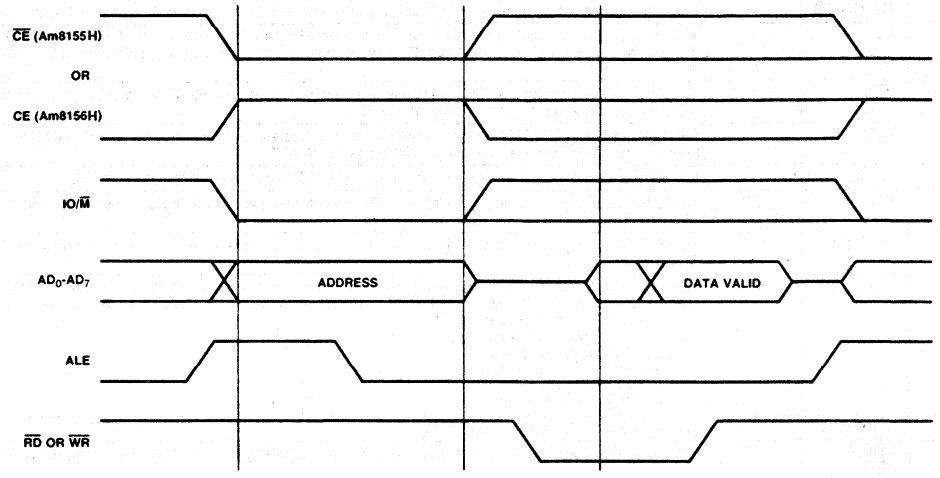
DETAILED DESCRIPTION

The 8155H/8156H includes the following operational features:

- 2K Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA and PB) and one 6-bit I/O port (PC)
- 14-bit down counter

The I/O portion contains four registers (Command/Status, PA_{0-7}, PB_{0-7}, PC_{0-5}). The IO/$\overline{M}$ (IO/$\overline{\text{Memory}}$ Select) pin selects the I/O or the memory (RAM) portion. Detailed descriptions of memory, I/O ports and timer functions will follow.

The 8-bit address on the AD lines, the Chip Enable input, and IO/$\overline{M}$ are all latched on chip at the falling edge of ALE. A LOW on the IO/$\overline{M}$ must be provided to select the memory section.



Note: For detailed timing diagram information, see Figure 7 and Switching Characteristics.

Figure 1. Memory Read/Write Cycle

PROGRAMMING INFORMATION

The Command/Status Register

The command register consists of eight latches, one for each bit. Four bits (0 – 3) define the mode of the ports. Two bits (4 – 5) enable or disable the interrupt from Port C when it acts as control port, and the last two bits (6 – 7) are for the timer.

The C/S register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation. The meaning of each bit of the command byte is defined as follows:

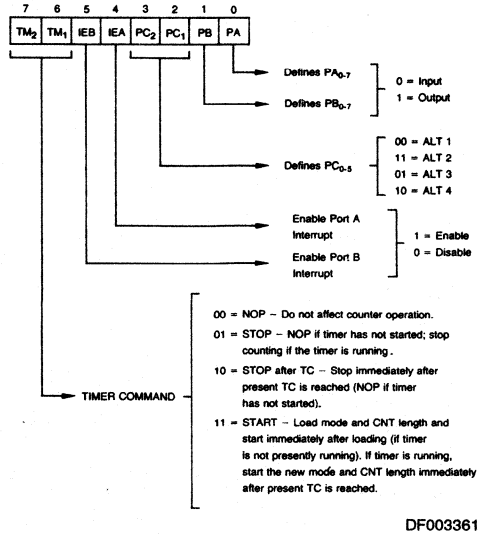


Figure 2. Command/Status Register Bit Assignment

Reading the Command/Status Register

The status register consists of seven latches, one for each bit: six (0 – 5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the C/S Register (Address XXXXX000). Status word format is shown below:

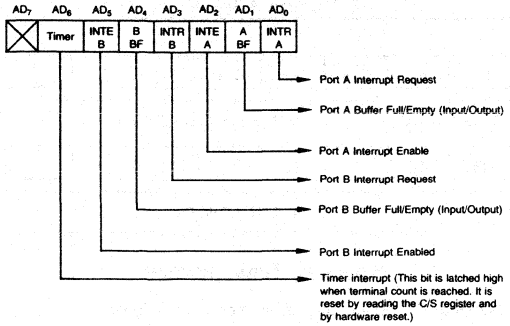


Figure 3. Command/Status Register Status Word Format

Input/Output Section

The I/O section of the 8155H/56H consists of four registers as described below.

- Command/Status Register (C/S) – This register is assigned the address XXXXX000. The C/S address serves a dual purpose.
  When the C/S register is selected during WRITE operation, a command is written into the command register. The contents of this register are not accessible through the pins.
  When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD_{0-7} lines.
- PA Register – This register can be programmed to be either input or output ports, depending on the status of the contents of the C/S Register. Also, depending on the command, this port can operate in either the basic mode or the strobed mode (see timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.
- PB Register – This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.
- PC Register – This register has the address XXXXX011 and contains only 6 bits. The 6 bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD_2 and AD_3 bits of the C/S register.
  When PC_{0-5} is used as a control port, 3 bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155H sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. See Table 1.

When the "C' port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

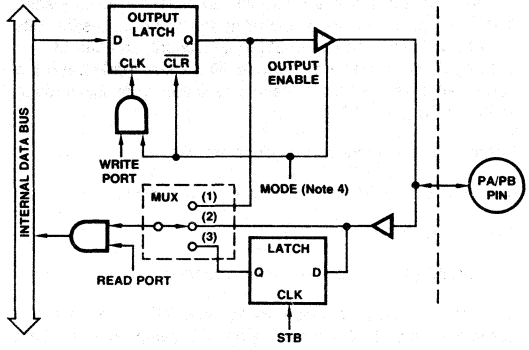
Control	Input Mode	Output Mode
BF	LOW	LOW
INTR	LOW	HIGH
$\overline{STB}$	Input Control	Input Control

The set and reset of INTR and BF with respect to $\overline{STB}$, $\overline{WR}$ and $\overline{RD}$ timing are shown in Figure 8.

To summarize, the register's assignments are:

Address	Pinouts	Functions	No of Bits
XXXXX000	Internal	Command/Status Register	8
XXXXX001	PA_{0-7}	General Purpose I/O Port	8
XXXXX010	PB_{0-7}	General Purpose I/O Port	8
XXXXX011	PC_{0-5}	General Purpose I/O Port or Control Lines	6

The following diagram shows how I/O Ports A and B are structured within the 8155H and 8156H:



8155H/8156H
One Bit of Port A or Port B

Notes: 1. Output Mode
2. Simple Input
3. Strobed Input
} Multiplexer Control
4. = 1 for output mode
= 0 for input mode.

Read Port = (IO/$\overline{M}$ = 1) • ($\overline{RD}$ = 0) • (CE active) • (Port address selected)

Write Port = (IO/$\overline{M}$ = 1) • ($\overline{WR}$ = 0) • (CE active) • (Port address selected)

Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pins will go LOW. When the 8155H/8156H is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of Port C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

Table 1. Table of Port Control Assignment

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{STB}$ (Port A Strobe)	A $\overline{STB}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{STB}$ (Port B Strobe)

Timer Section

The timer is a 14-bit down counter that counts the "timer input' pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register.

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0 – 13 will specify the length of the next count, and bits 14 – 15 will specify the timer output mode. The value loaded into the count length register can have any value from 2_H through $3FFF_H$ in bits 0 – 13.

There are four modes to choose from:

0 – Puts out LOW during second half of count

1 – Square wave

2 – Single pulse upon TC being reached

3 – Repetitive single pulse every time TC is readied and automatic reload of counter upon TC being reached until instructed to stop by a new command loaded into C/S.

Bits 6 – 7 of the Command/Status Register Contents are used to start and stop the counter. There are four commands to choose from. (See the further description on Command/ Status Register.)

C/S7	C/S6	
0	0	NOP – Do not affect counter operation.
0	1	STOP – NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC – Stop immediately after present TC is reached (NOP if timer has not started).
1	1	START-Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

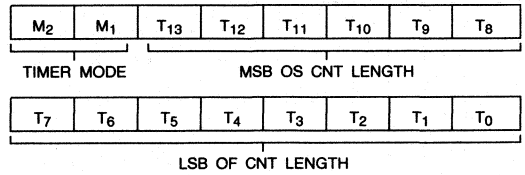
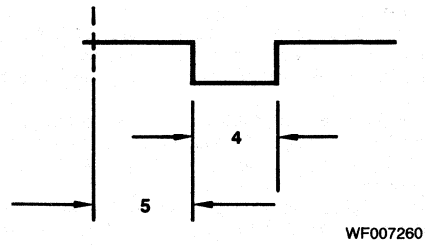


Figure 4. Timer Format

M2 and M1 define the timer mode as follows:

M2	M1	
0	0	Puts out LOW during second half of count.
0	1	Square wave, i.e., the period of the square wave equals the count length programmed with automatic reload at terminal count.
1	0	Single pulse upon TC being reached.
1	1	Automatic reload, i.e., single pulse every time TC is reached.

Note: In case of an asymmetric count, i.e., 9, larger half of the count will be HIGH, the larger count will stay active as shown in Figure 5.



Note: 5 and 4 refer to the number of clock cycles in that time period.

Figure 5. Asymmetric Count

The counter in the 8155H is not initialized to any particular mode or count when hardware RESET occurs, but RESET does stop the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages With
Respect to V_{SS} ..... −0.5V to +7.0V
Power Dissipation 1.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Part Number	T_A	V_{CC}	I_{CC}
8155 8156 8155-2 8156-2	0°C to 70°C	5V ±5%	180mA
8155H 8156H 8155H-2 8156H-2	0°C to 70°C	5V ±10%	125mA

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS (over Operating Ranges)

Parameters	Description		Test Conditions	Min	Max	Units
V_{IL}	Input Low Voltage			−0.5	0.8	Volts
V_{IH}	Input High Voltage			2.0	V_{CC}+0.5	Volts
V_{OL}	Output Low Voltage		I_{OL} = 2mA		0.45	Volts
V_{OH}	Output High Voltage		I_{OH} = −400μA	2.4		Volts
I_{IL}	Input Leakage		V_{IN} = V_{CC} to 0V		±10	μA
I_{LO}	Output Leakage Current		0.45V ≤ V_{OUT} ≤ V_{CC}		±10	μA
I_{CC}	V_{CC} Supply Current	8155, 8156	0.45V ≤ V_{OUT} ≤ V_{CC}		180	mA
		8155H, 8156H			125	mA
I_{IL} (CE)	Chip Enable Leakage	8155H, 8155	V_{IN} = V_{CC} to 0V		+100	μA
		8156H, 8156			−100	μA

SWITCHING TEST INPUT/OUTPUT WAVEFORM

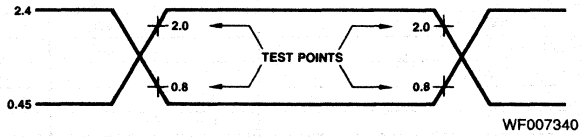


Figure 6.

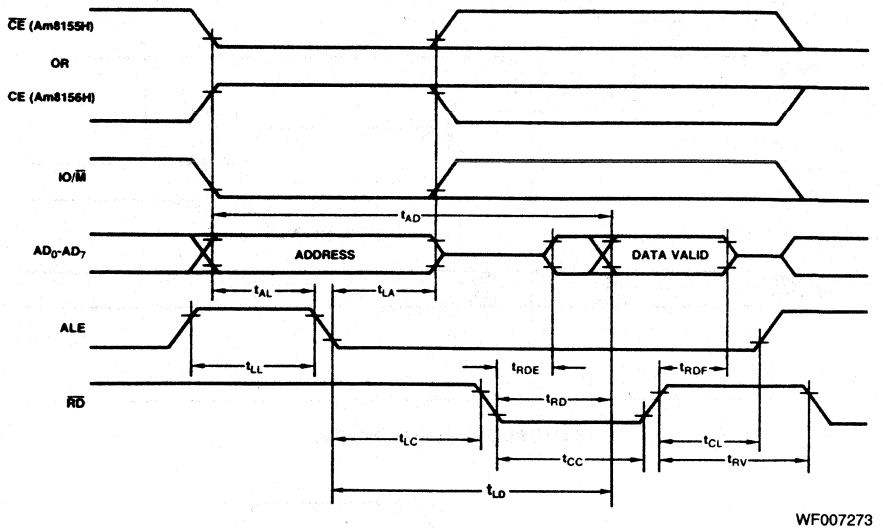
SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	8155/56/55H/56H		8155-2, 8156-2 8155H-2, 8156H-2		Units
		Min	Max	Min	Max	
t_{AL}	Address to Latch Set-up Time	50		30		ns
t_{LA}	Address Hold Time after Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{AD}	Address Stable to Data Out Valid		400		330	ns
t_{LD}	Latch to Data Out Valid		350		270	ns
t_{WT}	WRITE to TIMER-IN (For Writes Which Start Counting)	360		200		ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float After READ	0	100	0	80	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data in to WRITE Set-up Time	150		100		ns
t_{WD}	Data In Hold Time After WRITE	25		25		ns
t_{RV}	Recovery Time Between Controls	300		200		ns
t_{WP}	WRITE to Port Output		400		300	ns
t_{PR}	Port Input Set-up Time	70		50		ns
t_{RP}	Port Input Hold Time	50		10		ns
t_{SBF}	Strobe to Buffer Full		400		300	ns
t_{SS}	Strobe Width	200		150		ns
t_{RBE}	READ to Buffer Empty		400		300	ns
t_{SI}	Strobe to INTR On		400		300	ns
t_{RDI}	READ to INTR Off		400		300	ns
t_{PSS}	Port Set-up Time to Strobe	50		0		ns
t_{PHS}	Port Hold Time After Strobe	120		100		ns
t_{SBE}	Strobe to Buffer Empty		400		300	ns
t_{WBF}	WRITE to Buffer Full		400		300	ns
t_{WI}	WRITE to INTR Off		400		300	ns
t_{TL}	TIMER-IN to TIMER-OUT LOW		400		300	ns
t_{TH}	TIMER-IN to TIMER-OUT HIGH		400		300	ns
t_{RDE}	Data Bus Enable from READ Control	10		10		ns
t_1	TIMER-IN Low Time	80		40		ns
t_2	TIMER-IN High Time	120		70		ns

Note: Test Condition: 100pF Load.

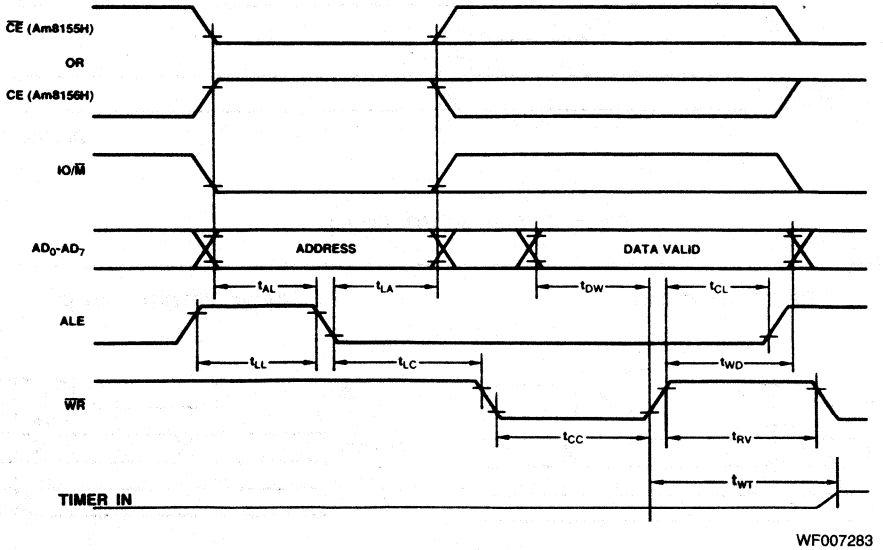
SWITCHING WAVEFORMS

A. READ CYCLE.



WF007273

B. WRITE CYCLE.

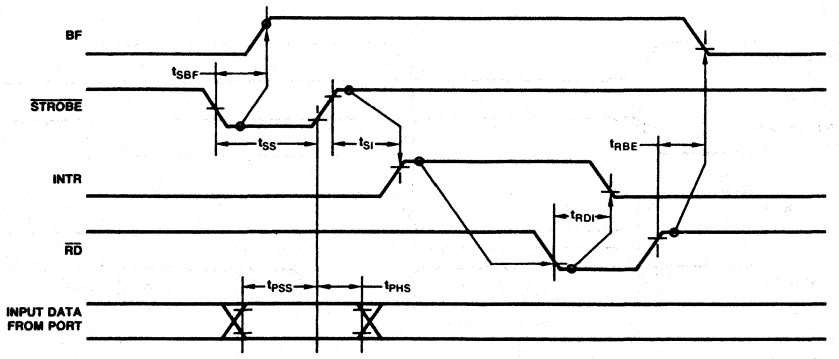


WF007283

Figure 7. 8155H/8156H Read/Write Timing Diagrams

SWITCHING WAVEFORMS (Cont.)

A. STROBED INPUT MODE.



B. STROBED OUTPUT MODE.

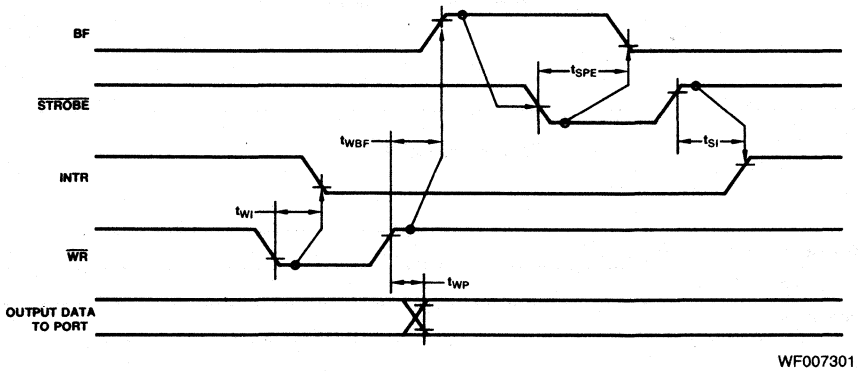
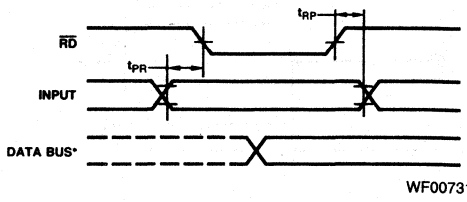
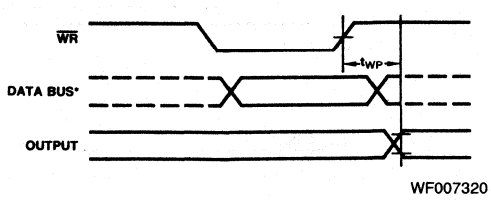


Figure 8. Strobed I/O Timing

BASIC INPUT MODE.



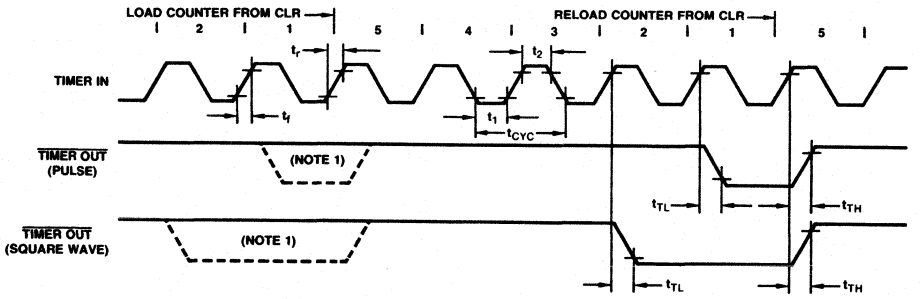
BASIC OUTPUT MODE.



*Data bus timing is shown in Figure 7.

Figure 9. Basic I/O Timing Waveform

SWITCHING WAVEFORMS (Cont.)



Note 1: The timer output is periodic if in an automatic reload mode (M_1 mode bit = 1).

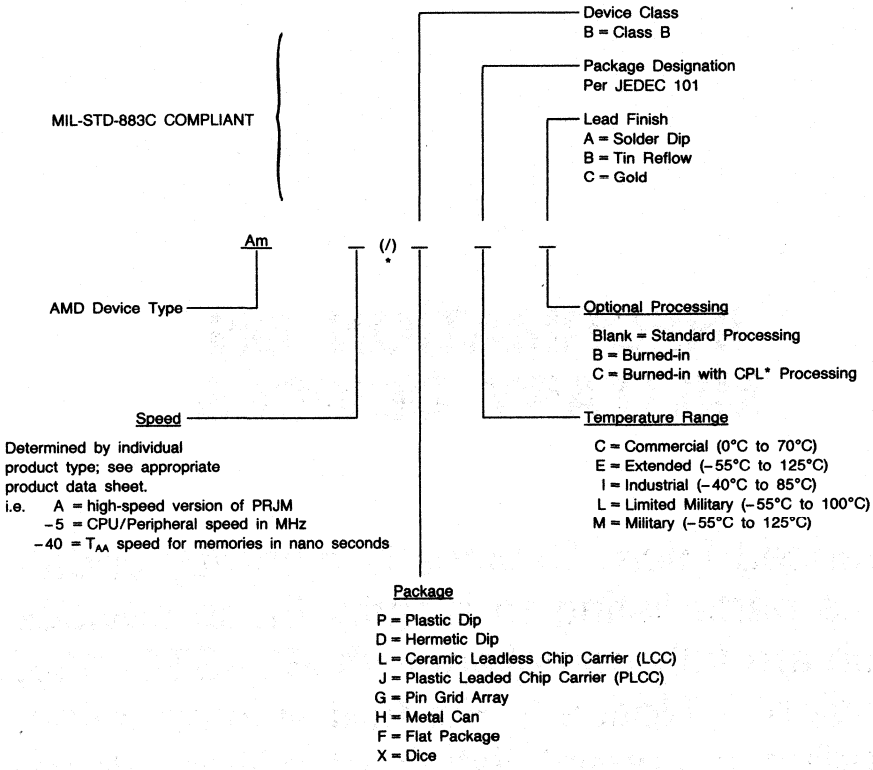
Figure 10. Timer Output Waveform Countdown from 5 to 1

General Information Index

INFORMATION ON MILITARY DEVICES

Advanced Micro Devices is currently updating its parts listing to identify those devices which are fully compliant with MIL–STD–883, Revision C, Notice 2. For further information, including a current listing of military flows, and to answer questions regarding specific AMD parts, please contact your local AMD sales representative.

AMD Originated Product Ordering Information



*The slash is used only for designation of APL (Approved Products List) and CPL (Controlled Products List) following processing flow conforming to MIL-STD-883C requirements.

Refer to page 7-1 for Essential Information on Military Devices

Advanced Micro Devices
General Product Information

The Information Products Division (IPD) of Advanced Micro Devices is headquartered in Austin, Texas. IPD is responsible for all MOS Microprocessor and Peripherals as well as telecommunication devices.

IPD products are offered in two standard operating temperatures (ambient) ranges:

1. Commercial : 0°C to +70°C
2. Industrial : −40°C to +85°C

Commercial product is available in plastic and hermetic packages. Industrial product is available in hermetic packages only.

Burn-In

1. Commercial and Industrial product can also be ordered with Burn-In. Burn-In is done at 125C for 96 hours (or equivalent).

Manufacturing locations for IPD products are:

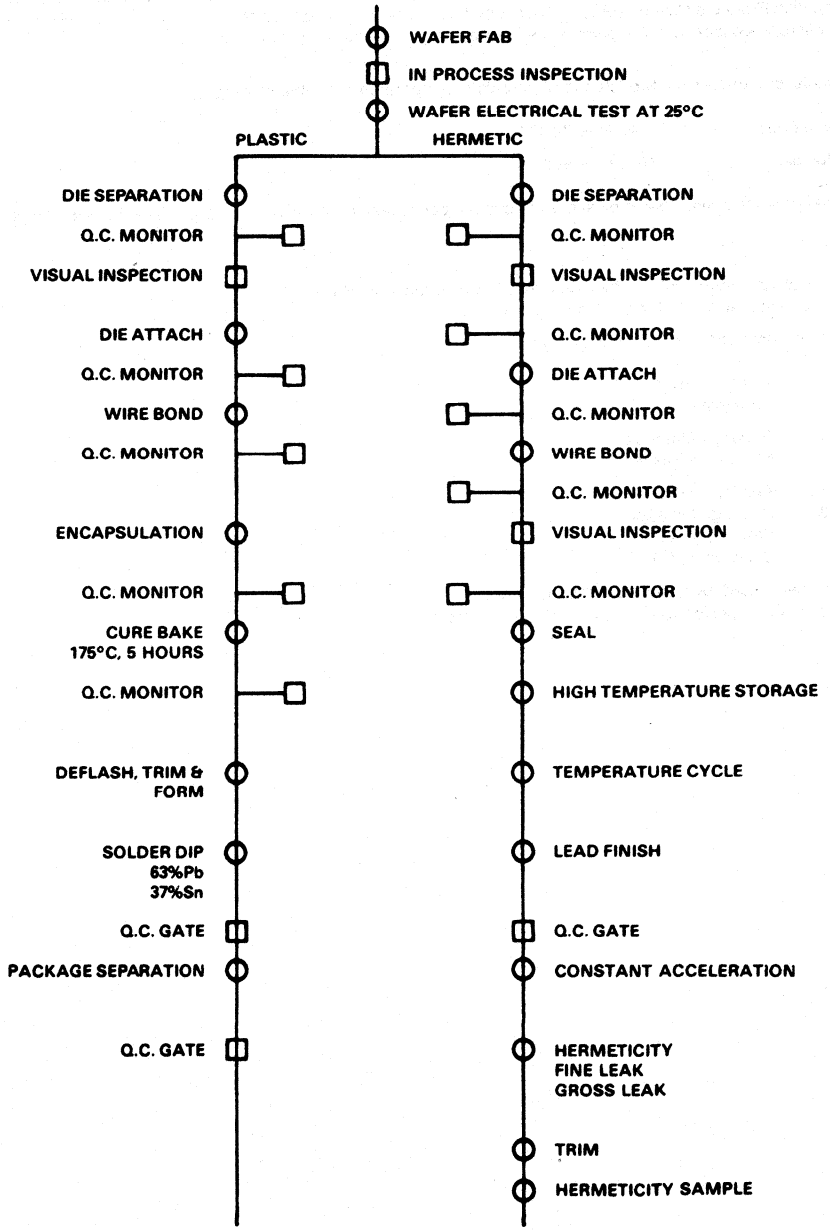
Wafer Fab	: Austin, Texas Sunnyvale, CA
*Assembly	: Penang, Malaysia Manila, Phillipines
Test, Mark, *Burn-In, & Ship	: Austin, Texas Penang, Malaysia. Woking, U.K. Sunnyvale, CA

*Burn-In is performed by qualified subcontractors.
*Assembly may be performed by qualified subcontractors.

Refer to page 7-1 for Essential Information on Military Devices

MANUFACTURING FLOWS

The typical manufacturing flow is described in two sections. Section I deals with flow from wafer fab to assembly. Section II describes the post-assembly flows for commercial product.



Section I

Refer to page 7-1 for Essential Information on Military Devices

TYPICAL AMD COMMERCIAL FLOW (HERMETIC/PLASTIC)

1	INITIAL ELECTRICAL TEST	100% screen
2	MARK	
3	BURN-IN	OPTIONAL. Conditions per AMD 07-XXX.
4	FINAL ELECTRICAL TEST	100% screen
5	LEAD SCAN/STRAIGHTEN	
6	VISUAL/MECHANICAL INSPECTION	100% per AMD 16-049.
7	Q.A. INSPECTION DOCUMENTATION/TEST VERIFICATION	Per AMD 06-027.
8	Q.A. INSPECTION VISUAL/MECHANICAL	Per AMD 06-027.
9	Q.A. INSPECTION HERMETICITY	Hermetic product only. Per AMD 00-006. Fine leak only per AMD 05-519.
10	EXCELSIOR MONITOR ELECTRICAL	Per AMD 00-005 and 00-006.
11	PACK/SHIP	Per AMD 16-050.

Section II

Refer to page 7-1 for Essential Information on Military Devices

Package Configurations

	MULTILAYER CERAMIC		CERAMIC	PLASTIC	HEADERS
	BRAZED PACKAGES	CHIP CARRIER	CERDIP	DIP and CHIP CARRIER	
PACKAGE BODY MATERIAL	90% Alumina (min)	90% Alumina (min)	90% Alumina (min)	Novolac Epoxy	Alloy 42
DIE ATTACH PAD METALLIZATION	Gold	Gold	Gold / Silver Palladium	Gold / Silver	Gold
DIE ATTACH MATERIAL	Gold/Silicon	Gold/Silicon	Gold/Silicon	Gold/Silicon / Silver Epoxy	Gold/Silicon
DIE ATTACH TEMPERATURE	440°C. Max	440°C. Max	440°C. Max	440°C. Max / 180°C. (Curing Temp)	390°C. Max.
BOND FINGER METALLIZATION	Gold	Gold	Aluminum	Gold / Silver	Gold
BONDING WIRE	Aluminum	Aluminum	Aluminum	Gold	Aluminum
BONDING METHOD	Ultrasonic	Ultrasonic	Ultrasonic	Ball-Bonding	Ultrasonic
SEAL RING METALLIZATION	Gold	Gold	N/A	N/A	Gold / Nickel
SEAL MATERIAL	Gold/Tin Eutectic	Gold/Tin Eutectic	Vitreous Glass	N/A	N/A / N/A
LID MATERIAL	Alloy 42 (Gold Plated) / Alloy 42 (Tin Plated)	Alloy 42 (Gold Plated)	90% Alumina (min)	N/A	Nickel/Nickel Clad Stainless Steel
SEAL TEMPERATURE	370°C Max.	370°C Max.	460°C Max.	185°C Max. (Mold Temp)	Resistance Weld
SEAL AMBIENT	Nitrogen	Nitrogen	Air	N/A	Nitrogen
LEAD MATERIAL	Alloy 42	N/A	Alloy 42	Alloy 42 / Alloy 42 Copper	Alloy 42
LEAD FINISH	Gold / Tin	Gold (Solder Pad)	Tin	Solder	Gold / Nickel

Note: N/A = Not Applicable

Refer to page 7-1 for Essential Information on Military Devices

Surface Mount Technology

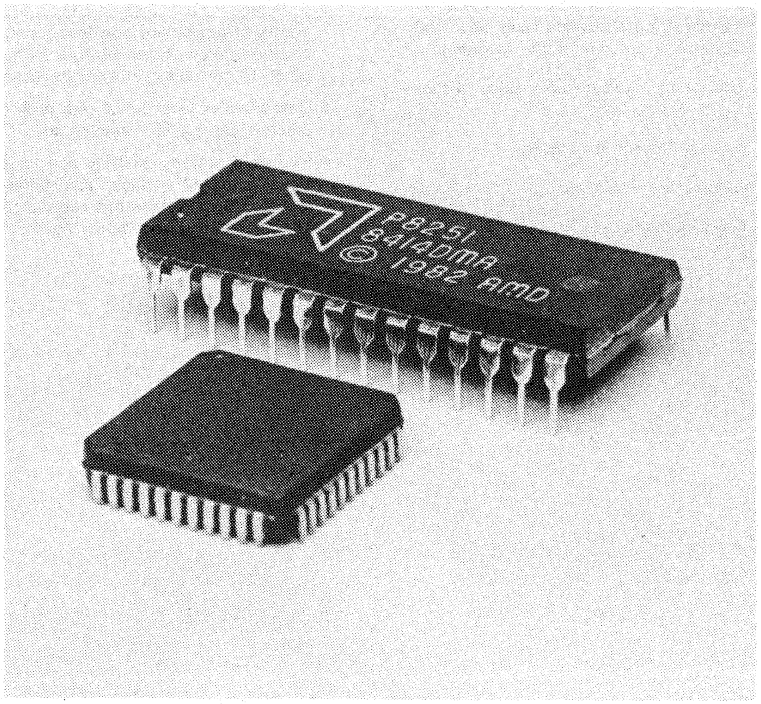
NEED FOR SURFACE MOUNT PRODUCTS

The demand for high-density, cost-effective printed circuit boards has prompted the electronics industry to seek alternative methods to traditional plated-through-hole technology. One such alternative is surface mounting. The advantages of surface mounting are numerous but the bottom line is that it is cost effective and will begin to displace plated-through-hole technology as the availability of surface-mount components increase.

AMD is **breaking away** to fully support the growth of the surface-mount industry. Within MMP we plan to provide a full range of surface-mount products in the immediate future which will allow our solid-state customers to pack more functions in a given size enclosure or maintain the same functional capability, but reduce the size.

WHAT IS SURFACE MOUNT TECHNOLOGY

Basically, a surface-mounted product is literally the heart of the more traditional dip package. As one can see, the most obvious difference is the smaller size of the surface mount part. In 1983, more than 9,000 part numbers were available in surface mountable packages. MMP will provide plastic leaded chip carrier, ceramic leadless chip carrier, and pin-grid array packages for the majority of our processors/controllers and peripheral IC's. Of these types of surface mounted packages, the plastic leaded chip carrier (PLCC) will be the star. A PLCC is a surface mounted device; i.e., it is physically mounted on top of PC board. It is held in place by the PLCC leads which have been reflowed soldered on the PC board traces. The PLCC offers the most advantages of surface mounted packaging.



Refer to page 7-1 for Essential Information on Military Devices

This table illustrates why surface mounted PLCC's will be the dominant packaging concept for the future and how it compares to conventional plastic DIP.

Characteristics	Insertion Plastic DIP	Surface Mount PLCC
High Pin Count	0	*
Small Package Area	0	*
Device Cost	*	*
Double-Sided Board Mounting	0	*
Automated Board Assembly	*	*
Package Power Dissipation Capability	+	+
Ease of Board Debug	*	*
Ease of Board Repair	+	*
Relative PC Board Cost	+	*
Board Reliability (Temp. Cycle)	*	*

*Excellent + Acceptable 0 Marginal

WHY SURFACE MOUNT

One of the primary goals of today's solid-state equipment manufacturers is increased density – pack more functions into a given size enclosure or maintain the same functional capability, but reduce the size. Another goal is to reduce cost.

Surface mount technology offers overall board size reduction of as much as 50 percent, cost savings, and improved reliability. These improvements are due to the following:

- Components can be mounted on both sides of the board.
- The number of board layers is reduced.
- Plated through-holes are eliminated.
- Assembly is fully automated.
- Placement equipment runs at higher throughout rates than is possible with insertion equipment.
- Inductive effects are decreased because package leads are shorter. Shorter distances between components allows elimination of some of the semiconductors usually required for driving capacitive loads.
- A surface mount assembly line requires fewer pieces of equipment and less manufacturing space because certain types of equipment, such as lead clinching machines, are not necessary.
- Cost-effective onshore assembly is possible as the process can be fully automated.
- Increased system reliability due to: (1) shortened signal paths, (2) reduced inductions and resistances, (3) more vibration resistance, and (4) more efficient thermal impedance paths.

Refer to page 7-1 for Essential Information on Military Devices

TECHNICAL REPORT

No. TR202

THERMAL CHARACTERIZATION OF PACKAGE DEVICES

BY J. L. HAYWARD

(REVISED - JANUARY 1985)

TECHNNICAL LIBRARY

Refer to page 7-1 for Essential Information on Military Devices

ABSTRACT

Determination of the Thermal Resistance of Packaged Devices is of concern to the designer of new devices and to AMD customers. The advanced package and material development group has undertaken the task of characterizing current AMD products and quantifying package-related influences on Thermal Resistance. This report describes some of these effects and the technique used to measure Thermal Resistance.

1.0 DEFINITION OF THERMAL RESISTANCE

The reliability of an integrated circuit is largely dependent on the maximum temperature which the device will attain during operation. Because the stability of a semiconductor junction declines with increasing temperature, knowledge of the thermal properties of the packaged device becomes an important factor during device design. In order to increase the operating lifetime of a given device, the junction temperatures must be minimized. This demands knowledge of the thermal resistance of the completed assembly and specification of the conditions in which the device will function properly. As devices become both smaller and more complex and the requirement for high speed operation becomes more important, heat dissipation will become an ever more critical parameter.

Thermal resistance is defined as the temperature rise per unit power dissipation above some referenced condition. The unit of measure is typically °C/watt. The relationship between junction temperature and thermal resistance is given by:

$$T_j = T_X + P_d \theta_{jx} \qquad (1)$$

where: T_j = junction temperature
T_X = reference temperature
P_d = power dissipation
θ_{jx} = thermal resistance
X = some defined test condition

In general, one of three conditions is defined for measurement of thermal resistance:

θ_{jc} – thermal resistance measured with reference to the temperature at some specified point on the package surface.

θ_{ja} (still air) – thermal resistance measured with respect to the temperature of a specified volume of still air.

θ_{ja} (moving air) – thermal resistance measured with respect to the temperature of air moving at a specified velocity.

The relationship between θ_{jc} and θ_{ca} is

$$\theta_{ja} = \theta_{jc} + \theta_{ca}$$

where θ_{ca} is a measure of the heat dissipation due to natural convection (still air) or forced convection (moving air) and the effect of heat radiation and mounting techniques. θ_{jc} is dependent solely on material properties and package geometry; θ_{ja} includes the influence of the surface area of the package and environmental conditions. Each of these definitions of thermal resistance is an attempt to simulate some manner in which the package device may be used.

The thermal resistance of a packaged device, however measured, is a summation of the thermal resistances of the individual components of the assembly. These in turn are functions of the thermal conductivity of the component materials and the geometry of the heat flow paths. Like other material properties, thermal conductivity is usually temperature dependent. For alumina and silicon, two common package materials, this dependence can amount to a 30% variation in thermal conductivity over the operating temperature range of the device. The thermal resistance of a component is given by

$$\theta = \frac{L}{K(T)A} \qquad (2)$$

where: L = length of the heat flow path
A = cross sectional area of the heat flow path
K(T) = thermal conductivity as a function of temperature

and the overall thermal resistance of the assembly (discounting convective effects) will be:

$$\theta = \Sigma\theta_n = \Sigma\frac{L}{K_n A}$$

But since the heat flow path through a component is influenced by the materials surrounding it, determination of L and A is not always straightforward.

A second factor that affects the thermal resistance of a packaged device is the power dissipation level and, more particularly, the relationship between power level and die geometry, i.e., power distribution and power density. By rearrangement of equation 1 to

$$P_d = \frac{1}{\theta_{jx}}(T_j - T_X) = \frac{1}{\Sigma\theta_n}(T_j - T_X) \qquad (3)$$

the relationship between P_d and T_j can be more clearly seen. Thus, to dissipate a greater quantity of heat for a given geometry, T_j must increase and, since the individual $R_{\theta n}$ will also increase with temperature, the increase in T_j will not be a linear function of increasing power levels.

A third factor of concern is the quality of the material interfaces. In terms of package construction, this relates specifically to the die attach bond, and for those packages having a heatsink, the heatsink attach bond. The quality of the die attach bond will most severely influence the package thermal resistance as this is the area which first impedes the transfer of heat out of the silicon die. Indeed, it seems likely that the initial thermal response of a powered device can be directly related to the quality of the die attach bond.

2.0 EXPERIMENTAL METHOD

The technique for measurement of thermal resistance involves the identification of a temperature-sensitive parameter on the device and monitoring this parameter while the device is powered. For bipolar integrated circuits the forward voltage of the substrate isolation diode provides a convenient parameter to measure and has the advantage of a linear dependence on temperature. MOS devices which do not have an accessible substrate diode present greater measurement difficulties and may require simulation through use of a specially designed thermal test die. Choice of the parameter to be measured must be made with some care to insure that the results of the measurement are truly representative of the thermal state of the device being investigated. Thus measurement of the substrate isolation diode which is generally diffused across the area of the die yields a weighted average of the condition of the individual junctions across the die surface. Measurement of a more local source would yield a less generalized result.

Refer to page 7-1 for Essential Information on Military Devices

For MOS devices, simulation is accomplished using the thermal test die. The basis for this test die is a 25 mil square cell containing an isolated diode and a 1KΩ resistor. The resistors are interconnected from cell to cell on the wafer before it is cut into multiple arrays of the basic unit cell. In use the device is powered via the resistors with voltage or current adjusted for the proper level and the voltage drop of the individual diodes is monitored as in the case of actual devices.

Prior to the thermal resistance test, the diode voltage/temperature calibration must be determined. This is done by measuring the forward voltage at 1mA current level at two different temperatures. The diode calibration factor is then:

$$K_f = \frac{T_2 - T_1}{V_2 - V_1} = \frac{\Delta T}{\Delta V} \qquad (4)$$

in units of °C/mV. For most diodes used for this test the voltage/temperature relationship is linear and these two measurement points are sufficient to determine the calibration.

The actual thermal resistance measurement has two alternating phases: measurement and power on. The device under test is pulse powered with an ON duty cycle of 99% and a repetition rate of < 100Hz. During the brief OFF states the device is reverse-biased with a 1mA current and the voltage drop is measured. The series of voltage readings are averaged over short periods and compared to the voltage reading obtained before the device was first powered ON. The thermal resistance is then computed as:

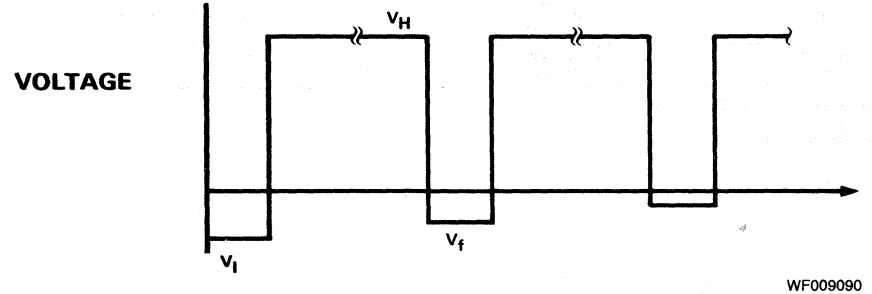
$$\theta_{jx} = \frac{K_f(V_f - V_i)}{V_H I_H} = \frac{K_f \Delta V}{P_d} \qquad (5)$$

where: K_f = calibration factor
V_i = initial forward voltage value
V_f = current forward voltage value
V_H = heating voltage
I_H = heating current

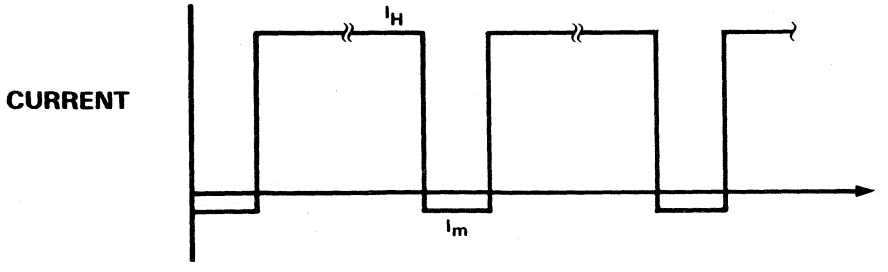
The pulsing measurement is continued until the device has reached thermal equilibrium and the final value measured is the equilibrium thermal resistance of the device under test.

When the end result desired is θ_{ja} (still air), the device and the test fixture (typically a standard burn-in socket) are enclosed in a box containing approximately 1 cubic foot of air. For θ_{jc} measurements the device is attached to a large metal heatsink. This insures that the reference point on the device surface is maintained at a constant temperature. The requirements for measurement of θ_{ja} (moving air) are rather more complex and involve the use of a small wind tunnel with capability for monitoring air pressure, temperature and velocity in the area immediately surrounding the device tested. Standardization of this last test requires much careful attention.

WAVEFORMS FOR PULSED THERMAL RESISTANCE TEST



WF009090



WF009080

3.0 Experimental Results

The thermal resistance data included in the attached table was collected using the procedure outlined in the preceding section. Thus, what is presented is the output of tests on representative samples of AMD products. This data has resulted from an ongoing program undertaken by members of the Advanced Package and Material Development group and is accurate at the date of this writing. The values listed for power, power density, θ_{ja} and θ_{jc} are the mean values for the specified test sample size. The accuracy of the individual measurements is about ±5%. This table will be updated and expanded at regular intervals as new or revised data becomes available.

THERMAL RESISTANCE OF AMD PRODUCTS

Lead Count	Package	Sample Size	Device Type	Die Size (mils)	Power (mW)	Power Density	θ_{ja} (C/W)	θ_{jc} (C/W)
24	Cerdip	10	Am74S181	83 x 91	750	99	49	11
	Plastic	10	Am74LS181	78 x 92	140	19	99	43
		10	Am74S181	83 x 91	770	102	115	57
		10(cu)	Am74S191	83 x 91	767	62	68	35
		15(cu)	Am74LS81	78 x 92	120	17	14	N/A
	Cerpak	20	Am74S181	83 x 91	740	98	99	8
24 (300 mil)	Cerdip	10	AmZ8127	88 x 98	750	87	55	14
		10	Am74S181	83 x 91	720	95	57	13
28	Chip Carrier	22	Am25LS2525	97 x 122	280	24	69	N/A
	Cerdip	16	Am2902A	110 x 160	530	30	29	N/A
	Plastic	7	Am2902A	110 x 160	520	29	85	N/A
	Flatpack	8	Am2955	100 x 129	608	47	105	38
32	Chip Carrier	9	Am27S191	137 x 213	668	22	75	N/A
40	Sidebraze	10	Am2901B	117 x 128	850	57	35	7
	Cerdip	12	Am2910A	170 x 194	1150	35	37	9
		18	Am2901B	117 x 128	910	61	36	7
	Plastic	11	Am2910A	170 x 194	1090	33	66	27
		11	Am2901B	117 x 128	1010	68	73	34
		12(cu)	Am2910A	170 x 194	1290	39	44	19
		11(cu)	Am29013	117 x 128	1130	76	46	20
44	Chip Carrier	12	Am25LS2516	197 x 205	1580	40	52	N/A
		12	Am2901A	117 x 128	870	58	57	N/A
48	Sidebraze	14	Am2903A	163 x 197	1080	34	37	10
	Plastic	10(cu)	Am2903A	163 x 197	1160	37	55	N/A
52	Chip Carrier	9	Am2903A	163 x 197	1060	33	44	N/A
	Topbraze	9	Am29116	311 x 251	3135	40	19	4

(1) Tests Performed with LCC in Burn-In Sockets
(cu) Copperlead Frame
(N/A) Not Presently Available

Refer to page 7-1 for Essential Information on Military Devices

PLCC PINOUTS FOR MMP DEVICES (44/28 LEAD)

PACK. PIN#	DEVICE PIN						
	Z8530	8086/8088	8237A	8253	8255A	8259A	Am9518/ AmZ8068
1	40	NC	NC	1	1	1	1
2	1	1	1	2	2	2	NC
3	2	2	2	3	3	3	2
4	3	3	3	4	4	4	3
5	4	4	4	5	5	5	4
6	5	5	NC	6	6	6	5
7	6	6	5	7	7	7	6
8	7	7	6	8	8	8	7
9	8	8	7	9	9	9	8
10	9	9	8	10	10	10	9
11	10	10	9	11	11	11	10
12	11	11	10	12	12	12	11
13	12	12	11	13	13	13	12
14	13	13	12	14	14	14	13
15	14	14	13	15	15	15	14
16	15	15	14	16	NC	16	15
17	NC	16	NC	17	NC	17	NC
18	NC	NC	15	18	16	18	16
19	16	17	16	19	17	19	17
20	17	18	17	20	18	20	18
21	18	19	18	21	19	21	19
22	19	20	19	22	20	22	20
23	20	NC	20	23	21	23	21
24	21	21	21	24	22	24	22
25	22	22	22	25	23	25	23
26	23	23	23	26	24	26	24
27	24	24	24	27	25	27	25
28	NC	25	25	28	NC	28	NC
29	25	26	26		26		26
30	26	NC	27		27		27
31	27	27	28		28		28
32	28	28	29		29		29
33	29	29	30		30		30
34	30	30	31		31		31
35	31	31	32		32		NC
36	*	32	33		33		32
37	32	33	34		34		33
38	33	34	35		35		34
39	34	35	NC		36		35
40	35	36	36		NC		36
41	36	37	37		37		37
42	37	38	38		38		38
43	38	39	39		39		39
44	39	40	40		40		40

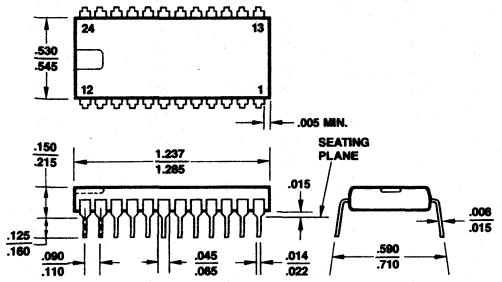
*THIS PIN ALLOWS THE IMPLEMENTATION OF BOTH Z8530 AND Z8030 IN A SINGLE PACKAGE CONTROLLED BY SELECT PIN REQUIRING:
5 VOLTS FOR Z8030
0 VOLTS FOR Z8530

Refer to page 7-1 for Essential Information on Military Devices

Package Outlines

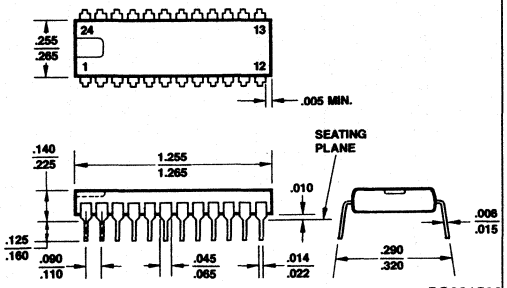
MOLDED DUAL IN-LINE PACKAGES

PC-024



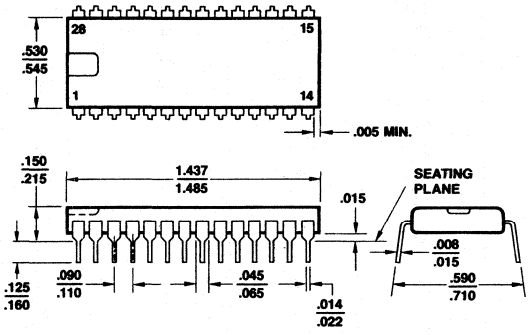
PO000722

PC-024 (Slim)



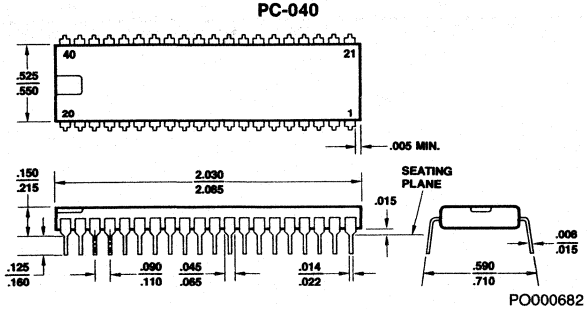
PO001590

PC-028



PO001560

PC-040

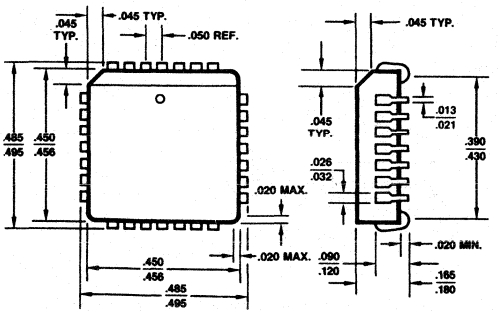


PO000682

Refer to page 7-1 for Essential Information on Military Devices

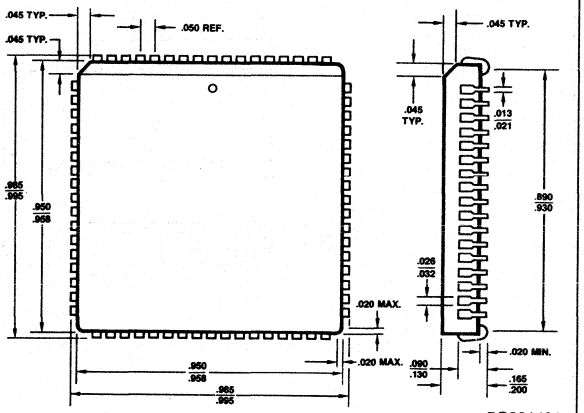
PLASTIC LEADED CHIP CARRIERS

28-PLCC



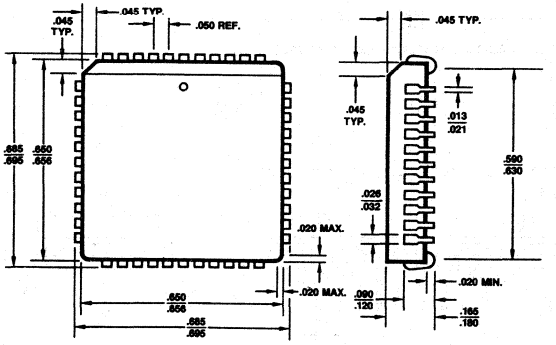
PO001481

68-PLCC



PO001491

44-PLCC

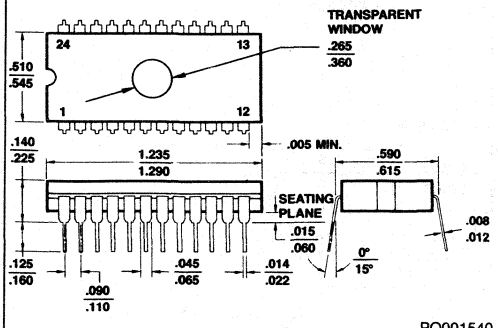


PO001471

Note. Standard lead finish is tin plate or solder dip.

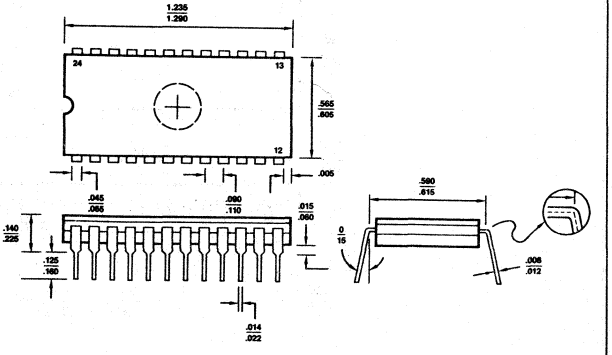
HERMETIC DUAL IN-LINE PACKAGES

CDV 024-1



PO001540

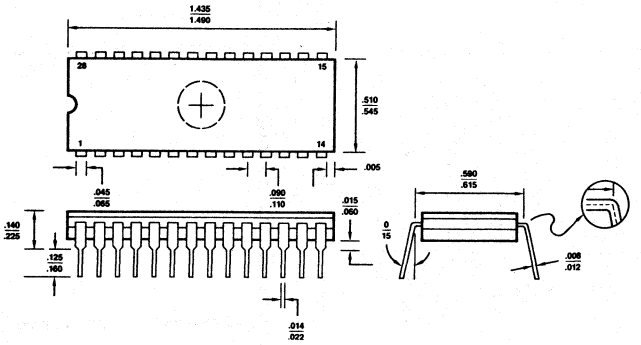
CD 6024-2



PO001340

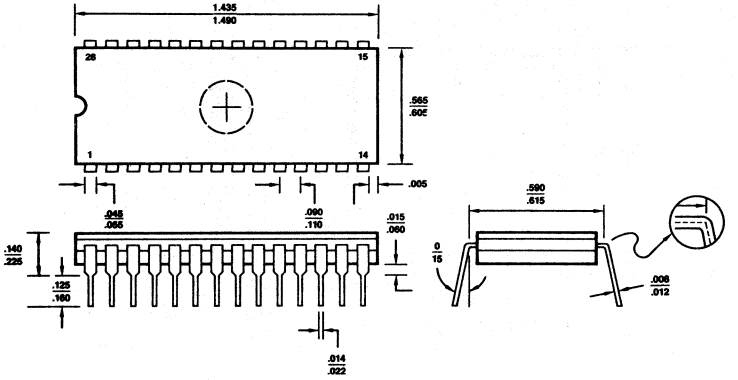
HERMETIC DUAL IN-LINE PACKAGES (Cont.)

CD 028-1



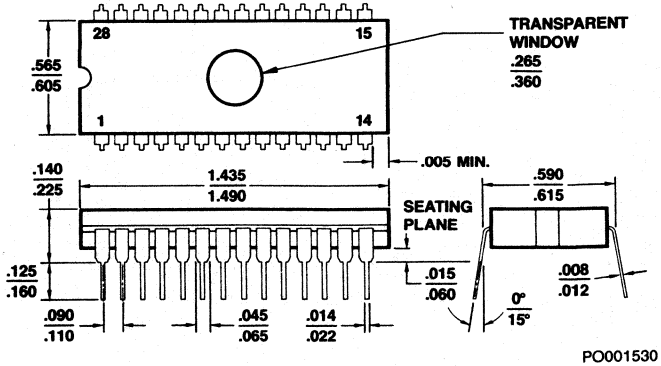
PO001330

CD 028-2



PO001350

CDV 028

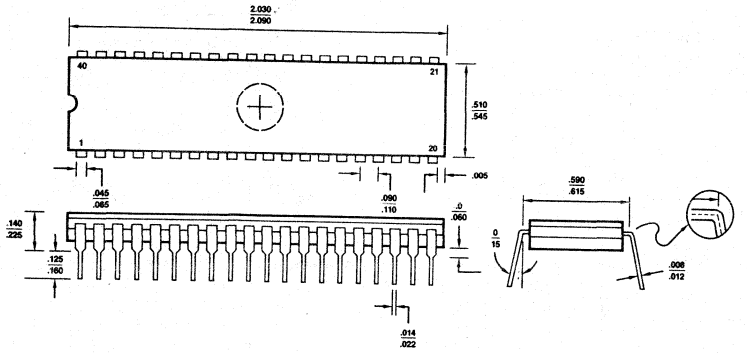


PO001530

Refer to page 7-1 for Essential Information on Military Devices

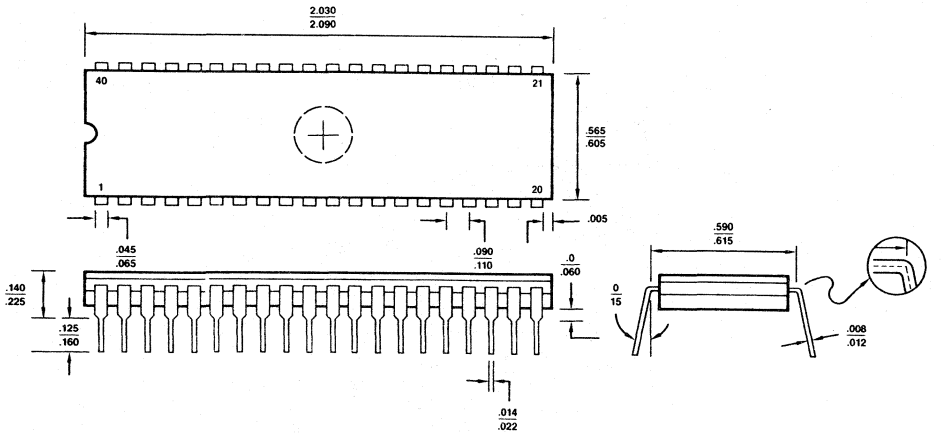
HERMETIC DUAL IN-LINE PACKAGES (Cont.)

CD 040-1



PO001360

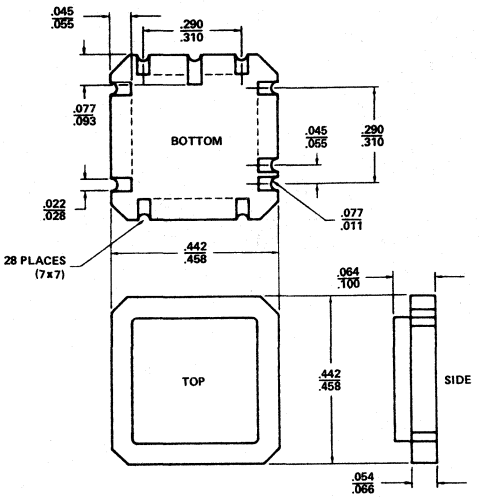
CD 040-2



PO001310

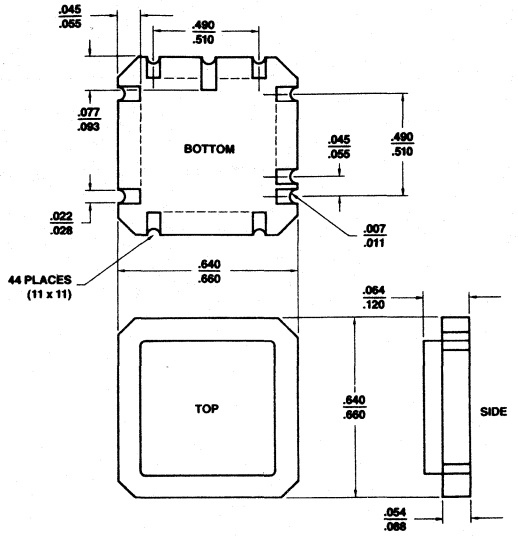
CERAMIC LEADLESS CHIP CARRIERS

CL 028-2



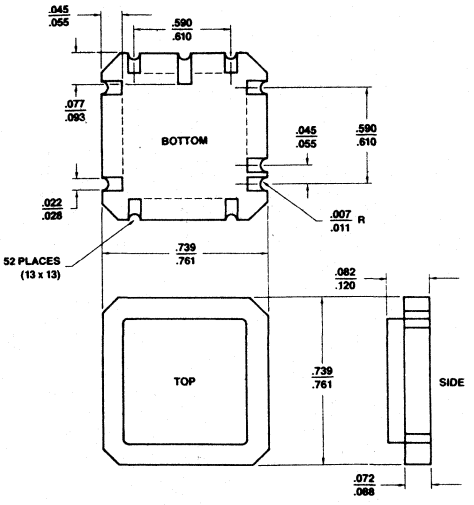
PO001131

CL 044



PO001140

CL 052-2

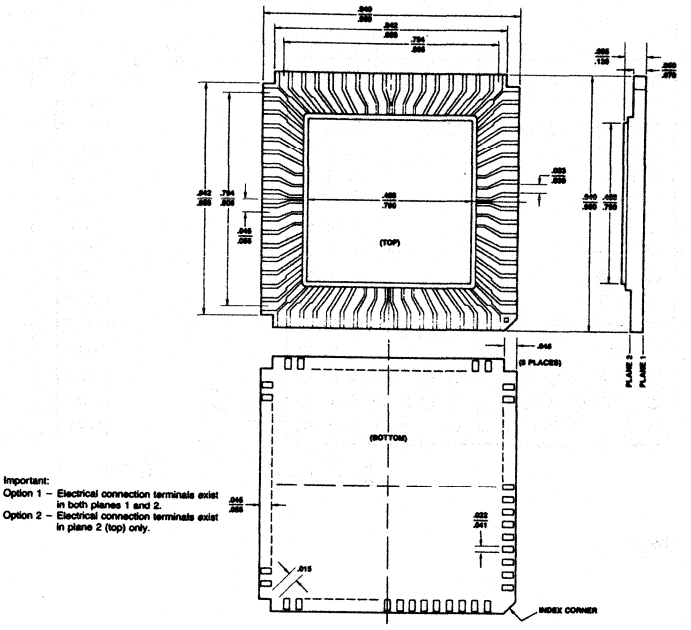


PO001150

Refer to page 7-1 for Essential Information on Military Devices

CERAMIC LEADLESS CHIP CARRIERS (Cont.)

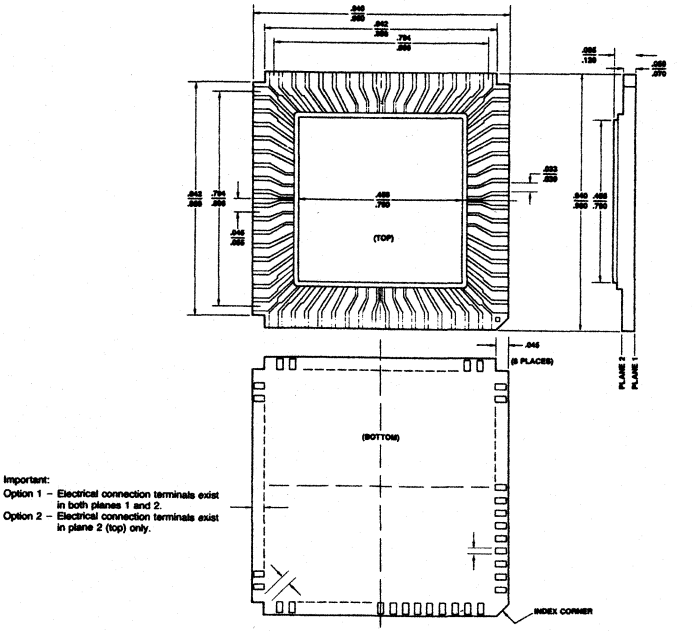
CL 068-1
(Option 1)



Important:
Option 1 – Electrical connection terminals exist in both planes 1 and 2.
Option 2 – Electrical connection terminals exist in plane 2 (top) only.

CD005720

CL 068-2
(Option 2)

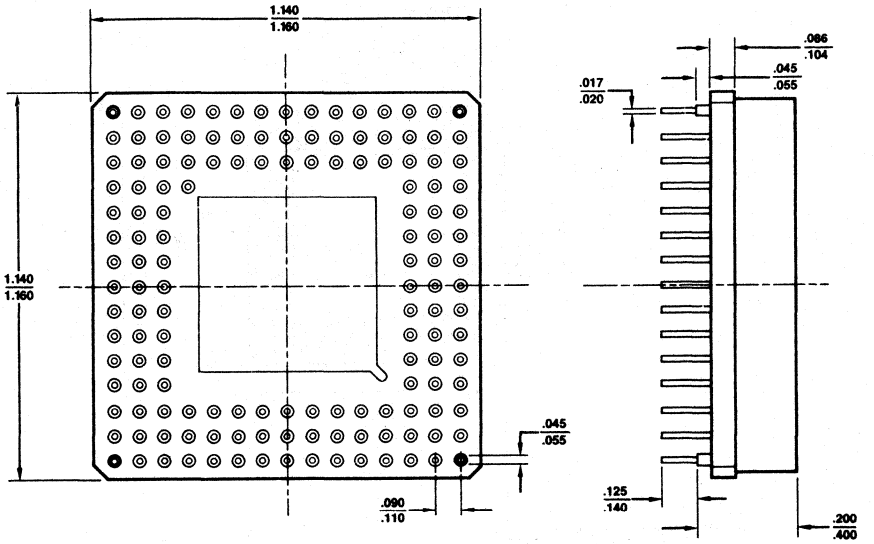


Important:
Option 1 – Electrical connection terminals exist in both planes 1 and 2.
Option 2 – Electrical connection terminals exist in plane 2 (top) only.

CD005710

CERAMIC LEADLESS CHIP CARRIERS (Cont.)

G-68-1



*Subject to change.

PO001761

U.S. AND CANADIAN SALES OFFICES

NORTHEAST AREA

Advanced Micro Devices
20 Mall Road
Burlington, Massachusetts 01803
Tel: (617) 273-3970

Advanced Micro Devices (Canada) Ltd.
2 Sheppard Avenue East
Suite 1610
Willowdale, Ontario
Canada M2N5Y7
Tel: (416) 224-5193

Advanced Micro Devices (Canada) Ltd.
AMD
4019 Carling #301
Kanata, Ottawa
Canada K2K2A3
Tel: (613) 592-0060

Advanced Micro Devices
290 Elwood Davis Road
Suite 316
Liverpool, New York 13088
Tel: (315) 457-5400

MID-ATLANTIC AREA

Advanced Micro Devices
Gateways
1000 Woodbury Road
Woodbury, New York 11797
Tel: (516) 364-8020

Advanced Micro Devices
Waterview Plaza, Suite 303
2001 U.S. Route #46
Parsippany, New Jersey 07054
Tel: (201) 299-0002

Advanced Micro Devices
2300 Computer Avenue
Suite D 19
Willow Grove, PA 19090
Tel: (215) 657-3101

Advanced Micro Devices
Commerce Plaza
5100 Tilghman Street, Suite 320
Allentown, Pennsylvania 18104
Tel: (215) 398-8006

Advanced Micro Devices
205 South Avenue
Poughkeepsie, New York 12601
Tel: (914) 471-8180

Advanced Micro Devices
10 Main Street South
Southbury, Connecticut 06488
Tel: (203) 264-7800

Advanced Micro Devices
7223 Parkway Drive #203
Dorsey, Maryland 21076
Tel: (301) 796-9310

SOUTHEAST AREA

Advanced Micro Devices
4740 North State Road #7
Bldg. C - Suite 202
Ft. Lauderdale, Florida 33319
Tel: (305) 484-8600

Advanced Micro Devices
15351Roosevelt Boulevard #201
Clearwater, Florida 33520
Tel: (813) 530-9971

Advanced Micro Devices
478Ballard Drive #14
Melbourne, Florida 32935
Tel: (305) 254-2915

Advanced Micro Devices
701 East Altamonte Drive
Suite 204
Altamonte Springs, Florida 32701
Tel: (305) 834-3333

Advanced Micro Devices
15 Technology Parkway #200
Norcross, Georgia 30092
Tel: (404) 449-7920

Advanced Micro Devices
8 Woodlawn Green, Suite 220
Woodlawn Road
Charlotte, North Carolina 28210
Tel: (704) 525-1875

Advanced Micro Devices
303 Williams Avenue Southwest
Suite 118
Huntsville, Alabama 35801
Tel: (205) 536-5505

Advanced Micro Devices
6501 Six Forks, Suite 150
Raleigh, North Carolina 27609
Tel: (919) 847-8471

MID-AMERICA AREA

Advanced Micro Devices
500 Park Boulevard, Suite 940
Itasca, Illinois 60143
Tel: (312) 773-4422

Advanced Micro Devices
5726 Professional Circle #205G
Indianapolis, Indiana 46241
Tel: (317) 244-7207

Advanced Micro Devices
7007 College Blvd.
Suite 330
Overland Park, KS 66211
Tel: (913) 451-3115

Advanced Micro Devices
9900 Bren Road East, Suite 601
Minnetonka, Minnesota 55343
Tel: (612) 938-0001

Advanced Micro Devices
3592 Corporate Drive, Suite 108
Columbus, Ohio 43229
Tel: (614) 891-6455

Advanced Micro Devices
16985 West Blue Mound Road,
Suite 201
Brookfield, Wisconsin 53005
Tel: (414) 782-7748

NORTHWEST AREA

Advanced Micro Devices
1245 Oakmead Parkway
Suite 2900
Sunnyvale, California 94086
Tel: (408) 720-8811

Advanced Micro Devices
One Lincoln Center, Suite 230
10300 Southwest Greenburg Road
Portland, Oregon 97223
Tel: (503) 245-0080

Advanced Micro Devices
Honeywell Ctr., Suite 1002
600 108th Avenue N.E.
Bellevue, Washington 98004
Tel: (206) 455-3600

MID-CALIF AREA

Advanced Micro Devices
360 N. Sepulveda, Suite 2075
El Segundo, California 90245
Tel: (213) 640-3210

Advanced Micro Devices
21600 Oxnard Street, Suite 675
Woodland Hills, California 91367
Tel: (818) 992-4155

SOUTHERN CALIF AREA

Advanced Micro Devices
5000 Birch Street
Suite 6000
Newport Beach, California 92660
Tel: (714) 752-6262

Advanced Micro Devices
9619 Chesapeake Drive #210
San Diego, California 92123
Tel: (619) 560-7030

MOUNTAIN WEST AREA

Advanced Micro Devices
14755 Preston Road, Suite 700
Dallas, Texas 75240
Tel: (214) 934-9099

Advanced Micro Devices
9020 Capital of Texas
Highway North, Suite 345
Austin, Texas 78759
Tel: (512) 346-7830

Advanced Micro Devices
2925 Briar Park #410
Houston, Texas 77042
Tel: (713) 785-9001

Advanced Micro Devices
1873 South Bellaire Street
Suite 920
Denver, Colorado 80222
Tel: (303) 691-5100

Advanced Micro Devices
40 W. Baseline Road #206
Tempe, Arizona 85283
Tel: (602) 242-4400

Advanced Micro Devices
1955 W. Grant Road #125
Tucson, Arizona 85745
Tel: (602) 792-1200

INTERNATIONAL SALES OFFICES

BELGIUM

Advanced Micro Devices
Belgium S.A.—N.V.
Avenue de Tervueren, 412, bte 9
B-1150 Bruxelles
Tel: (02) 771 99 93
TELEX: 61028
FAX: 7623712

FRANCE

Advanced Micro Devices, S.A.
Silic 314, Immeuble Helsinki
74, rue d'Arcueil
F-94588 Rungis Cedex
Tel: (01) 687.36.66
TELEX: 202053
FAX: 686.21.85

GERMANY

Advanced Micro Devices GmbH
Rosenheimer Strasse 143B
D-8000 Muenchen 80
West Germany
Tel: (089) 4114-0
TELEX: 05-23883
FAX: 406 490

GERMANY

Advanced Micro Devices GmbH
Feuerseeplatz 4/5
D-7000 Stuttgart 1
Tel: (0711) 62 33 77
TELEX: 07-21882
FAX: 625 187

Advanced Micro Devices GmbH
Wünning Weg 4
D-3108 Winsen/Aller
Tel: (05143) 5055
TELEX: 925287
FAX: (05143) 5553

HONG KONG

Advanced Micro Devices
Room 1602 World Finance Centre
South Tower
Harbour City
17 Canton Road
Tsimshatsui, Kowloon
Tel: (852) 3 695377
TELEX: 50426
FAX: (852) 123 4276

ITALY

Advanced Micro Devices S.R.L.
Centro Direzionale
Via Novara, 570
I-20153 Milano
Tel: (02) 3390541
TELEX: 315286
FAX: (39) 349 8000

JAPAN

Advanced Micro Devices, K.K.
Shinjuku Kokusai Building
6-6-2 Nishi-Shinjuku
Shinju-Ku, Tokyo 160
Tel: (03) 345-8241
TELEX: 24064
FAX: 03 (342) 5196

SWEDEN

Advanced Micro Devices AB
Box 2028
Rissneleden 144, 5tr
S-172 07 Sundbyberg
Tel: (08) 733 03 50
TELEX: 11602
FAX: 7332285

UNITED KINGDOM

Advanced Micro Devices (U.K.) Ltd.
A.M.D. House,
Goldsworth Road,
Woking,
Surrey GU21 1JT
Tel: Woking (04862) 22121
TELEX: 859103
FAX: 22179

Advanced Micro Devices (U.K.) Ltd.
The Genesis Centre
Garrett Field
Science Park South
Birchwood
Warrington WA3 7BH
Tel: Warrington (0925) 828008
TELEX: 628524
FAX: 827693

The International Standard of Quality guarantees a 0.65% AQL on all electrical parameters, AC and DC, over the entire operating range.

INT-STD-883

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450

NOTES