

quick reference guide for datapoint processors and peripherals

**A summary of assembly
and machine language instructions**

SECOND EDITION

DATAPoint CORPORATION



The Leader in Dispersed Data Processing™

REFERENCE GUIDE CONVENTIONS

The following conventions are adhered to throughout this guide.

1. All numbers not subscripted are assumed to be decimal unless noted in a table or chart
2. A "\$" appearing anywhere in this text other than character charts, shall denote an unused or unassigned item.
3. All items of information are generated for those model codes with which they are listed and can not be assumed to apply to any models not listed.

This guide is designed to aid the programmer and system analyst in using Datapoint equipment.

Under no circumstances is this guide to be used as a reference in the establishment of specifications or performance criteria. The appropriate product specification, reference manual, or software user's guide should be consulted for that purpose.

Suggestions and additions will be gratefully accepted. Write the Software Support Group, 9725 Datapoint Drive, San Antonio, Texas 78284.

DEVICE ADDRESS ASSIGNMENTS

DEVICE	BINARY	OCTAL
Asynchronous Communications Adaptor	11010010	322
Cassette Deck	11110000	360
9402 - DSREMOTE #2	11110100	350
9402 - DSREMOTE #3	11100100	344
9402 - DSREMOTE #4	11100010	342
9402 - DSREMOTE #5	11011000	330
9402 - DSREMOTE #6	11010100	324
9481 - Multifunction	11010001	321
Communications Adaptor #1		
9481 - Multifunction	11001100	314
Communications Adaptor #2		
9481 - Multifunction	11001010	312
Communication Adaptor #3		
9481 - Multifunction	11001001	311
Communications Adaptor #4		
Local Printer #1	11000011	303
Local Printer #2	11000101	305
Local Printer #3	11000110	306
Local Printer #4	10111000	270
Magnetic Tape	10110100	264
Magnetic Tape	10110010	262
Magnetic Tape	10110001	261
Magnetic Tape	10101100	254
Magnetic Tape	10101010	252
Magnetic Tape	10101001	251
9404 Synchronous Combox #2	10100110	246
9404 Synchronous Combox #1	10100101	245
9404 Synchronous Combox #3	10100011	243
Reserved	10011100	234
Reserved	10011010	232
Reserved	10011001	231
9420 Parallel Interface	10010110	226
9420 Parallel Interface	10010101	225
9420 Parallel Interface	10010011	226
9420 Parallel Interface	10001110	216
9420 Parallel Interface	10001101	215
9420 Parallel Interface	10001011	213
Card Reader	10000111	207
Cartridge Disk	01111000	170
Cartridge Disk	01110100	164
Cartridge Disk	01110010	162
Cartridge Disk	01110001	161
9402 - DSREMOTE #7	01101100	154
9402 - DSREMOTE #8	01101010	152
9462 - Multi Port Adaptor #1	01101001	151
9402 - DSREMOTE #9	01100110	146
9402 - DSREMOTE #10	01100101	145
9402 - DSREMOTE #11	01100011	143
Servo Printer	01011100	134
Servo Printer	01011010	132
Servo Printer	01011001	131
Mass Storage Disk	01010110	126
Mass Storage Disk	01010101	125
Mass Storage Disk	01010011	123
Mass Storage Disk	01001110	116
Mass Storage Disk	01001101	115
Mass Storage Disk	01001011	113
Mass Storage Disk	01000111	107
Diskette	00111100	074
Diskette	00111010	072
Diskette	00111001	071

DEVICE ADDRESS ASSIGNMENTS
(continued)

DEVICE	BINARY	OCTAL
9402 - DSREMOTE #12	00110110	066
9402 - DSREMOTE #13	00110101	065
9402 - DSREMOTE #14	00110011	063
9402 - DSREMOTE #15	00101110	056
9462 - Multi Port Adaptor #2	00101101	055
9402 - DSREMOTE #16	00101011	053
Unassigned	00100111	047
Unassigned	00011110	036
Unassigned	00011101	035
Unassigned	00011011	033
Unassigned	00010111	027
Unassigned	00001111	017

EXTERNAL COMMANDS

COMMAND WORD - EX (EXPRESSION)

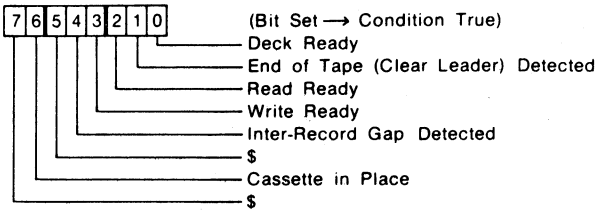
ALL DEVICES			
EXPRESSION	COMMAND NUMBER	OPERATION CODE (OCTAL)	COMMAND FUNCTION
ADR	1	121	Address Device
STATUS	2	123	Sense Status
DATA	3	125	Sense Data
WRITE	4	127	Write Strobe
COM1	5	131	Command 1
COM2	6	133	Command 2
COM3	7	135	Command 3
COM4	8	137	Command 4
\$	9	141	\$
\$	10	143	\$
\$	11	145	\$
\$	12	147	\$
BEEP	13	151	Beep
CLICK	14	153	Click
CASSETTE TAPE DECKS			
EXPRESSION	COMMAND NUMBER	OPERATION CODE (OCTAL)	COMMAND FUNCTION
DECK1	15	155	Select Rear Deck
DECK2	16	157	Select Front Deck
RBK	17	161	Read Block
WBK	18	163	Write Block
\$	19	165	\$
BSP	20	167	Backspace One Block
SF	21	171	Slew Forward
SB	22	173	Slew Backward
REWIND	23	175	Rewind Tape
STOP	24	177	Stop Tape

NOTE: These device commands are in reference to ANSI cassette decks.

EXTERNAL COMMANDS

CASSETTE TAPE DECKS

STATUS WORD - EX STATUS



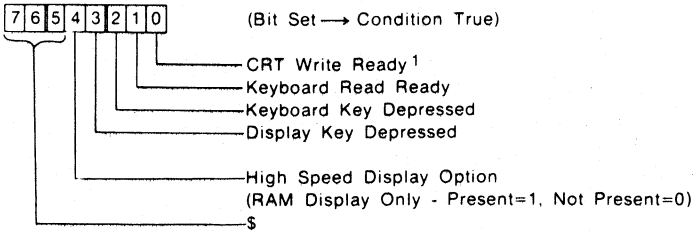
TAPE UNIT PHYSICAL SPECIFICATIONS (ANSI DECKS)

Density	47 characters/inch
Speed	7.5 inches/second
Recording Rate	350 character/second
Capacity	115,000 characters (Typical)
Start/Stop Time	
(Inter-Record Gap)	305 msec.
Start/Stop Distance	
(Inter-Record Gap)	2.2 inches
Rewind Speed	90 inches/second
Rewind Time (maximum)	40 seconds
Characters Transfer Time	2.8 msec.

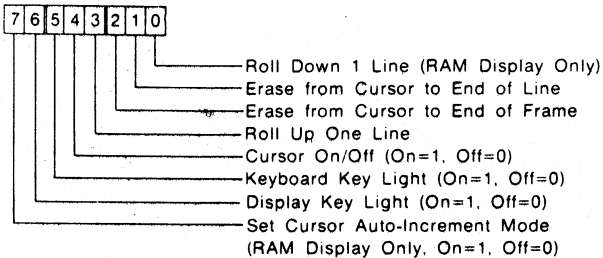
EXTERNAL COMMANDS

CRT/KEYBOARD

STATUS WORD - EX STATUS



CONTROL WORD - EX COM1



CONTROL WORD - EX COM2

Horizontal Cursor Position
(Decimal 0-79, 0-177₈, Starting at left of screen)

CONTROL WORD - EX COM3

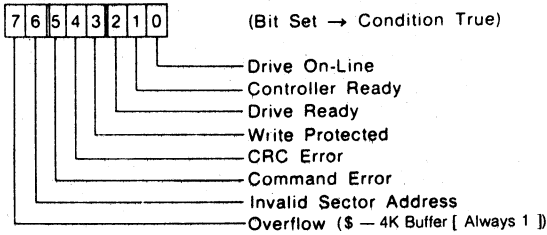
Vertical Cursor Position
(Decimal 0-11, 0-13₈, Starting at top of screen)

NOTE 1: CRT "Write Ready" is valid only if cursor is positioned to a valid screen position.

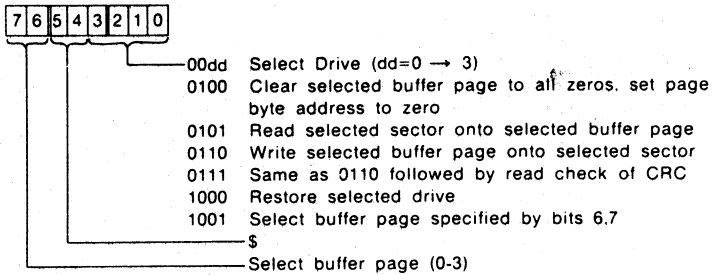
EXTERNAL COMMANDS

9350 DISK CONTROLLER/DRIVE

STATUS WORD - EX STATUS



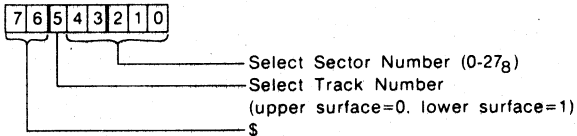
CONTROL WORD - EX COM1



CONTROL WORD - EX COM2

Select Cylinder Number (0-312_g)

CONTROL WORD - EX COM3



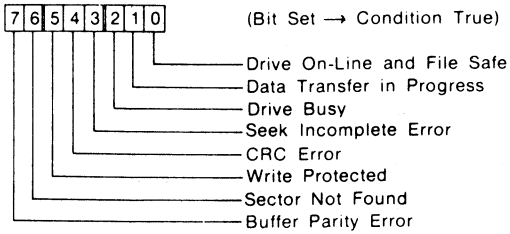
CONTROL WORD - EX COM4

Select Buffer Page Byte Address (0-256)

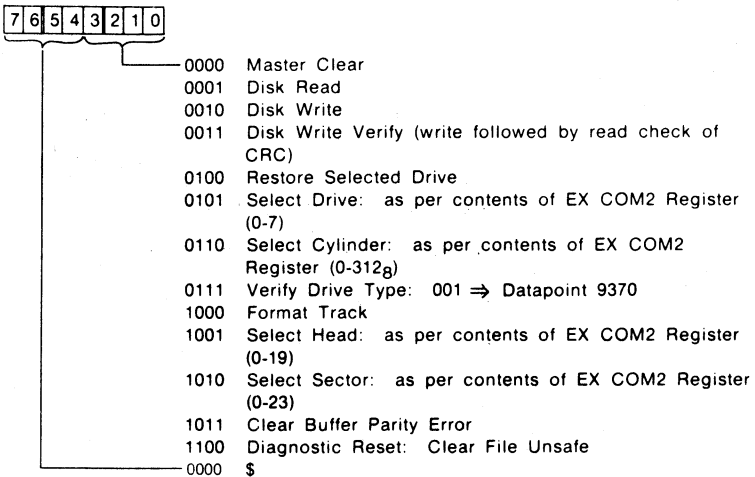
EXTERNAL COMMANDS

9370 DISK CONTROLLER/DRIVE

STATUS WORD - EX STATUS



CONTROL WORD - EX COM1



CONTROL WORD - EX COM2

Selects drive, cylinder, sector and head in conjunction with proper EX COM1 command

CONTROL WORD - EX COM3

Select buffer page (0-15)

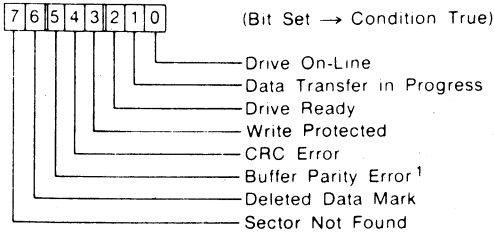
CONTROL WORD - EX COM4

Select buffer page byte address

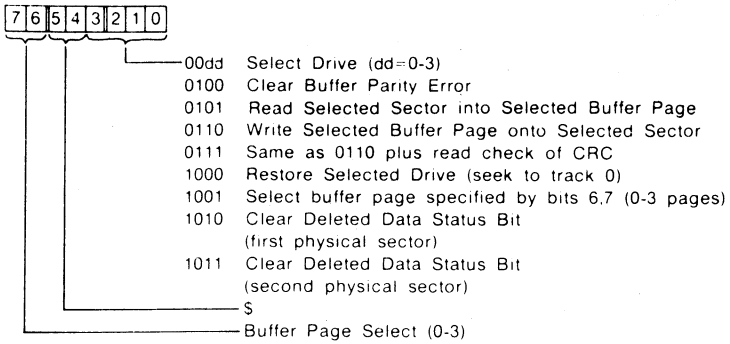
EXTERNAL COMMANDS

9380 DISK CONTROLLER/DRIVE

STATUS WORD - EX STATUS



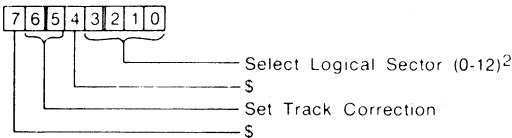
CONTROL WORD - EX COM1



CONTROL WORD - EX COM2

Select Track and Seek (0-76)

CONTROL WORD - EX COM3



CONTROL WORD - EX COM4

Select Buffer Page Byte Address (0-256)

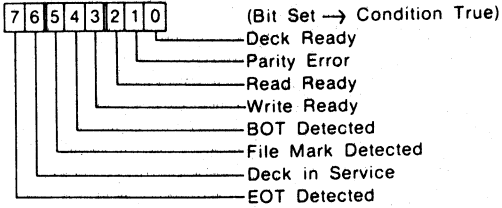
NOTE 1: Buffer memory parity will be in error on power-up until buffer is written in.

NOTE 2: The 9380 contains two physical sectors for each logical sector

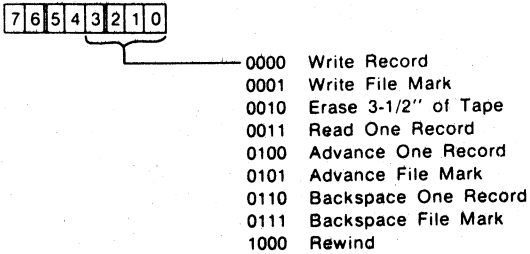
EXTERNAL COMMANDS

7 AND 9 TRACK TAPE TRANSPORTS (9550, 9551)

STATUS WORD - EX STATUS



CONTROL WORD - EX COM1



CONTROL WORD - EX COM2

Write Buffer Content on Tape

CONTROL WORD - EX COM3

Clear Buffer

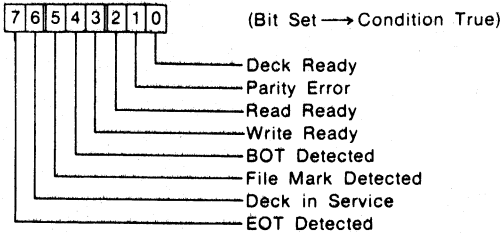
CONTROL WORD - EX COM4

\$

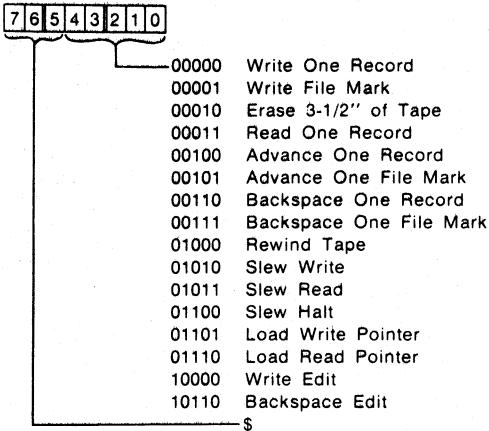
EXTERNAL COMMANDS

1600 BPI MAGNETIC TAPE SYSTEM (9580)

STATUS WORD - EX STATUS



CONTROL WORD - EX COM1



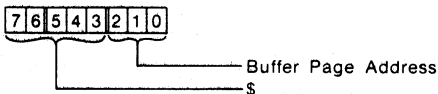
CONTROL WORD - EX COM2

Write Buffer Contents on Tape

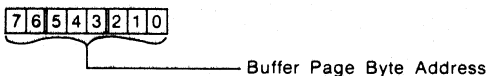
CONTROL WORD - EX COM3

Clear Buffer

CONTROL WORD - EX COM4 (First Output)



CONTROL WORD - EX COM4 (Second Output)

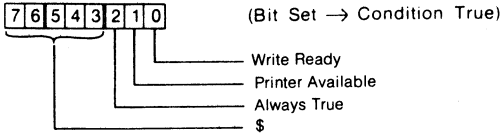


EXTERNAL COMMANDS

PRINTERS (9242, 9260, 9280, 9291, 9292)

LOCAL PRINTERS (OTHER THAN SERVO PRINTERS)

STATUS WORD - EX STATUS



CHARACTER TRANSMISSION - EX WRITE

See - "Character Transmission and Translation Table"

Printer Control Codes¹

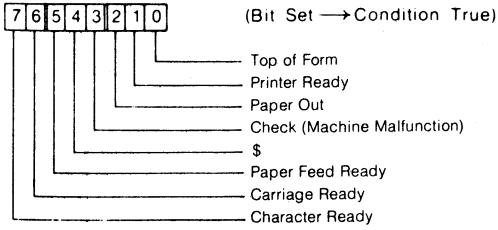
- 012₈ Line Feed
- 013₈ Vertical Tab (Centronics only)
- 014₈ Form Feed
- 015₈ Carriage Return
- 016₈ Elongated Print (Centronics only)

NOTE 1: Printer Control Codes are in addition to printing the buffer (i.e. data followed by a 014 will print the buffer and form feed).

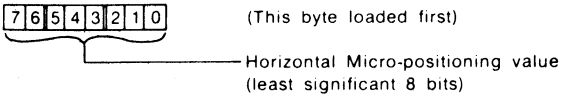
EXTERNAL COMMANDS

SERVO PRINTER (9250)

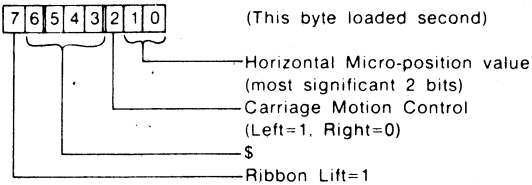
STATUS WORD - EX STATUS



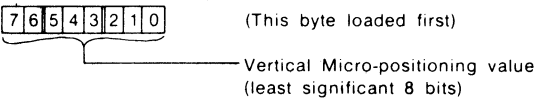
CONTROL WORD - EX WRITE (Used in conjunction with EX COM3)



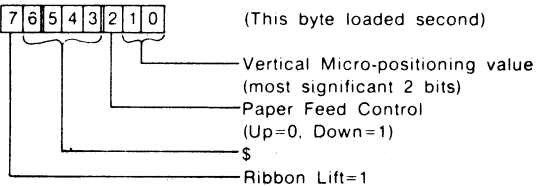
CONTROL WORD - EX COM1 (Used in conjunction with EX COM3)



CONTROL WORD - EX WRITE (Used in conjunction with EX COM4)



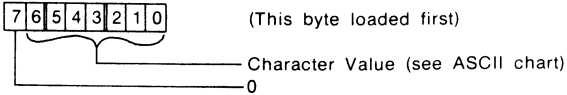
CONTROL WORD - EX COM1 (Used in conjunction with EX COM4)



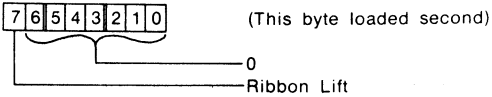
EXTERNAL COMMANDS

SERVO PRINTER 9250 (Cont.)

CONTROL WORD - EX WRITE (Used in conjunction with EX COM2)



CONTROL WORD - EX COM1 (used in conjunction with EX COM2)



CONTROL WORD - EX COM2

Generates Print Strobe

(Used in conjunction with EX WRITE and EX COM1)

CONTROL WORD - EX COM3

Generates Carriage Motion Strobe

(Used in conjunction with EX WRITE and EX COM1)

CONTROL WORD - EX COM4

Generates Paper Feed Strobe

(Used in conjunction with EX WRITE and EX COM1)

CONTROL WORD - EX DATA

Generates Form Feed Command

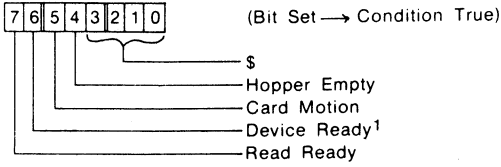
CONTROL WORD - EX STATUS

Generates "RESTORE" Command

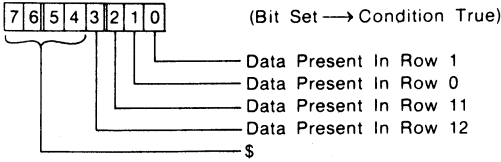
EXTERNAL COMMANDS

CARD READER (9504)

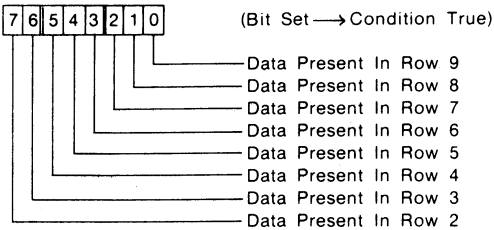
STATUS WORD - EX STATUS (FIRST INPUT)



STATUS WORD - EX STATUS (SECOND INPUT)



DATA WORD - EX DATA



CONTROL WORD - EX COM1

\$

CONTROL WORD - EX COM2

Card Feed Command

CONTROL WORD - EX COM3

Motor 'ON' Command

CONTROL WORD - EX COM4

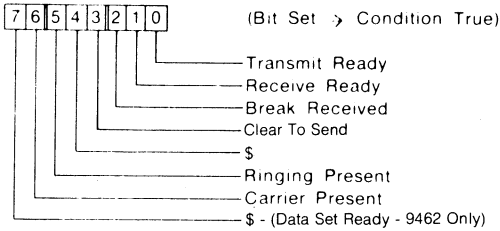
Motor 'OFF' Command

NOTE 1: Setting of this bit may be delayed as much as 10 seconds.

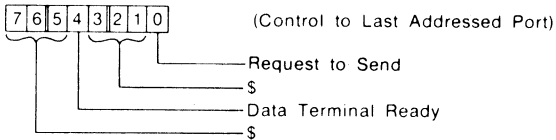
EXTERNAL COMMANDS

MULTIPLE PORT COMMUNICATIONS ADAPTOR (9460) (9462)

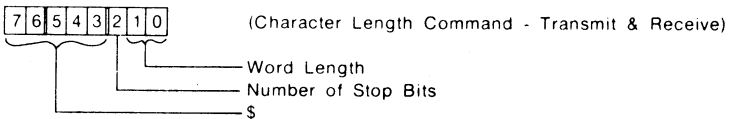
STATUS WORD¹ - EX STATUS



CONTROL WORD - EX COM1



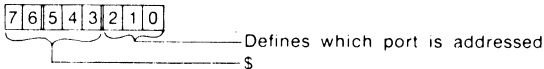
CONTROL WORD - EX COM2



TRANSMIT AND RECEIVE CHARACTER FORMATS

CONTROL BIT POSITIONS 2 1 0 - OCTAL	START BITS	INFORMATION BITS	STOP BITS	CODE BIT POSITIONS
				7 6 5 4 3 2 1 0
000 - 0	1	5	1	XXX54321
001 - 1	1	6	1	XX654321
010 - 2	1	7	1	X7654321
011 - 3	1	8	1	87654321
100 - 4	1	5	2	XXX54321
101 - 5	1	6	2	XX654321
110 - 6	1	7	2	X7654321
111 - 7	1	8	2	87654321

CONTROL WORD - EX COM3



CONTROL WORD - EX COM 4

\$

CONTROL WORD - EX WRITE

Transfers character in A-register to the currently selected port for transmission.

NOTE1: Until a port is selected, the STATUS word is a logical OR value of the status of all eight ports.

EXTERNAL COMMANDS

ASYNCHRONOUS COMMUNICATIONS ADAPTORS

STATUS WORD - EX STATUS (Bit Set → Condition True)

Status Word Bits	Model Number				
	9400	9401	9402	9403	9410
0	Transmit Ready	Same	Same	Same	Present Next Digit
1	Receive Ready	Same	Same	Same	Data Line Occupied
2	Break Received	Same	Same	Same	Distant Station Connected
3	Clear to Send (CB)	Same	Same	\$	Abandon Call
4	Reverse Carrier Present (SB)	\$	Same	\$	Power Indication
5	Ringing Present (CE)	Same	Same	\$	Standby Indicator
6	Main Carrier Present (CF)	Carrier Present (CAR)	Same	\$	\$
7	\$	Data Coupler Ready (DCR)	Data Coupler Ready (DCR)	\$	\$

EXTERNAL COMMANDS

ASYNCHRONOUS COMMUNICATIONS ADAPTORS (Cont.)

CONTROL WORD - EX COM1

EX COM1 Word Bits	Model Number				
	9400	9401	9402	9403	9410
0	Request to Send	Same	Same	\$	Data Terminal Ready
1	Invert Received Data Line	Same	Same	Same	Request to Send Mn. Channel
2	Supervisory Channel On	\$	Same	\$	Sig. Rate Sel./Transmit Freq. Sel.
3	Invert Transmitted Data Line	Same	Same	Same	Select Stand-By
4	Data Terminal Ready	Off-Hook	Off-Hook	\$	Receiver Cut-off Main Channel
5	\$	\$	Send 2025 Hz	\$	Return to Non-Data Mode
6	\$	Originate=1 ¹ Answer=0	Originate=1 Answer=0	\$	Request to Send Rev. Channel
7	\$	Send Dial Pulses	Send Dial Pulses	\$	Receiver Cut-Off Rev. Channel

CONTROL WORD - EX COM2 (1st Execution)

Receive Time Base - least significant byte (see charts)

CONTROL WORD - EX COM2 (2nd Execution)

Receive Time Base - most significant byte (see charts)

CONTROL WORD - EX COM3 (1st Execution)

Transmit Time Base - least significant byte (see charts)

CONTROL WORD - EX COM3 (2nd Execution)

Transmit Time Base - most significant byte (see charts)

NOTE1: 2025 Hz is used for transmission if this bit is 0.

EXTERNAL COMMANDS

ASYNCHRONOUS COMMUNICATIONS ADAPTORS (Cont.)

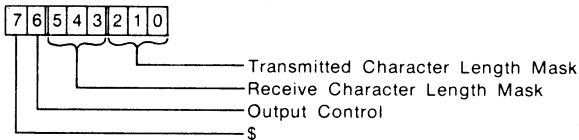
TIME BASE CHART (EIA) 9400, 9401, 9402, 9403

RECEIVE TRANSMIT	EX COM2 EX COM3	EX COM2 EX COM3
BIT RATE	1st MASK WORD (Octal)	2nd MASK WORD (Octal)
100	375(dialing)	000
110	375	106
220	376	243
440	377	121
150	376	000
300	377	000
600	377	200
1200	377	300
1800	377	325
2400	377	340

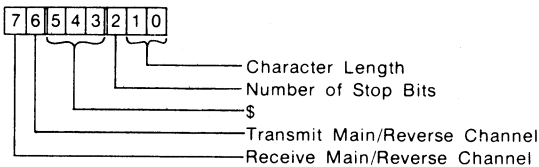
TIME-BASE CHART (CITT) 9410

RECEIVE TRANSMIT	EX COM2 EX COM3	EX COM2 EX COM3
BIT RATE	1st MASK WORD (Octal)	2nd MASK WORD (Octal)
50	030	000
75	020	000
100	014	000
200	006	000
110	365	027
220	372	214
440	375	106
150	370	000
300	374	000
600	376	000
1200	377	000
2400	377	200
4800	377	300
9600	377	340

CONTROL WORD - EX COM4 (9400, 9401, 9402, 9403)

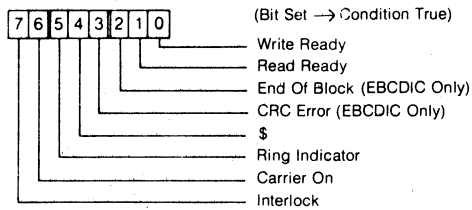


CONTROL WORD - EX COM4 (9410)

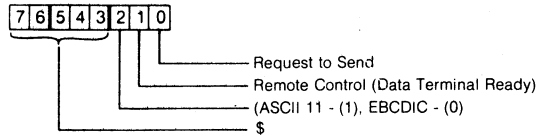


9404 - SYNCHRONOUS COMMUNICATIONS ADAPTOR

STATUS WORD - EX STATUS



CONTROL WORD - EX COM1



CONTROL WORD - EX COM2

Write Single 'DLE' ('DLE' Character
Loaded In The A-Register) - ENCDIC Only)

CONTROL WORD - EX COM3

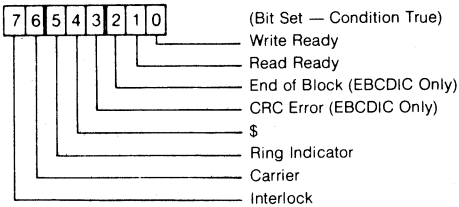
New SYNC

CONTROL WORD / EX COM4

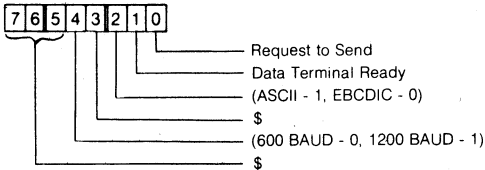
\$

9405 - SYNCHRONOUS COMMUNICATIONS ADAPTOR
(9405 with 712 Clock Option)

STATUS WORD - EX STATUS



CONTROL WORD - EX COM1



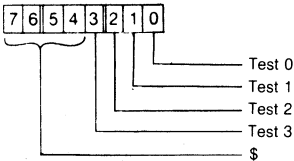
CONTROL WORD - EX COM2

Write Single 'DLE' (DLE) Character
 Loaded In The A-Register - EBDIC Only)

CONTROLWORD - EX COM3

New Sync

CONTROL WORD - EX COM4



NOTE 1: The toggle switch on the 9405 board
 must be flipped for 600/1200 BAUD operation.

2200 MACHINE INSTRUCTIONS
(ASSEMBLER 4, DOSASM5, SNAP/1, SNAP/2)

FUNCTION (Mnemonic Code)	OPERATION (OP) Code Expressed in Octal	DESCRIPTION (Definition of Function)	TIMING (in usec) *
Lr _d (exp)	0n _d 6,vvv	Load Immediate	3.2
Lr _d f _s	3n _d n _s	Load	3.2 ¹
AD (exp)	004,vvv	Add Immediate	4.8
ADr _s	20n _s	Add	3.2 ¹
AC (exp)	014,vvv	Add with Carry Immediate	4.8
ACr _s	21n _s	Add with Carry	3.2 ¹
SU (exp)	024,vvv	Subtract Immediate	4.8
SUr _s	22n _s	Subtract	3.2 ¹
SB (exp)	034,vvv	Subtract with Borrow Immediate	4.8
SBr _s	23n _s	Subtract with Borrow	3.2 ¹
ND (exp)	044,vvv	And Immediate	4.8
NDr _s	24n _s	And	3.2 ¹
OR (exp)	064,vvv	Or Immediate	4.8
ORr _s	26n _s	Or	3.2 ¹
XR (exp)	054,vvv	Exclusive - Or Immediate	4.8
XRr _s	25n _s	Exclusive - Or	3.2 ¹
CP (exp)	074,vvv	Compare Immediate	4.8
CPr _s	27n _s	Compare	3.2 ¹
SRC	012	Shift Right Circular	3.2
SLC	002	Shift Left Circular	3.2
JMP (adr)	104,lsb,msb	Unconditional Jump	6.4
JTc (adr)	1p0,lsb,msb	Jump on True Flag	4.8 ²
JFc (adr)	1m0,lsb,msb	Jump on False Flag	4.8 ²
CALL (adr)	106,lsb,msb	Unconditional Call	6.4
CTc (adr)	1p2,lsb,msb	Call on True Flag	4.8 ²
CFc (adr)	1m2,lsb,msb	Call on False Flag	4.8 ²
RET	007	Unconditional Return	3.2
RTc	0p3	Return on True Flag	1.6 ²
RFc	0m3	Return on False Flag	1.6 ²
BETA*	020	Register and F/F Mode Swap	3.2
ALPHA*	030	Register and F/F Mode Swap	3.2
PUSH*	070	Address onto Stack	3.2
POP*	060	Address from Stack	4.8
DI*	040	Disable Interrupt	3.2
EI*	050	Enable Interrupt	3.2
NOP	300	No Operation	3.2
HALT	377	Halt	--
INPUT	101	Input	9.6
EX	See Tables	External Command (see external command tables)	9.6

r_s =source register
r_d =destination register
(exp)=one-byte expression
(adr)=two-byte address
c=condition flag
n_d,n_s=register reference
number
vvv=expression value
lsb=least significant byte
of address

msb=most significant byte
of address

*Version II 2200 only

1. add 1.6 usec if
memory reference
2. add 1.6 usec if transfer
occurs

p, m=condition code reference

REGISTER REFERENCE TABLE

r	n
A	0
B	1
C	2
D	3
E	4
H	5
L	6
*X	7
M	7

M=memory reference,
memory location specified
by HL*(or currently
selected register pair)

*5500 only

CONDITION (Flip-Flop) CODE REFERENCE TABLE

c	m	p
C (Carry)	0	4
Z (Zero)	1	5
S (Sign)	2	6
P (Parity) (true=odd parity)	3	7

REGISTER CODES (5500 ONLY)

r	[r] (register select opcode)	[pr] (reg. sel. for paged instructions)
A	no code (implicit)	105
B	111	114
C	062	124
D	113	134
E	174	144
H	115	154
L	176	164
X	117	no code (cannot be used)

REGISTER PAIR CODES (5500 ONLY)

rp	[rp] - register pair select code
HL	176 (implicit)
BC	062
DE	174
XA	022

5500 MACHINE INSTRUCTIONS
(SNAP/1, SNAP/2)

(1 of 4)

FUNCTION (Mnemonic Code)	OPERATION (OP) Code Expressed in Octal	DESCRIPTION (Definition of Function)	TIMING (in usec) *
L _r d (exp)	0n _d 6,vvv	Load Immediate	2.0
L _r d _r s	3n _d n _s	Load	1.2 (2.6)
AD (exp)	004,vvv	Add Immediate	2.0
AD _r s	20n _s	Add	1.4 (2.6)
AC (exp)	014,vvv	Add with Carry Immediate	2.0
AC _r s	21n _s	Add with Carry	1.4 (2.6)
SU (exp)	024,vvv	Subtract Immediate	2.0
SU _r s	22n _s	Subtract	1.4 (2.6)
SB (exp)	034,vvv	Subtract with Borrow Immediate	2.0
SB _r s	23n _s	Subtract with Borrow	1.4 (2.6)
ND (exp)	044,vvv	And Immediate	2.0
ND _r s	24n _s	And	1.4 (2.6)
OR (exp)	064,vvv	Or Immediate	2.0
OR _r s	26n _s	Or	1.4 (2.6)
XR (exp)	054,vvv	Exclusive - Or Immediate	2.0
XR _r s	25n _s	Exclusive - Or	1.4 (2.6)
CP (exp)	074,vvv	Compare Immediate	1.8
CP _r s	27n _s	Compare	1.2 (2.4)
SRC	012	Shift Right Circular	1.4
SLC	002	Shift Left Circular	1.4
JMP (adr)	104,lsb,msb	Unconditional Jump	2.8
JTc (adr)	1p0,lsb,msb	Jump on True Flag	1.4 (3.0)
JFc (adr)	1m0,lsb,msb	Jump on False Flag	1.4 (3.0)
CALL (adr)	106,lsb,msb	Unconditional Call	3.0
CTc (adr)	1p2,lsb,msb	Call on True Flag	1.6 (3.2)
CFc (adr)	1m2,lsb,msb	Call on False Flag	1.6 (3.2)
RET	007	Unconditional Return	1.8
RTc	0p3	Return on True Flag	1.0 (1.8)
RFc	0m3	Return on False Flag	1.0 (1.8)
BETA	020	Register and F/F Mode Swap	1.2
ALPHA	030	Register and F/F Mode Swap	1.0
PUSH	070	Address onto Stack	2.0
POP	060	Address from Stack	2.2
DI	040	Disable Interrupt	1.2
EI	050	Enable Interrupt	1.0
NOP	300	No Operation	1.2
HALT	377	Halt	--
INPUT	101	Input	5.0
EX	See tables	External Command (see external command tables)	9.2

r_d=source register

r_s=destination register

(exp)=one-byte expression

(adr)=two-byte address

c=condition flag

n_d,n_s=register reference number

vvv=expression value

lsb=least significant byte
of address

msb=most significant byte
of address

*value in parenthesis is for
instruction making a memory
reference or causing a transfer.

5500 MACHINE INSTRUCTIONS
(SNAP/1, SNAP/2)

(2 of 4)

FUNCTION (Mnemonic Code)	OPERATION (OP) Code Expressed in Octal	DESCRIPTION (Definition of Function)	TIMING (in usec)
Lr _d M rp	[rp].3n _d 7	Load register from memory, memory address in rp	3.4
LMr _s rp	[rp].37n _s	Load memory from register, memory address in rp	3.4
(op)r _s r _d	[r _d].[*]	Arithmetic or logical operation to other than A register	+1.0
(op)r _d (exp)	[r _d].[*]vvv	Immediate arithmetic or logical operation to other than A register	+1.0
SLCr _d	[r _d].002	Shift left circular, other than A reg	2.4
SRCr _d	[r _d].012	Shift right circular, other than A reg	2.4
SRE	032	Shift right extended	1.4
SREr _d	[r _d].032	Shift right extended, other than A reg	2.4
INr _d	[r _d].101	Input, to other than A register	6.0
EXr _s	[r _d].[ext]	External command, from other than A reg (see external command tables for [ext])	10.2
PIN	103	Input with parity testing	5.4
PINr _d	[r _d].103	Input with parity testing, to other than A register	6.4
PUSH rp	[rp].070	Address onto stack, from rp	2.8
PUSH (adr)	051.lsb.msb	PUSH Immediate	2.8
POP rp	[rp].060	Address from stack, into rp	3.0
BT	021	Block transfer	<820
BTR	111.021	Block transfer reverse	<820
BFAC	011	Binary field add with carry	<50
BFSB	031	Binary field subtract with borrow	<50
BCP	041	Block Compare	<722
DFAC	111.041	Decimal field add with carry	<79
DFSB	062.041	Decimal field subtract with borrow	<70
BFSL	075	Binary field shift left	<39
BFSR	111.075	Binary field shift right	<37
MIN	061	Multiple In, DMA - type command, I/O device to memory starting at HL	<155 (8.4/byte)
MOUT	071	Multiple Out, DMA - type command, I/O device from memory starting at HL	<161 (8.8/byte)
STKS	065	Stack Store, save stack in memory	<40
STKL	111.065	Stack Load, restore stack from memory	<40
REGS	055	Register Store, save registers in memory, descending from top-of-stack address	13.2
REGL	111.055	Register Load, restore registers from memory, descending from address in HL	12.0

rp=register pair

[rp]=register pair select code

[r]=register select code

(op)=arithmetic or logical operator:

AD,AC,SU,SB,ND,OR,XR,CP

[*]=op code for (op) from page 1

TIMING NOTE:

+ prior to time indicates
time to be added to timing for
equivalent instruction from page 1

< prior to time indicates
maximum possible time. These
instructions are extremely complex;
check 5500 reference.

5500 MACHINE INSTRUCTIONS
(SNAP/1, SNAP/2)

(3 of 4)

FUNCTION (Mnemonic Code)	OPERATION (OP) Code (Expressed in Octal)	DESCRIPTION (Definition of Function)	TIMING (in usec)
CCSR _d	[_d] .042	Condition Code Save in _d , add _d to itself to restore conditions	3.0
Increment Register Pair Instructions			
INCP HL	015	HL by 1	2.8
INCP HL.2	117.015	HL by 2	3.8
INCP HL.A	017	HL by contents of A	3.0
INCP BC	062.015	BC by 1	3.6
INCP BC.2	113.015	BC by 2	3.8
INCP BC.A	062.017	BC by contents of A	3.8
INCP DE	174.015	DE by 1	3.6
INCP DE.2	115.015	DE by 2	3.8
INCP DE.A	174.017	DE by contents of A	3.8
Decrement Register Pair Instructions			
DECP HL	035	HL by 1	2.8
DECP HL.2	117.035	HL by 2	3.8
DECP HL.A	037	HL by contents of A	3.0
DECP BC	062.035	BC by 1	3.6
DECP BC.2	113.035	BC by 2	3.8
DECP BC.A	062.037	BC by contents of A	3.8
DECP DE	174.035	DE by 1	3.6
DECP DE.2	115.035	DE by 2	3.8
DECP DE.A	174.037	DE by contents of A	3.8
Double Load Instructions			
DL _{rp_d} . _{rp_s}	--	load _{rp_d} from memory locations pointed to by _{rp_s} .lsb first	--
DL DE.HL	047		3.4
DL BC.HL	111.047		5.2
DL BC.BC	062.047		4.6
DL BC.DE	113.047		5.0
DL DE.BC	174.047		4.6
DL DE.DE	115.047		5.0
DL HL.BC	176.047		4.6
DL HL.DE	117.047		5.0
DL HL.HL	057		4.0
NOJ	045	NOP Jump, skip next two bytes	1.4
SC	067	System Call, call 0170025	1.8
UR	111.012	User Return	2.0
STL	077	Sector Table Load	27.4
BP	052	Break Point, call 0170030 (DEBUG)	1.8

_d=destination register
_{rp_d}=destination register pair
_{rp_s}=source register pair
lsb=least significant byte
[_d]=register select code

5500 MACHINE INSTRUCTIONS
(SNAP/1, SNAP/2)

(4 of 4)

FUNCTION (Mnemonic Code)	OPERATION (OP) Code Expressed in Octal	DESCRIPTION (Definition of Function)	TIMING (in usec)
DS rp_s, rp_d	--	Double Store Instructions store contents of rp_s into memory locations pointed to by rp_d , lsb first	--
DS DE,HL	027		3.4
DS BC,HL	111,027		5.2
DS BC,DE	113,027		5.0
DS DE,BC	174,027		4.6
DS HL,BC	176,027		4.6
DS HL,DE	117,027		5.0
PL $r_d(\text{exp})$	$[pr], \text{lsb}$	Paged Load, load r_d from memory location given by lsb, msb in X	3.0
PS $r_s(\text{exp})$	$[pr+2], \text{lsb}$	Paged Store, store r_d into memory	3.0
DPL $rp(\text{exp})$	$[r], [pr], \text{lsb}$	Double Paged Load, load rp from memory location and location +1 ($[r]$ is for first register of pair, $[pr]$ for second) ($rp \neq XA$)	4.8
DPS $rp(\text{exp})$	$[r], [pr+2], \text{lsb}$	Double Paged Store, store rp into memory (similar to DPL)	4.8
Index Instructions: index is a two-byte value in memory, X contains msb and (i) gives lsb of location of the index. (dsp) is used to modify the value in the index.			
INCI $(dsp), (i)$	$005, \text{lsb}, [i]$	Increment Index by lsb of (dsp)	7.6
DECI $(dsp), (i)$	$025, \text{lsb}, [i]$	Decrement Index by lsb of (dsp)	7.8
INCI $*(dsp), (i)$	$111, 005, \text{lsb}, \text{msb}, [i]$	Increment Index by msb, lsb of (dsp)	9.6
DECI $*(dsp), (i)$	$111, 025, \text{lsb}, \text{msb}, [i]$	Decrement Index by msb, lsb of (dsp)	9.8
LFII $rp, (dsp), (i)$	$[rp], 005, \text{lsb}, [i]$	Load From Index Incremented, add lsb of (dsp) to index value and save result in rp (does not modify value in index)	7.4
LFID $rp, (dsp), (i)$	$[rp], 025, \text{lsb}, [i]$	Load From Index Decrementated, subtract lsb from index and save result in rp (does not modify value in index)	7.6
LFII $rp, *(dsp), (i)$	$[srp], 005, \text{lsb}, \text{msb}, [i]$	Load From Index Incremented, same as LFII above, but using msb, lsb of (dsp)	8.4
LFID $rp, *(dsp), (i)$	$[srp], 025, \text{lsb}, \text{msb}, [i]$	Load From Index Decrementated, same as LFID above, but using msb, lsb of (dsp)	8.6

$[pr]$ =register select code for
paged instructions
(i)=expression for lsb of index
address
(dsp)=expression for displacement
of index instructions
 $[i]$ =lsb value of (i)

$[srp]$ =special register pair select
code for index instructions:
BC=113
DE=115
HL=117

ASSEMBLER DIRECTIVES

ASSEMBLER 4, DOSASM5, SNAP/1, and SNAP/2		
(L)	EQU (e)	Set value of label to (e)
	SET (e)	Set and use ABSOLUTE PAB (ASSEMBLER 4: set LC)
	SKIP (e)	Increment AC and LC by (e)
	TP	Tabulate AC and LC to next page boundary
	TM (e)	Tabulate page if less than (e) bytes in present page
	DC (e)	Generate 1-byte values for expressions (1 byte per character for string expressions)
	DA (e)	Generate 2-byte values for expressions
	RPT (e)	Repeat next line of code (e) times
	END (e)	End assembly pass; (e) is program transfer address
	LIST (e)	Set assembly listing control flags Comment line (. in first column)
	+	Form feed, then print comment line
	*	Form feed if within 2" of end of page (comment line)
DOSASM5, SNAP/1, SNAP/2		
	INC (e)	Include source file named by (e)
	LOC (e)	Set LC to (e) and turn L flag on
	LOC *	Set LC to AC and turn L flag off
(L)	ORG (e)	Set first and current word address of a new PAB named (L)
	USE (e)	Use PAB (e), set AC to current word address of PAB
	USE *	Revert to use of last PAB used
	ERR	Produce a P error
	IFnn (e)	Turn assembly off if condition "nn" is not met. Condition test compares first field of expression to second field. If second field not given, assume 0. Conditions: EQ - equal GT - greater than LT - less than NE - not equal GE - greater than or equal LE - less than or equal Z - field 1 zero NZ - field 1 not zero C - field 1 zero (clear) S - field 1 not zero (set)
	XIF	Turn assembly back on if it has been turned off
SNAP/1, SNAP/2		
	TITLE	Page eject and print following line as title
	MLIB (e)	Include macro library (e)
	MACRO	Macro definition follows
	MEND	End of macro definition
	ALIGN (e)	Increment AC and LC to next memory location that is a multiple of (e) ((e)=2^n)
SNAP/2		
(L)	PROG	Label is name for following program module

AC=Address Counter
LC=Location Counter

(L)=label required
(e)=expression (allowed or required)

LABELS

Labels consist only of alphanumeric characters and \$. A label must begin with an alpha character. Special terminating characters (not part of the label) indicate special qualities for the label.

LABEL LENGTH (characters)	OVER-LENGTH LABEL ACTION	TERMINATING CHARACTER	CHARACTERISTIC ASSIGNED
ASSEMBLER 4			
6	uses first 6 characters	none	
DOSASM5			
6	uses first 6 characters	* =	program entry point redefinition of label value
SNAP/1			
8	uses first 7 and last 1 characters	* =	program entry point redefinition of label value
SNAP/2			
8	uses first 7 and last 1 characters	* = :	program entry point redefinition external definition

EXPRESSIONS

Numeric expressions use 16-bit two's complement values. If the instruction requires only one byte, the msb of the expression is discarded.

Expression evaluation is strictly left to right, all operators having equal precedence. SNAP/1 and SNAP/2 allow the use of parentheses to modify order of evaluation per normal algebraic convention.

Binary Operators

+ add
 - subtract
 * multiply
 / integer division*
 .AND. logical AND*
 .OR. logical OR*
 .XOR. logical exclusive-OR*

Unary Operators

< shift left number of
 places indicated by
 next value
 > shift right number of
 places indicated by
 next value
 - negation
 * set star flag

*Not available on ASSEMBLER 4

Strings can be included in all expressions. A string is delimited by apostrophes. The value of a character in a string is the ASCII value for the character with the parity bit (bit 7) always zero.

NOTE: only the DC directive allows strings more than one character long. For this case, one byte of code is generated for each character.

ASSEMBLER PSEUDO-INSTRUCTIONS
called **MACROS** in **ASSEMBLER 4** and **DOSASM5**

ASSEMBLER 4, DOSASM5, SNAP/1, and SNAP/2		
INSTRUCTION	EXPANSION	CODE
HL (e)	LL lsb	066 lsb 056 msb
DE (e)	LH msb	
	LE lsb	046 lsb 036 msb
	LD msb	
BC (e)	LC lsb	036 lsb 026 msb
	LB msb	
MSr _s (e)	LL lsb	066 lsb 37n _s
	LMr _s	
MSr _s *(e)	LL lsb	066 lsb 056 msb 37n _s
	LH msb	
	LMr _s	
MLr _d (e)	LL lsb	066 lsb 3n _d 7
	Lr _d M	
MLr _d *(e)	LL lsb	066 lsb 056 msb 3n _d 7
	LH msb	
	Lr _d M	
SRN (e)	RPT (e)	012 012 ((e) times)
	SRC	
SLN (e)	RPT (e)	002 002 ((e) times)
	SLC	

r_s=source register

r_d=destination register

(e)=expression

lsb=low-order byte of expression value

msb=high-order byte of expression value

n=register reference number

ASSEMBLY ERROR FLAGS

ASSEMBLER 4, DOSASM5, SNAP/1, and SNAP/2	
D	Different definition of labels (pass 1 only, all but first occurrence ignored on pass 2)
I	Instruction mnemonic undefined
E	Expression or label error (unrecognizable character)
U	Undefined label (value of zero assigned)
DOSASM5, SNAP/1, SNAP/2	
F	File error, inclusion limits exceeded or END found in included file
P	Programmer Produced, ERR instruction encountered
SNAP/2	
>	Indicates external reference - not an error condition
O	Overflow on page sensitive PAB

MACROS (SNAP/1 and SNAP/2 only)

Macro Definition (Prototype)

```
MACRO
[label] name [symbol[(default)]] [.symbol[(default)]] ...
    . code
MEND
```

Macro Call

```
[label] name [expression][ expression] ...
```

The replacement of symbols by expressions is position-dependent. If no expression is given for a symbol, the default replaces the symbol; if no default is given, the symbol disappears from the expanded code.

Macro names follow the same syntax rules as labels.

[] above encloses optional fields.

Macro Directives

MIFnn Identical to IFnn directive, for use only in macro definitions
(MIFnn compares strings, rather than numeric values)

MXIF Identical to XIF directive, for use only in macro definitions

ASSEMBLER EXECUTION

ASSEMBLER 4

Run assembler from rear deck (LGO or CTOS tape). Source program as file #0 in front deck. Object file will be written as file #1 on front deck. A series of questions will allow selection of print and display options.

DOSASM5 and SNAP/1

Command line:

```
ASM <sourcefile> [ ,<objectfile> [ ,<entrypt> ] ] [ ;options ]
```

Options:

D - display output

I - list INCLUDED lines

L - print output

G - list lines of generated code

X - print cross-reference

M - list macro expansion lines

F - list IF - skipped lines

(SNAP/1 only)

SNAP/2

Command line:

```
SNAP <sourcefile>[<relocfile>] [;options]
```

Options: same as SNAP/1

SNAP/2 produces relocatable code. The LINK utility must be used to convert the relocatable code to absolute (executable) code.

CHARACTER TRANSMISSION AND TRANSLATION TABLE

DEC	OCTAL	HEX	ASCII	EBDIC	IBM BCD	Honeywell BCD	EBCDIC CARD CODE	BINARY
0	000	00	NUL	NUL		0	12-0-1-8-9	00 000 000
1	001	01	SOH	SOH	1	1	12-1-9	00 000 001
2	002	02	STX	STX	2	2	12-2-9	00 000 010
3	003	03	ETX	ETX	3	3	12-3-9	00 000 011
4	004	04	EOT	PF	4	4	12-4-9	00 000 100
5	005	05	ENQ	HT	5	5	12-5-9	00 000 101
6	006	06	ACK	LC	6	6	12-6-9	00 000 110
7	007	07	BEL	DEL	7	7	12-7-9	00 000 111
8	010	08	BS		8	8	12-8-9	00 001 000
9	011	09	HT	RLF	9	9	12-1-8-9	00 001 001
10	012	0A	LF	SMM	0	1	12-2-8-9	00 001 010
11	013	0B	VT	VT	=	=	12-3-8-9	00 001 011
12	014	0C	FF	FF	/	:	12-4-8-9	00 001 100
13	015	0D	CR	CR	:	:	12-5-8-9	00 001 101
14	016	0E	SO	SO	>	>	12-6-8-9	00 001 110
15	017	0F	SI	SI	@	&	12-7-8-9	00 001 111
16	020	10	DLE	DLE	Space	+	12-11-1-8-9	00 010 000
17	021	11	DC1	DC1	/	A	11-1-9	00 010 001
18	022	12	DC2	DC2	S	B	12-2-9	00 010 010
19	023	13	DC3	TM/DC3	T	C	11-3-9	00 010 011
20	024	14	DC4	RES	U	D	11-4-9	00 010 100
21	025	15	NAK	NL	V	E	11-5-9	00 010 101
22	026	16	SYN	BS	W	F	11-6-9	00 010 110
23	027	17	ETB	IL	X	G	11-7-9	00 010 111
24	030	18	CAN	CAN	Y	H	11-8-9	00 001 000
25	031	19	EM	Em	Z	I	11-1-8-9	00 011 001
26	032	1A	SUB	CC	=	:	11-2-8-9	00 011 010
27	033	1B	ESC	CU1	.	.	11-3-8-9	00 011 011
28	034	1C	FS	IFS	(	)	11-4-8-9	00 011 100
29	035	1D	GS	IGS	-	%	11-5-8-9	00 011 101
30	036	1E	RS	IRS	%	%	11-6-8-9	00 011 110
31	037	1F	US	IUS	//	dirty lozenge ?	11-7-8-9	00 011 111
32	040	20	SPACE	DS	-	-	11-0-1-8-9	00 100 000
33	041	21	!	SOS	J	J	0-1-9	00 100 001
34	042	22	"	FS	K	K	0-2-9	00 100 010
35	043	23	#		L	L	0-3-9	00 100 011
36	044	24	\$	BYP	M	M	0-4-9	00 100 100
37	045	25	%	LF	N	N	0-5-9	00 100 101
38	046	26	&	ETB	O	O	0-6-9	00 100 110
39	047	27	'	ESC	P	P	0-7-9	00 100 111
40	050	28	(		Q	Q	0-8-9	00 101 000
41	051	29	)		R	R	0-1-8-9	00 101 001
42	052	2A	*	SM	!	#	0-2-8-9	00 101 010
43	053	2B	+	CU2	\$	\$	0-3-8-9	00 101 011
44	054	2C	,		*	*	0-4-8-9	00 101 100
45	055	2D	-	ENQ	}	"	0-5-8-9	00 101 101
46	056	2E	.	ACK	;	=	0-6-8-9	00 101 110
47	057	2F	/	BEL	^	!	0-7-8-9	00 101 111

DEC	OCTAL	HEX	ASCII	EBDIC	IBM BCD	Honeywell BCD	EBCDIC CARD CODE	BINARY
48	060	30	0		+	<	12-11-0-1-8-9	00 110 000
49	061	31	1		A	/	1-9	00 110 001
50	062	32	2	SYN	B	S	2-9	00 110 010
51	063	33	3		C	T	3-9	00 110 011
52	064	34	4	PN	D	U	4-9	00 110 100
53	065	35	5	RS	E	V	5-9	00 110 101
54	066	36	6	UC	F	W	6-9	00 110 110
55	067	37	7	EOT	G	X	7-9	00 110 111
56	070	38	8		H	Y	8-9	00 111 000
57	071	39	9		I	Z	1-8-9	00 111 001
58	072	3A	:		?	@	2-8-9	00 111 010
59	073	3B	;	CU3	.	.	3-8-9P	00 111 011
60	074	3C	<	DC4	)	CR	4-8-9	00 111 100
61	075	3D	=	NAK	{	clean lozenge	5-8-9	00 111 101
62	076	3E	>		<	¢	6-8-9	00 111 110
63	077	3F	?	SUB	#		7-8-9	00 111 111
64	100	40	@	SPACE			NO PUNCHES	01 000 000
65	101	41	A				12-0-1-9	01 000 001
66	102	42	B				12-0-2-9	01 000 010
67	103	43	C				12-0-3-9	01 000 011
68	104	44	D				12-0-4-9	01 000 100
69	105	45	E				12-0-5-9	01 000 101
70	106	46	F				12-0-7-9	01 000 110
71	107	47	G				12-0-6-9	01 000 111
72	110	48	H				12-0-8-9	01 001 000
73	111	49	I				12-1-8	01 001 001
74	112	4A	J	¢			12-2-8	01 001 010
75	113	4B	K	.			12-3-8	01 001 011
76	114	4C	L	<			12-4-8	01 001 100
77	115	4D	M	(			12-5-8	01 001 101
78	116	4E	N	+			12-6-8	01 001 110
79	117	4F	O				12-7-8	01 001 111
80	120	50	P	&			12	01 010 000
81	121	51	Q				12-11-1-9	01 010 001
82	122	52	R				12-11-2-9	01 010 010
83	123	53	S				12-11-3-9	01 010 011
84	124	54	T				12-11-4-9	01 010 100
85	125	55	U				12-11-5-9	01 010 101
86	126	56	V				12-11-6-9	01 010 110
87	127	57	W				12-11-7-9	01 010 111
88	130	58	X				12-11-8-9	01 011 000
89	131	59	Y				11-1-8	01 011 001
90	132	5A	Z	!			11-2-8	01 011 010
91	133	5B	[	\$			11-3-8	01 011 011
92	134	5C	]	*			11-4-8	01 011 100
93	135	5D	^	)			11-5-8	01 011 101
94	136	5E	^	;			11-6-8	01 011 110
95	137	5F	_	?			11-7-8	01 011 111
96	140	60	\	-			11	01 100 000
97	141	61	a	/			0-1	01 100 001
98	142	62	b				11-0-2-9	01 100 010
99	143	63	c				11-0-3-9	01 100 011
100	144	64	d				11-0-4-9	01 100 100
101	145	65	e				11-0-5-9	01 100 101
102	146	66	f				11-0-6-9	01 100 110
103	147	67	g				11-0-7-9	01 100 111

DEC	OCTAL	HEX	ASCII	EBDIC	IBM BCD	Honeywell BCD	EBCDIC CARD CODE	BINARY
104	150	68	h				11-0-8-9	01 101 000
105	151	69	i				0-1-8	01 101 001
106	152	6A	j				12-11	01 101 010
107	153	6B	k	,			0-3-8	01 101 011
108	154	6C	l	%			0-4-8	01 101 100
109	155	6D	m	-			0-5-8	01 101 101
110	156	6E	n	>			0-6-8	01 101 110
111	157	6F	o	?			0-7-8	01 101 111
112	160	70	p				12-11-0	01 110 000
113	161	71	q				12-11-0-1-9	01 110 001
114	162	72	r				12-11-0-2-9	01 110 010
115	163	73	s				12-11-0-3-9	01 110 011
116	164	74	t				12-11-0-4-9	01 110 100
117	165	75	u				12-11-0-5-9	01 110 101
118	166	76	v				12-11-0-6-9	01 110 110
119	167	77	w				12-11-0-7-9	01 110 111
120	170	78	x				12-11-0-8-9	01 111 000
121	171	79	y	\			1-8	01 111 001
122	172	7A	z	:			2-8	01 111 010
123	173	7B		#			3-8	01 111 011
124	174	7C		@			4-8	01 111 100
125	175	7D	}	'			5-8	01 111 101
126	176	7E	~	=			6-8	01 111 110
127	177	7F	DEL	"			7-8	01 111 111
128	200	80					12-0-1-8	10 000 000
129	201	81		a			12-0-1	10 000 001
130	202	82		b			12-0-2	10 000 010
131	203	83		c			12-0-3	10 000 011
132	204	84		d			12-0-4	10 000 100
133	205	85		e			12-0-5	10 000 101
134	206	86		f			12-0-6	10 000 110
135	207	87		g			12-0-7	10 000 111
136	210	88		h			12-0-8	10 001 000
137	211	89		i			12-0-9	10 001 001
138	212	8A					12-0-2-8	10 001 010
139	213	8B					12-0-3-8	10 001 011
140	214	8C					12-0-4-8	10 001 100
141	215	8D					12-0-5-8	10 001 101
142	216	8E					12-0-6-8	10 001 110
143	217	8F					12-0-7-8	10 001 111
144	220	90					12-11-1-8	10 010 000
145	221	91		j			12-11-1	10 010 001
146	222	92		k			12-11-2	10 010 010
147	223	93		l			12-11-3	10 010 011
148	224	94		m			12-11-4	10 010 100
149	225	95		n			12-11-5	10 010 101
150	226	96		o			12-11-6	10 010 110
151	227	97		p			12-11-7	10 010 111
152	230	98		q			12-11-8	10 011 000
153	231	99		r			12-11-9	10 011 001
154	232	9A					12-11-2-8	10 011 010
155	233	9B					12-11-3-8	10 011 011
156	234	9C					12-11-4-8	10 011 100
157	235	9D					12-11-5-8	10 011 101
158	236	9E					12-11-6-8	10 011 110
159	237	9F					12-11-7-8	10 011 111

DEC	OCTAL	HEX	ASCII	EBDIC	IBM BCD	Honeywell BCD	EBCDIC CARD CODE	BINARY
160	240	A0					11-0-1-8	10 100 000
161	241	A1		~			11-0-1	10 100 001
162	242	A2		s			11-0-2	10 100 010
163	243	A3		t			11-0-3	10 100 011
164	244	A4		u			11-0-4	10 100 100
165	245	A5		v			11-0-5	10 100 101
166	246	A6		w			11-0-6	10 100 110
167	247	A7		x			11-0-7	10 100 111
168	250	A8		y			11-0-8	10 101 000
169	251	A9		z			11-0-9	10 101 001
170	252	AA					11-0-2-8	10 101 010
171	253	AB		.			11-0-3-8	10 101 011
172	254	AC					11-0-4-8	10 101 100
173	255	AD					11-0-5-8	10 101 101
174	256	AE					11-0-6-8	10 101 110
175	257	AF					11-0-7-8	10 101 111
176	260	B0					12-11-0-1-8	10 110 000
177	261	B1					12-11-0-1	10 110 001
178	262	B2					12-11-0-2	10 110 010
179	263	B3					12-11-0-3	10 110 011
180	264	B4					12-11-0-4	10 110 100
181	265	B5					12-11-0-5	10 110 101
182	266	B6					12-11-0-6	10 110 110
183	267	B7					12-11-0-7	10 110 111
184	270	B8					12-11-0-8	10 111 000
185	271	B9					12-11-0-9	10 111 001
186	272	BA					12-11-0-2-8	10 111 010
187	273	BB					12-11-0-3-8	10 111 011
188	274	BC					12-11-0-4-8	10 111 100
189	275	BD					12-11-0-5-8	10 111 101
190	276	BE					12-11-0-6-8	10 111 110
191	277	BF					12-11-0-7-8	10 111 111
192	300	C0					12-0	11 000 000
193	301	C1		A			12-1	11 000 001
194	302	C2		B			12-2	11 000 010
195	303	C3		C			12-3	11 000 011
196	304	C4		D			12-4	11 000 100
197	305	C5		E			12-5	11 000 101
198	306	C6		F			12-6	11 000 110
199	307	C7		G			12-7	11 000 111
200	310	C8		H			12-8	11 001 000
201	311	C9		I			12-9	11 001 001
202	312	CA					12-0-2-8-9	11 001 010
203	313	CB					12-0-3-8-9	11 001 011
204	314	CC					12-0-4-8-9	11 001 100
205	315	CD					12-0-5-8-9	11 001 101
206	316	CE					12-0-6-8-9	11 001 110
207	317	CF					12-0-7-8-9	11 001 111
208	320	D0					11-0	11 010 000
209	321	D1		J			11-1	11 010 001
210	322	D2		K			11-2	11 010 010
211	323	D3		L			11-3	11 010 011
212	324	D4		M			11-4	11 010 100
213	325	D5		N			11-5	11 010 101
214	326	D6		O			11-6	11 010 110
215	327	D7		P			11-7	11 010 111

DEC	OCTAL	HEX	ASCII	EBDIC	IBM BCD	Honeywell BCD	EBCDIC CARD CODE	BINARY
216	330	D8		Q			11-8	11 011 000
217	331	D9		R			11-9	11 011 001
218	332	DA					12-11-2-8-9	11 011 010
219	333	DB					12-11-3-8-9	11 011 011
220	334	DC					12-11-4-8-9	11 011 100
221	335	DD					12-11-5-8-9	11 011 101
222	336	DE					12-11-6-8-9	11 011 110
223	337	DF					12-11-7-8-9	11 011 111
224	340	E0		0			0-2-8	11 100 000
225	341	E1					11-0-1-9	11 100 001
226	342	E2		S			0-2	11 100 010
227	343	E3		T			0-3	11 100 011
228	344	E4		U			0-4	11 100 100
229	345	E5		V			0-5	11 100 101
230	346	E6		W			0-6	11 100 110
231	347	E7		X			0-7	11 100 111
232	350	E8		Y			0-8	11 101 000
233	351	E9		Z			0-9	11 101 001
234	352	EA					11-0-2-8-9	11 101 010
235	353	EB					11-0-3-8-9	11 101 011
236	354	EC					11-0-4-8-9	11 101 100
237	355	ED					11-0-5-8-9	11 101 101
238	356	EE					11-0-6-8-9	11 101 110
239	357	EF					11-0-7-8-9	11 101 111
240	360	F0		0			0	11 110 000
241	361	F1		1			1	11 110 001
242	362	F2		2			2	11 110 010
243	363	F3		3			3	11 110 011
244	364	F4		4			4	11 110 100
245	365	F5		5			5	11 110 101
246	366	F6		6			6	11 110 110
247	367	F7		7			7	11 110 111
248	370	F8		8			8	11 111 000
249	371	F9		9			9	11 111 001
250	372	FA					12-11-0-2-8-9	11 111 010
251	373	FB					12-11-0-3-8-9	11 111 011
252	374	FC					12-11-0-4-8-9	11 111 100
253	375	FD					12-11-0-5-8-9	11 111 101
254	376	FE					12-11-0-6-8-9	11 111 110
255	377	FF					12-11-0-7-8-9	11 111 111

5500 ROM DEBUG COMMAND SUMMARY:

nnnA	Address the given or current I/O device.
(nnnnnn)B	Set a breakpoint at the given or current address.
(nnnnnn)C	Call the given or current address.
D	Decrement the current address.
E	Continue execution from a breakpoint.
(nnn)F	Fetch next data byte from the given or current device.
(nnn)G	Goto Data mode in the given or current device on 'E'.
I	Increment the current address.
(nnnnnn)J	Jump to the given or current address.
12345K	Set ASCII key in mode.
L	Link to the address pointed to by the current address.
(nnn) nnnM	Modify the contents of the current address.
nnnnnnP	Load the base register.
12345Q	Load the sector table.
R	Switch Alpha/Beta mode flip-flop and display.
nnS	Display the specified stack item.
12345T	Start the memory test.
U	Set User flag, then perform 'E' command.
nnnV	EX COM4 Device must be addressed for I/O commands.
nnnW	EX WRITE Status is displayed after command issue.
nnnX	EX COM1 'nnn' is the output byte.
nnnY	EX COM2
nnnZ	EX COM3
nnnX	Display register and pair (with modify option):
(nnn) nnnA	If input argument exceeds eight bits,
nnnb	the command modifies a register pair.
(nnn) nnnC	The LSB register specifies a pair: e.g. L for HL.
nnnd	The shift key must be depressed during command.
(nnn) nnne	
nnnh	
(nnn) nnnI	
nnnf	Condition flags.
	Adding the number to itself will restore the flags.
nnnnnnENT	Set current address to 'nnnnnn'.
CAN	Cancel entry line.
BKSP	Backspace on entry line.
(nnn) nnn.	Modify and increment.
(nnn) nnn	Modify and increment using last non-null value.
#	Clear all (DEBUG set) breakpoints.

ROM DEBUG DISPLAY FORMAT:

AAAAAA	CURADR (The current address in octal)
X NNN	ASCII, 8-bit octal C(CURADR)
MMMMMM	16-bit (LSB, MSB) address formed at CURADR, CURADR +1
nnnnnnn	Command entry position

MODEL CODE NO. 60311

DATAPoint CORPORATION



The Leader in Dispersed Data Processing™