A.2.2.5 16-Color Graphics Modes (Modes 10 hex, D, E, and 12 hex)

These modes support 16 colors. For all modes, the bit image data is stored in all four memory maps. Each memory map contains the data for one bit plane. The bit planes are C0 through C3 and represent the following colors:

C0= Blue

C1 = Green

C2 = Red

C3 = Intensified

The four bits define each PEL on the screen by acting as an address (pointer) into the internal palette in the Type 2 video.

The display buffer resides at address 0xA0000. The Map Mask register selects any or all of the maps to be updated when the system writes to the display buffer.

A.2.2.6 256-Color Graphics Mode (mode 13 hex)

This mode provides graphics with the capability of displaying 256 colors at one time

The display buffer is sequential, starts at address 0xA0000, and is 64,000 bytes long. The first byte contains the color information for the upper-left PEL. The second byte contains the second PEL, and so on, for 64,000 pels (320 x 200). The bit image data is stored in all four memory maps and comprises four bit planes. The four bit planes are sampled twice to produce eight bit-plane values that address the video DAC.

In this mode, the internal palette of the video subsystem is loaded by BIOS and should not be changed. The first 16 locations in the external palette, which is in the video DAC, contain the colors compatible with the alphanumeric modes. The second 16 locations contain 16 evenly spaced gray shades. The next 216 locations contain values based on a hue-saturation-intensity model tuned to provide a usable, generic color set that covers a wide range of color values.

The following figure shows the color information that is compatible with the colors in other modes:

Table 359. Compatible Color Coding

PEL Bits 76543210	Color Output
00000000	Black
00000001	Blue
00000010	Green
00000011	Cyan
00000100	Red
00000101	Magenta
00000110	Brown
00000111	White
00001000	Gray
00001001	Light Blue
00001010	Light Green

Table 359. Compatible Color Coding *(Continued)*

PEL Bits 76543210	Color Output
00001011	Light Cyan
00001100	Light Red
00001101	Light Magenta
00001110	Yellow
00001111	White (High Intensity)

Each color in the palette can be programmed to one of 256K different colors.

The features of this mode are:

- A maximum of 200 rows with 320 pels
- Double scanned to display as 400 rows
- Memory-mapped graphics
- 256 of 256K colors for each PEL
- One byte per PEL
- 64,000 bytes of video memory.

A.3 Registers

There are six groups of registers in the video subsystem. All video registers are readable except the system data latches and the attribute address flip-flop. The following table lists the register groups, their l/O addresses with the type of access (read or write), and page reference numbers.

The question mark in the address can be a hex B or D depending on the setting of the I/O address bit in the Miscellaneous Output register, described in “General Registers” on page 203.

Software Implementation Note: All registers in the video subsystem are read/write. The value of reserved bits in these registers must be preserved. Read the register first and change only the bits required.

Table 360. VGA Subsystem Register Overview:

Registers	R/W	Port Address
General Registers		
Sequencer Registers		
Address Register	R/W	0x03C4
Data Registers	R/W	0x03C5
CRT Controller Registers		
Address Register	R/W	0x03?4
Data Registers	R/W	0x03?5
Graphics Controller Registers		
Address Register	R/W	0x03CE
Data Registers	R/W	0x03CF

Table 360. VGA Subsystem Register Overview: ***(Continued)***

Registers	R/W	Port Address
Attribute Controller Registers		
Address Register	R/W	0x03C0
Data Registers	W	0x03C0
	R	0x03C1
Video DAC Palette Registers		
Write Address	R/W	0x03C8
Read Address	W	0x03C7
Data	R/W	0x03C9
PEL Mask	R/W	0x03C6

A.3.1 General Registers

Table 361. General Registers:

Register	Read Address	Write Address
Miscellaneous Output Register	0x03CC	0x03C2
Input Status Register 0	0x03C2	-
Input Status Register 1	0x03?A	-
Feature Control Register	0x03CA	0x03?A
Video Subsystem Enable Register	0x03C3	0x03C3

A.3.1.1 Miscellaneous Output Register

Read Address: 0x03CC

Write Address: 0x03C2

7	6	5	4	3	2	1	0
VSP	HSP	—	—	CS		ERAM	IOS

Miscellaneous Output Register

— : Set to 0, undefined on Read
VSP : Vertical Sync Polarity
HSP : Horizontal Sync Polarity
PB : Page Bit for Odd/Even
CS : Clock Select
ERAM : Enable RAM
IOS : I/O Address Select

The register fields are defined as follows:

VSP Determines the polarity of the vertical sync pulse and can be used (with HSP) to control the vertical size of the display by utilizing the autosynchronization feature of VGA displays.
= 0 selects a positive vertical retrace sync pulse.

HSP Determines the polarity of the horizontal sync pulse.
= 0 selects a positive horizontal retrace sync pulse.
Bits 7 (VSP) and 6 (HSP) select the vertical size as follows:

Bits	
7 6	***Vertical Size***
0 0	- Reserved
0 1	- 400 lines
1 0	- 350 lines
1 1	- 480 lines

PB Selects the upper/lower 64K page of memory when the system is in an eve/odd mode (modes 0,1,2,3,7).
= 0 selects the low page
= 1 selects the high page

CS These two bits select the clock source as below: The external clock is driven through the auxiliary video extension. The input clock should be kept between 14.3 MHz and 28.4 MHz.

Bits	
3 2	***Function***
0 0	- Selects 25 MHz clock for 640/320 Horizontal pels
0 1	- Selects 28 MHz clock for 720/360 Horizontal pels
1 0	- Reserved
1 1	- Reserved

ERAM Controls system access to the display buffer.
= 0 disables address decode for the display buffer from the system
= 1 enables address decode for the display buffer from the system

IOS This bit selects the CRT controller addresses. When set to 0, this bit sets the CRT controller addresses to 0x03Bx and the address for the Input Status Register 1 to 0x03BA for compatibility with the monochrome adapter.
When set to 1, this bit sets CRT controller addresses to 0x03Dx and the Input Status Register 1 address to 0x03DA for compatibility with the color/graphics adapter.
The Write addresses to the Feature Control register are affected in the same manner.

A.3.1.2 Input Status Register 0

Read-only Address: 0x03C2

7	6	5	4	3	2	1	0
CI	—	—	SS	—	—	—	—

Input Status Register 0

— : Set to 0, undefined on Read
CI : CRT Interrupt
SS : Switch Sense

The register fields are defined as follows:

CI When set to 1, this bit indicates a vertical retrace interrupt is pending

SS Returns the status of the four sense switches as selected by the CS field of the Miscellaneous Output Register.

A.3.1.3 Input Status Register 1

Read-only Address: 0x03?A

7	6	5	4	3	2	1	0
—	—	—	—	VR	—	—	DE

Input Status Register

— : Set to 0, undefined on Read
VR : Vertical Retrace
DE : Display Enable

The register fields are defined as follows:

VR When set to 1, this bit indicates that the display is in a vertical retrace interval.This bit can be programmed, through the Vertical Retrace End register, to generate an interrupt at the start of the vertical retrace.

DE When set to 1, this bit indicates a horizontal or vertical retrace interval. This bit is the real-time status of the inverted 'display enable' signal. Programs have used this status bit to restrict screen updates to the inactive display intervals in order to reduce screen flicker. The video subsystem is designed to eliminate this software requirement; screen updates may be made at any time without screen degradation.

A.3.1.4 Feature Control Register

Read Address 0x03CA

Write Address: 0x03?A

All bits are reserved.

7	6	5	4	3	2	1	0
Feature Control							

A.3.1.5 Video Subsystem Enable Register

This register (0x03C3) is reserved.

7	6	5	4	3	2	1	0
Video Subsystem Enable							

A.3.2 Sequencer Registers

The Address register is at address **0x03C4** and the data registers are at address **0x03C5**. All registers within the sequencer are read/write.

Table 362. Sequencer Registers

Register	Index (hex)
Sequencer Address	—
Reset	00
Clocking Mode	01
Map Mask	02
Character Map Select	03
Memory Mode	04

A.3.2.1 Sequencer Address Register

The Address register is at address **0x03C4**. This register is loaded with an index value that points to the desired sequencer data register.

7	6	5	4	3	2	1	0
—	—	—	—	—	SA		

Sequencer Address Register

— : Set to 0, undefined on Read
SA : Sequencer Address

The register field is defined as follows:

SA These bits contain the index value that points to the data register to be accessed.

A.3.2.2 Reset Register

Address 0x03C4; Data 0x03C5; Index 0x00.

7	6	5	4	3	2	1	0
—	—	—	—	—	—	SR	ASR

Reset Register

— : Set to 0, undefined on Read
SR : Synchronous Reset
ASR : Asynchronous Reset

The register fields are defined as follows:

SR When set to 0, this bit commands the sequencer to synchronously clear and halt. Bits 1 and 0 must be 1 to allow the sequencer to operate. To prevent the loss of data, bit 1 must be set to 0 during the active display interval before changing the clock selection. The clock is changed through the Clocking Mode register or the Miscellaneous Output register.

ASR When set to 0, this bit commands the sequencer to asynchronously clear and halt. Resetting the sequencer with this bit can cause loss of video data.

A.3.2.3 Clocking Mode Register

Address 0x03C4; Data 0x03C5; Index 0x01.

7	6	5	4	3	2	1	0
—	—	SO	SH4	DC	SL	1	D89

Clocking Mode Register

— : Set to 0, undefined on Read
1 : Set to 1, undefined on Read
SO : Screen Off
SH4 : Shift 4
DC : Dot Clock
SL : Shift Load
D89 : 8/9 Dot Clocks

The register fields are defined as follows:

SO When set to 1, this bit turns off the display and assigns maximum memory bandwidth to the system. Although the display is blanked, the synchronization pulses are maintained. This bit can be used for rapid full-screen updates.

SH4 When the Shift 4 field and the Shift Load Field are set to 0, the video serializers are loaded every character clock. When the Shift 4 field is set to 1, the video serializers are loaded every forth character clock, which is useful when 32 bits are fetched per cycle and chained together in the shift registers.

DC When set to 0, this bit selects the normal dot clocks derived from the sequencer master clock input. When this bit is set to 1, the master clock will be divided by 2 to generate the dot clock. All other timings are affected because they are derived from the dot clock. The dot clock divided by 2 is used for 320 and 360 horizontal PEL modes.

SL When this bit and bit 4 are set to 0, the video serializers are loaded every character clock. When this bit is set to 1, the video serializers are loaded every other character clock, which is useful when 16 bits are fetched per cycle and chained together in the shift registers. The Type 2 video behaves as if this bit is set to 0; therefore, programs should set it to 0.

D89 When set to 0, this bit directs the sequencer to generate character clocks 9 dots wide; when set to 1, it directs the sequencer to generate character clocks 8 dots wide. The 9-dot mode is for alphanumeric modes 0+, 1+, 2+, 3+, 7, and 7 + only; the 9th dot equals the 8th dot for ASCII codes 0xC0 through 0xDF. All other modes must use 8 dots per character clock. See the line graphics character bit in the “Attribute Mode Control Register” on page 228

A.3.2.4 Map Mask Register

Address 0x03C4; Data 0x03C5; Index 0x02.

7	6	5	4	3	2	1	0
—	—	—	—	M3E	M2E	M1E	M0E

Map Mask Register, Index 0x02

— : Set to 0, undefined on Read
M3E : Map 3 Enable
M2E : Map 2 Enable
M1E : Map 1 Enable
M0E : Map 0 Enable
D89 : 8/9 Dot Clocks

The register fields are defined as follows:

M*E — When set to 1, these bits enable system access to the corresponding map. If all maps are enabled, the system can write its 8-bit value to all four maps in a single memory cycle. This substantially reduces the system overhead during display updates in graphics modes.
Data scrolling operations can be enhanced by enabling all maps and writing the display buffer address with the data stored in the system data latches. This is a Read-Modify-Write operation.
When odd/even modes are selected, maps 0 and 1 and maps 2 and 3 should have the same map mask value.
When chain 4 mode is selected, all maps should be enabled.

A.3.2.5 Character Map Select Register

Address 0x03C4; Data 0x03C5; Index 0x03.

In alphanumeric modes, bit 3 of the attribute byte normally defines the foreground intensity. This bit can be redefined as a switch between character sets allowing 512 displayable characters. To enable this feature:

1. Set the extended memory bit in the Memory Mode register (0x04) to 1.

2. Select different values for character map A and character map B.

This function is supported by BIOS and is a function call within the character generator routines.

7	6	5	4	3	2	1	0
—	—	MAH	MBH	MAL		MBL	

Character Map Select Register, Index 0x03

— : Set to 0, undefined on Read
MAH : Character Map A Select (MSB)
MBH : Character Map B Select (MSB)
MAL : Character Map A Select (LSB)
MBL : Character Map B Select (LSB)

The register fields are defined as follows:

MAH This bit is the most-significant bit for selecting the location of character map A.

MBH This bit is the most-significant bit for selecting the location of character map B.

MAL These bits and bit 5 select the location of character map A. Map A is the area of map 2 containing the character font table used to generate characters when attribute bit 3 is set to 1. The selection is shown in Table 363 on page 209.

MBL These bits and bit 4 select the location of character map B. Map B is the area of map 2 containing the character font table used to generate characters when attribute bit 3 is set to 0. The selection is shown in Table 364 on page 209.

Bits 532	Map Selected	Table Location
000	0	1st 8KB of display memory plane 2
001	1	3rd 8KB
010	2	5th 8KB
011	3	7th 8KB
100	4	2nd 8KB
101	5	4th 8KB
110	6	6th 8KB
111	7	8th 8KB

Table 363. Character Map Select A

Bits 410	Map Selected	Table Location
000	0	1st 8KB of display memory plane 2
001	1	3rd 8KB
010	2	5th 8KB
011	3	7th 8KB
100	4	2nd 8KB
101	5	4th 8KB
110	6	6th 8KB
111	7	8th 8KB

Table 364. Character Map Select B

A.3.2.6 Memory Mode Register

Address 0x03C4; Data 0x03C5; Index 0x04.

7	6	5	4	3	2	1	0
—	—	—	—	CH4	OE	EM	—

Memory Mode Register, Index 0x04

— : Set to 0, undefined on Read
CH4 : Chain 4
OE : Odd/Even
EM : Extended Memory

The register fields are defined as follows:

CH4 This bit controls the map selected during system read operations. When set to 0, this bit enables system addresses to sequentially access data within a bit map by using the Map Mask register. When set to 1, this bit causes the two low-order bits to select the map accessed as shown below.

Address Bits	
A0	***A1***
0	0
0	1
1	0
1	1

Map Selected
0
1
2
3

OE When this bit is set to 0, even system addresses access maps 0 and 2, while odd system addresses access maps 1 and 3. When this bit is set to 1, system addresses sequentially access data within a bit map, and the maps are accessed according to the value in the Map Mask register (index 0x02).

EM When set to 1, this bit enables the video memory from 64KB to 256KB. This bit must be set to 1 to enable the character map selection described for the previous register.

A.3.3 CRT Controller Registers

A data register is accessed by writing its index to the Address register at address 0x03D4 or 0x03B4, and then writing the data to the access port at address 0x03D5 or 0x03B5. The l/O address used depends on the setting of the l/O address select bit (bit 0) in the Miscellaneous Output register, which is described in “General Registers” on page 203. The following figure shows the variable part of the address as a question mark.

Software Implementation Note: When modifying a register, the setting of reserved bits must be preserved. Read the register first and change only the bits required.

Table 365. CRT Controller Registers

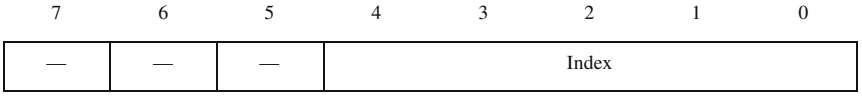
Register Name	Address	Index
Address	0x03?4	—
Horizontal Total	0x03?5	0x0
Horizontal Display-Enable End	0x03?5	0x01
Start Horizontal Blanking	0x03?5	0x02
End Horizontal Blanking	0x03?5	0x03
Start Horizontal Retrace Pulse	0x03?5	0x04
Index values not listed are reserved.		

Table 365. CRT Controller Registers *(Continued)*

Register Name	Address	Index
End Horizontal Retrace	0x03?5	0x05
Vertical Total	0x03?5	0x06
Overflow	0x03?5	0x07
Preset Row Scan	0x03?5	0x08
Maximum Scan Line	0x03?5	0x09
Cursor Start	0x03?5	0x0A
Cursor End	0x03?5	0x0B
Start Address High	0x03?5	0x0C
Start Address Low	0x03?5	0x0D
Cursor Location High	0x03?5	0x0E
Cursor Location Low	0x03?5	0x0F
Vertical Retrace Start	0x03?5	0x10
Vertical Retrace End	0x03?5	0x11
Vertical Display-Enable End	0x03?5	0x12
Offset	0x03?5	0x13
Underline Location	0x03?5	0x14
Start Vertical Blanking	0x03?5	0x15
End Vertical Blanking	0x03?5	0x16
CRT Mode Control	0x03?5	0x17
Line Compare	0x03?5	0x18
Index values not listed are reserved.		

A.3.3.1 Address Register

This register is at address 0x03?4, and is loaded with an index value that points to the data registers within the CRT controller.



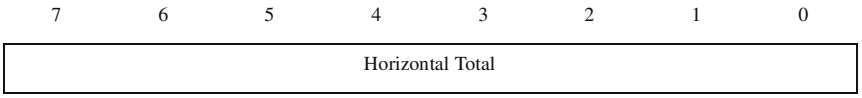
Address Register

— Set to 0, undefined on Read

Index : These bits are the index that points to the data register accessed through address 0x03D5 or 0x03B5.

A.3.3.2 Horizontal Total Register

Address 0x03?4; Data 0x03?5; Index 0x00

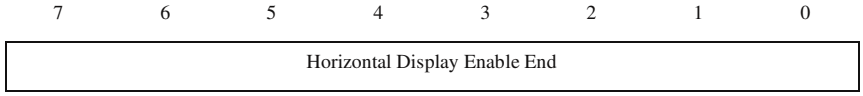


This register the total number of characters in the horizontal scan interval including the retrace time. The value directly controls the period of the 'horizontal retrace' signal. A horizontal character counter in the CRT controller counts the character clock inputs; comparators are used to compare the register value with the character's horizontal width to provide horizontal timings. All horizontal and vertical timings are based on this register.

The value contained in this register is the total number of characters minus 5.

A.3.3.3 Horizontal Display-Enable End Register

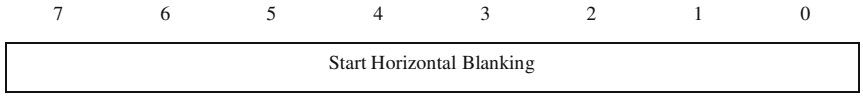
Address 0x03?4; Data 0x03?5; Index 0x01



The value in this register defines the length of the 'horizontal display-enable' signal, and determines the number of character positions per horizontal line. The value contained in this register is the total number of displayed characters minus 1.

A.3.3.4 Start Horizontal Blanking Register

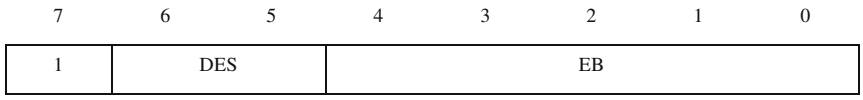
Address 0x03?4; Data 0x03?5; Index 0x02



The value in this register is the horizontal character count where the 'horizontal blanking' signal goes active.

A.3.3.5 End Horizontal Blanking Register

Address 0x03?4; Data 0x03?5; Index 0x03



End Horizontal Blanking Register, Index 0x03

1 : Set to 1, undefined on Read
DES : Display Enable Skew Control
EB : End Blanking

The register fields are defined as follows:

DES — These two bits determine the amount of skew of the 'display enable' signal. This skew control is needed to provide sufficient time for the CRT controller to read a character and attribute code from the video buffer, to gain access to the character generator, and go through the Horizontal PEL Panning register in the attribute controller. Each access requires the 'display enable' signal to be skewed one character clock so that the video output is synchronized with the horizontal and vertical retrace signals. The skew values are shown below.

DES Field

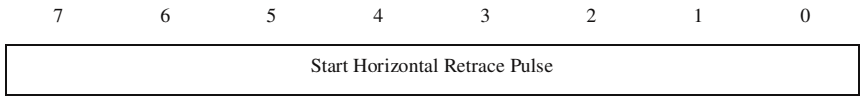
6	*5*	*Amount of Skew*
0	0	No character clock skew
0	1	One character clock skew
1	0	Two character clock skew
1	1	Three character clock skew

Note: Character skew is not adjustable on the Type 2 video and the bits are ignored; however, programs should set these bits for the appropriate skew to maintain compatibility.

EB — These bits are the five low-order bits of a 6-bit value that is compared with the value in the Start Horizontal Blanking register to determine when the 'horizontal blanking' signal will go inactive. The most-significant bit is bit 7 in the End Horizontal Retrace register (index 0x05).
To program these bits for a signal width of W, the following algorithm is used: the width W, in character clock units, is added to the value from the Start Horizontal Blanking register. The six low-order bits of the result are the 6-bit value programmed.

A.3.3.6 Start Horizontal Retrace Pulse Register

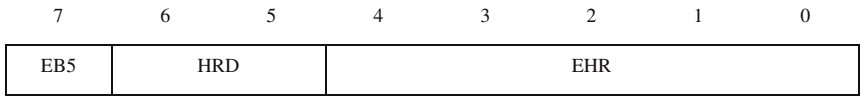
Address 0x03?4; Data 0x03?5; Index 0x04



These bits are used to center the screen horizontally by specifying the character position where the 'horizontal retrace' signal goes active.

A.3.3.7 End Horizontal Retrace Register

Address 0x03?4; Data 0x03?5; Index 0x05



End Horizontal Retrace Register, Index 0x05

EB5 : End Horizontal Blanking, Bit 5
HRD : Horizontal Retrace Delay
EHR : End Horizontal Retrace

The register fields are defined as follows:

EB5 — This bit is the most significant bit (bit 5) of the end horizontal blanking value in the End Horizontal Blanking register (index 0x03).

The register fields are defined as follows: ***(Continued)***

HRD — These bits control the skew of the 'horizontal retrace' signal. The value of these bits is the amount of skew provided (from 0 to 3 character clock units). For certain modes, the 'horizontal retrace' signal takes up the entire blanking interval. Some internal timings are generated by the falling edge of the 'horizontal retrace' signal. To ensure that the signals are latched properly, the 'retrace' signal is started before the end of the 'display enable' signal and then skewed several character clock times to provide the Proper screen centering.

EHR — These bits are compared with the Start Horizontal Retrace register to give a horizontal character count where the 'horizontal retrace' signal goes inactive.
To program these bits with a signal width of W, the following algorithm is used: the width W, in character clock units, is added to the value in the Start Retrace register. The five low-order bits of the result are the 5-bit value programmed.

A.3.3.8 Vertical Total Register

Address 0x03?4; Data 0x03?5; Index 0x06

7	6	5	4	3	2	1	0
Vertical Total							

These are the eight low-order bits of a 10-bit vertical total. The value for the vertical total is the number of horizontal raster scans on the display, including vertical retrace, minus 2. This value determines the period of the 'vertical retrace' signal.

Bits 8 and 9 are in the Overflow register (index 0x07).

A.3.3.9 Overflow Register

Address 0x03?4; Data 0x03?5; Index 0x07

7	6	5	4	3	2	1	0
VRS9	VDE9	VT9	LC8	VBS8	VRS8	VDE8	VT8

Overflow Register, Index 0x07

VRS9 : Vertical Retrace Start, Bit 9
VDE9 : Vertical Display Enable End, Bit 9
VT9 : Vertical Total, Bit 9
LC8 : Line Compare, Bit 8
VBS8 : Vertical Blanking Start, Bit 8
VRS8 : Vertical Retrace Start, Bit 8
VDE8 : Vertical Display Enable End, Bit 8
VT8 : Vertical Total, Bit 8

The register fields are defined as follows:

VRS9 — Bit 9 of the Vertical Retrace Start register (index 0x10).
VDE9 — Bit 9 of the Vertical Display Enable End register (index 0x12).
VT9 — Bit 9 of the Vertical Total register (index 0x06).
LC8 — Bit 8 of the Line Compare register (index 0x18).
VBS8 — Bit 8 of the Start Vertical Blanking register (index 0x15).
VRS8 — Bit 8 of the Vertical Retrace Start register (index 0x10).
VDE8 — Bit 8 of the Vertical Display Enable End register (index 0x12).

The register fields are defined as follows: ***(Continued)***

VRS9 Bit 9 of the Vertical Retrace Start register (index 0x10).

VT8 Bit 8 of the Vertical Total register (index 0x06).

A.3.3.10 Preset Row Scan Register

Address 0x03?4; Data 0x03?5; Index 0x08

7	6	5	4	3	2	1	0
—	BP						SRS

Preset Row Scan Register, Index 0x08

— : Set to 0, Undefined on Read
BP : Byte Panning
SRS : Starting Row Scan Count

The register fields are defined as follows:

BP These two bits control byte panning in multiple shift modes. These bits are used in pel-panning operations. and should normally be set to 0.

SRS These bits specify the row scan count for the row starting after a vertical retrace. The row scan counter is incremented every horizontal retrace time until the maximum row scan occurs. When the maximum row scan is reached, the row scan counter is cleared (not preset).

Note: The CRT controller latches the start address at the start of the vertical retrace. These register values should be loaded during the active display time.

A.3.3.11 Maximum Scan Line Register

Address 0x03?4; Data 0x03?5; Index 0x09

7	6	5	4	3	2	1	0
DSC	LC9	VBS9					MSL

Maximum Scan Line Register, Index 0x09

DSC : 200 to 400 Line Conversion
LC9 : Line Compare, Bit 9
VBS9 : Start Vertical Blanking, Bit 9
MSL : Maximum Scan Line

The register fields are defined as follows:

DSC When this bit is set to 1, 200-scan-line video data is converted to 400-scan-line output. To do this, the clock in the row scan counter is divided by 2, which allows the 200-line modes to be displayed as 400 lines on the display (this is called double scanning; each line is displayed twice). When this bit is set to 0, the clock to the row scan counter is equal to the horizontal scan rate.

LC9 Bit 9 of the Line Compare register (index 0x18).

VBS9 Bit 9 of the Start Vertical Blanking register (index 0x15).

The register fields are defined as follows: ***(Continued)***

DSC When this bit is set to 1, 200-scan-line video data is converted to 400-scan-line output. To do this, the clock in the row scan counter is divided by 2, which allows the 200-line modes to be displayed as 400 lines on the display (this is called double scanning; each line is displayed twice). When this bit is set to 0, the clock to the row scan counter is equal to the horizontal scan rate.

MSL These bits specify the number of scan lines per character row. The value of these bits is the maximum row scan number minus 1.

A.3.3.12 Cursor Start Register

Address 0x03?4; Data 0x03?5; Index 0x0A

7	6	5	4	3	2	1	0
—	—	CO	RSCB				

Cursor Start Register, Index 0x0A

— : Set to 0, Undefined on Read
CO : Cursor Off
RSCB : Row Scan Cursor Begins

The register fields are defined as follows:

CO When set to 1, this bit disables the cursor.

RSCB These bits specify the row within the character box where the cursor begins. The value of these bits is the first line of the cursor minus 1. When this value is greater than that in the Cursor End register, no cursor is displayed.

A.3.3.13 Cursor End Register

Address 0x03?4; Data 0x03?5; Index 0x0B

7	6	5	4	3	2	1	0
—	CSK		RSCE				

Cursor End Register, Index 0x0B

— : Set to 0, Undefined on Read
CSK : Cursor Skew Control
RSCE : Row Scan Cursor Ends

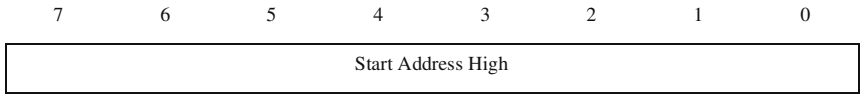
The register fields are defined as follows:

CSK These bits control the skew of the cursor. The skew value delays the cursor by the selected number of character clocks from 0 to 3. For example, a skew of 1 moves the cursor right one position on the screen.

RSCE These bits specify the row within the character box where the cursor ends. If this value is less that the cursor start value, no cursor is displayed.

A.3.3.14 Start Address High Register

Address 0x03?4; Data 0x03?5; Index 0x0C

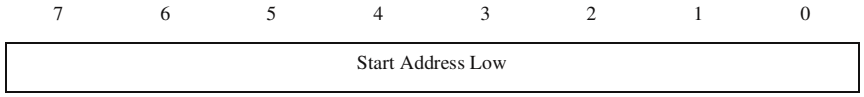


These are the eight high-order bits of a 16-bit value that specifies the starting address for the regenerative buffer. The start address points to the first address after the vertical retrace on each screen refresh.

Software Implementation Note: The CRT controller latches the start address at the start of the vertical retrace. These register values should be loaded during the active display time.

A.3.3.15 Start Address Low Register

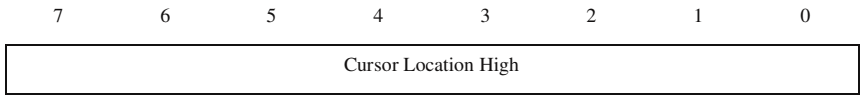
Address 0x03?4; Data 0x03?5; Index 0x0D



These are the eight low-order bits of the starting address for the regenerative buffer.

A.3.3.16 Cursor Location High Register

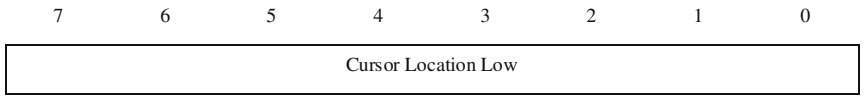
Address 0x03?4; Data 0x03?5; Index 0x0E



These are the eight high-order bits of the 16-bit cursor

A.3.3.17 Cursor Location Low Register

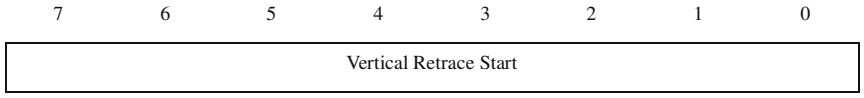
Address 0x03?4; Data 0x03?5; Index 0x0F



These are the eight low-order bits of the cursor location.

A.3.3.18 Vertical Retrace Start Register

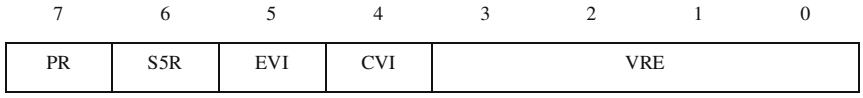
Address 0x03?4; Data 0x03?5; Index 0x10



These are the eight low-order bits of the 9-bit start position for the 'vertical retrace' pulse; it is programmed in horizontal scan lines. Bit 8 is in the Overflow register (index 0x07).

A.3.3.19 Vertical Retrace End Register

Address 0x03?4; Data 0x03?5; Index 0x11



Vertical Retrace End Register, Index 0x11

PR : Protect Registers 0-7
S5R : Select 5 Refresh Cycles
EVI : Enable Vertical Interrupt
CVI : Clear Vertical Interrupt
VRE : Vertical Retrace End

The register fields are defined as follows:

PR — When set to 1, this bit disables write access to the CRT controller registers at index 00 through 07. The line compare bit in the Overflow register (index 0x07) is not protected.

S5R — When set to 1, this bit generates five memory refresh cycles per horizontal line. When set to 0, this bit selects three refresh cycles. Selecting five refresh cycles allows use of the VGA chip with 15.75 kHz displays. This bit should be set to 0 for supported operations.

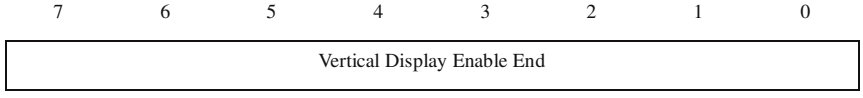
EVI — When set to 0, this bit enables a vertical retrace interrupt. The vertical retrace interrupt is IRQ2. This interrupt level can be shared; therefore, to determine whether the video generated the interrupt, check the CRT interrupt bit in Input Status Register 0.

CVI — When set to 0, this bit clears a vertical retrace interrupt. At the end of the active vertical display time, a flip-flop is set to indicate an interrupt. An interrupt handler resets this flip-flop by first setting this bit to 0, then resetting it to 1.

VRE — The Vertical Retrace Start register is compared with these four bits to determine where the 'vertical retrace' signal goes inactive. It is programmed in units of horizontal scan lines. To program these bits with a signal width of W, the following algorithm is used: the width W, in horizontal scan units, is added to the value in the Start Vertical Retrace register. The four low-order bits of the result are the 4-bit value programmed.

A.3.3.20 Vertical Display Enable End Register

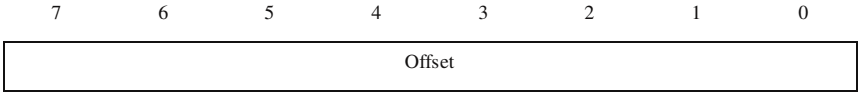
Address 0x03?4; Data 0x03?5; Index 0x12



These are the eight low-order bits of a 10-bit value that defines the vertical-display-enable end position. The two high-order bits are contained in the Overflow register (index 0x07). The 10-bit value is equal to the total number of scan lines minus 1.

A.3.3.21 Offset Register

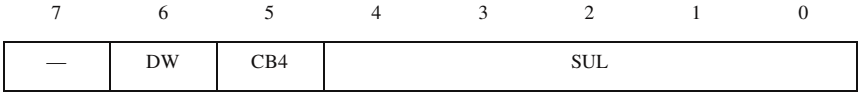
Address 0x03?4; Data 0x03?5; Index 0x013



These bits specify the logical line width of the screen. The starting memory address for the next character row is larger than the current character row by 2 or 4 times the value of these bits. Depending on the method of clocking the CRT controller, this address is either a word or doubleword address.

A.3.3.22 Underline Location Register

Address 0x03?4; Data 0x03?5; Index 0x14



Underline Location Register, Index 0x14

— : Set to 0, Undefined on Read
DW : Doubleword Mode
CB4 : Count by 4
SUL : Start Underline

The register fields are defined as follows:

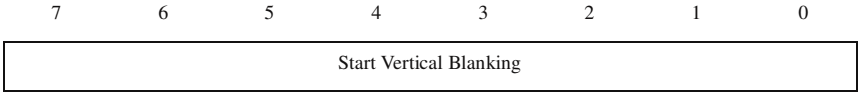
DW — When this bit is set to 1, memory addresses are doubleword addresses. See the description of the word/byte mode bit (bit 6) in the “CRT Mode Control Register” on page 220.

CB4 — When this bit is set to 1, the memory-address counter is clocked with the character clock divided by 4, which is used when doubleword addresses are used.

SUL — These bits specify the horizontal scan line of a character row on which an underline occurs. The value programmed is the scan line desired minus 1.

A.3.3.23 Start Vertical Blanking Register

Address 0x03?4; Data 0x03?5; Index 0x15



These are the eight low-order bits of a 10-bit value that specifies the starting location for the 'vertical blanking' signal. Bit 8 is in the Overflow register (index 0x07) and bit 9 is in the Maximum Scan Line register (index 0x09). The 10-bit value is the horizontal scan line count where the 'vertical blanking' signal becomes active minus 1.

A.3.3.24 End Vertical Blanking Register

Address 0x03?4; Data 0x03?5; Index 0x16

7	6	5	4	3	2	1	0
End Vertical Blanking							

This register specifies the horizontal scan count where the 'vertical blanking' signal becomes inactive. The register is Programmed in units of the horizontal scan line.

To program these bits with a 'vertical blanking' signal of width W, the following algorithm is used: the width W, in horizontal scan line units, is added to the value in the Start Vertical Blanking register minus 1. The eight low-order bits of the result are the 8-bit value programmed.

A.3.3.25 CRT Mode Control Register

Address 0x03?4; Data 0x03?5; Index 0x17

7	6	5	4	3	2	1	0
RST	WB	ADW	—	CB2	HRS	SRC	CMS0

CRT Mode Control Register, Index 0x17

— : Set to 0, Undefined on Read
RST : Hardware Reset
WB : Word/Byte Mode
ADW : Address Wrap
CB2 : Count by Two
HRS : Horizontal Retrace Select
SRC : Select Row Scan Counter
CMS0 : Compatibility Mode Support

The register fields are defined as follows:

RST — When set to 0, this bit disables the horizontal and vertical retrace signals and forces them to an inactive level. When set to 1, this bit enables the horizontal and vertical retrace signals. This bit does not reset any other registers or signal outputs.

WB — When this bit is set to 0, the word mode is selected. The word mode shifts the memory-address counter bits to the left by one bit; the most-significant bit of the counter appears on the least-significant bit of the memory address outputs.
The doubleword bit in the Underline Location register (0x14) also controls the addressing. When the doubleword bit is 0, the word/byte bit selects the mode. When the doubleword bit is set to 1, the addressing is shifted by two bits.
When set to 1, bit 6 selects the byte address mode.

ADW — This bit selects the memory-address bit, bit MA 13 or MA 15, that appears on the output pin MA 0, in the word address mode. If the VGA is not in the word address mode, bit 0 from the address counter appears on the output pin, MA 0.
When set to 1, this bit selects MA 15. In odd/even mode, this bit should be set to 1 because 256KB of video memory is installed on the system board. (Bit MA 13 is selected in applications where only 64KB is present. This function maintains compatibility with the IBM Color/Graphics Monitor Adapter.)

The register fields are defined as follows: ***(Continued)***

CB2 When this bit is set to 0, the address counter uses the character clock. When this bit is set to 1, the address counter uses the character clock input divided by 2. This bit is used to create either a byte or word refresh address for the display buffer.

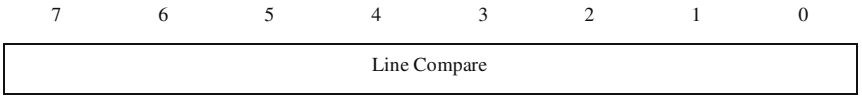
HRS This bit selects the clock that controls the vertical timing counter. The clocking is either the horizontal retrace clock or horizontal retrace clock divided by 2. When this bit is set to 1. the horizontal retrace clock is divided by 2.
Dividing the clock effectively doubles the vertical resolution of the CRT controller. The vertical counter has a maximum resolution of 1024 scan lines because the vertical total value is 10-bits wide. If the vertical counter is clocked with the horizontal retrace divided by 2, the vertical resolution is doubled to 2048 scan lines.

SRC This bit selects the source of bit 14 of the output multiplexer. When this bit is set to 0, bit 1 of the row scan counter is the source. When this bit is set to 1, the bit 14 of the address counter is the source.

CMS0 This bit selects the source of bit 13 of the output multiplexer. When this bit is set to 0, bit 0 of the row scan counter is the source, and when this bit is set to 1, bit 13 of the address counter is the source.
The CRT controller used on the IBM Color/Graphics Adapter was capable of using 128 horizontal scan-line addresses. For the VGA to obtain 640-by-200 graphics resolution, the CRT controller is programmed for 100 horizontal scan lines with two scan-line addresses per character row. Row scan address bit 0 becomes the most-significant address bit to the display buffer. Successive scan lines of the display image are displaced in 8KB of memory. This bit allows compatibility with the graphics modes of earlier adapters.

A.3.3.26 Line Compare Register

Address 0x03?4; Data 0x03?5; Index 0x18



This register contains the eight low-order bits of the 10-bit compare target. When the vertical counter reaches the target value, the internal start address of the line counter is cleared. This creates a split screen where the lower screen is immune to scrolling. Bit 8 is in the Overflow register (index 0x07), and bit 9 is in the Maximum Scan Line register (index 0x09).

A.3.4 Graphics Controller Registers

The Address register for the graphics controller is at address 0x03CE. The data registers are at address 0x03CF. All registers are read/write.

Table 366. Graphics Controller Registers

Register Name	Address	Index
Address	0x03CE	—
Set/Reset	0x03CF	0x0
Enable Set/Reset	0x03CF	0x01
Color Compare	0x03CF	0x02
Data Rotate	0x03CF	0x03
Read Map Select	0x03CF	0x04
Graphics Mode	0x03CF	0x05
Miscellaneous	0x03CF	0x06

Table 366. Graphics Controller Registers *(Continued)*

Register Name	Address	Index
Color Don't Care	0x03CF	0x07
Bit Mask	0x03CF	0x08

A.3.4.1 Address Register

The Address register is at address **0x03CE**

7	6	5	4	3	2	1	0
—	—	—	—	Index			

This register is loaded with the index value that points to the desired data register within the graphics controller.

A.3.4.2 Set/Reset Register

Address 0x03CE; Data 0x03CF; Index 0x00

7	6	5	4	3	2	1	0
—	—	—	—	SR3	SR2	SR1	SR0

Set/Reset Register, Index 0x00

— : Set to 0, Undefined on Read
SR3 : Set/Reset Map 3
SR2 : Set/Reset Map 2
SR1 : Set/Reset Map 1
SR0 : Set/Reset Map 0

The register fields are defined as follows:

SR* When write mode 0 is selected, the system writes the value of each set/reset bit to its respective memory map. For each write operation, the set/reset bit, if enabled, is written to all eight bits within that map. Set/reset operation can be enabled on a map-by-map basis through the Enable Set/Reset register.

A.3.4.3 Enable Set/Reset Register

Address 0x03CE; Data 0x03CF; Index 0x01

7	6	5	4	3	2	1	0
—	—	—	—	ESR3	ESR2	ESR1	ESR0

Enable Set/Reset Register, Index 0x01

— : Set to 0, Undefined on Read
ESR3 : Enable Set/Reset Map 3
ESR2 : Enable Set/Reset Map 2
ESR1 : Enable Set/Reset Map 1

Enable Set/Reset Register, Index 0x01

ESR0 : Enable Set/Reset Map 0

The register fields are defined as follows:

ESR* These bits enable the set/reset function used when write mode 0 is selected in the Graphics Mode register (index 0x05). When the bit is set to 1, the respective memory map receives the value specified in the Set/Reset register. When Set/Reset is not enabled for a map, that map receives the value sent by the system.

A.3.4.4 Color Compare Register

Address 0x03CE; Data 0x03CF; Index 0x02

7	6	5	4	3	2	1	0
—	—	—	—	CC3	CC2	CC1	CC0

Color Compare Register, Index 0x02

— : Set to 0, Undefined on Read
CC3 : Color Compare Map 3
CC2 : Color Compare Map 2
CC1 : Color Compare Map 1
CC0 : Color Compare Map 0

The register fields are defined as follows:

CC* These bits are the 4-bit color value to be compared when the read mode bit in the Graphics Mode register is set to 1. When the system does a memory read, the data returned from the memory cycle will be a 1 in each bit position where the four maps equal the Color Compare register. If the read mode bit is 0, the data is returned without comparison.
All bits of the corresponding map's byte are compared with the color compare bit. Each of the eight bit positions in the selected byte are compared across the four maps, and a 1 is returned in each position where the bits of all four maps equal their respective color compare values.

A.3.4.5 Data Rotate Register

Address 0x03CE; Data 0x03CF; Index 0x03

7	6	5	4	3	2	1	0
—	—	—	FS		RC		

Data Rotate Register, Index 0x03

— : Set to 0, Undefined on Read
FS : Function Select
RC : Rotate Count

The register fields are defined as follows:

FS Data written to the video buffer can be operated on logically by data already in the system latches. The Function Select field determines whether and how this is done.
Data can be any of the choices selected by the write mode bits except system latches, which cannot be modified. If rotated data is selected also, the rotate is performed before the logical operation. The logical operations selected are shown in the following table.

FS Field

4 3	***Function***
0 0	Data Unmodified
0 1	Data ANDed with Latched Data
1 0	Data ORed with Latched Data
1 1	Data XORed with Latched Data

RC In write mode 0, these bits select the number of positions the system data is rotated to the right during a system Memory Write operation. To write data that is not rotated in mode 0, all bits are set to 0.

A.3.4.6 Read Map Select Register

Address 0x03CE; Data 0x03CF; Index 0x04

7	6	5	4	3	2	1	0
—	—	—	—	—	—	MS	

Read Map Select Register, Index 0x04

— : Set to 0, Undefined on Read
MS : Map Select

The register field is defined as follows:

MS These bits select the memory map for system read operations. This register has no effect on the color compare read mode. In odd/even modes, the value can be a binary 00 or 01 to select the chained maps 0, 1 and the value can be a binary 10 or 11 to select the chained maps 2, 3.

A.3.4.7 Graphics Mode Register

Address 0x03CE; Data 0x03CF; Index 0x05

7	6	5	4	3	2	1	0
—	C256	SR	OE	RM	—	WM	

Graphics Mode Register, Index 0x05

— : Set to 0, Undefined on Read
C256 : 256 - Color Mode
SR : Shift Register Mode
OE : Odd/Even
RM : Read Mode
WM : Write Mode

The register fields are defined as follows:

C256 When set to 0, this bit allows bit 5 to control the loading of the shift registers. When set to 1, this bit causes the shift registers to be loaded in a manner that supports the 256-color mode.

SR When set to 1, this bit directs the shift registers in the graphics controller to format the serial data stream with even-numbered bits from both maps on even-numbered maps, and odd-numbered bits from both maps on the odd-numbered maps. This bit is used for modes 4 and 5.

OE When set to 1, this bit selects the odd/even addressing mode used by the IBM Color/Graphics Monitor Adapter. Normally, the value here follows the value of Memory Mode register bit 2 in the sequencer.

RM When this bit is set to 1, the system reads the results of the comparison of the four memory maps and the Color Compare register.
When this bit is set to 0, the system reads data from the memory map selected by the Read Map Select register, or by the two low-order bits of the memory address (this selection depends on the chain-4 bit in the Memory Mode register of the sequencer).

WM The write mode selected and its operation are defined below. The logic operation specified by the function select bits is performed on system data for modes 0, 2, and 3.

WM Field	*Mode Description*
00	Each memory map is written with the system data rotated by the count in the Data Rotate register. If the set/reset function is enabled for a specific map, that map receives the 8-bit value contained in the Set/Reset register.
01	Each memory map is written with the contents of the system latches. These latches are loaded by a system Read operation.
10	Memory map n (0 through 3) is filled with eight bits of the value of data bit n.
11	Each memory map is written with the 8-bit value contained in the Set/Reset register for that map (the Enable Set/Reset register has no effect). Rotated system data is ANDed with the Bit Mask register to form an 8-bit value that performs the same function as the Bit Mask register in write modes 0 and 2 (see also “Bit Mask Register” on page 226).

A.3.4.8 Miscellaneous Register

Address 0x03CE; Data 0x03CF; Index 0x06

7	6	5	4	3	2	1	0
—	—	—	—	MM		OE	GM

Miscellaneous Register, Index 0x06

— : Set to 0, Undefined on Read
MM : Memory Map
OE : Odd/Even
GM : Graphics Mode

The register fields are defined as follows:

MM — These bits control the mapping of the regenerative buffer into the system address space. The bit functions are defined below.

MM Field	
3 2	***Addressing Assignment***
0 0	A0000 for 128KB
0 1	A0000 for 64KB
1 0	B0000 for 32 KB
1 1	B8000 for 32 KB

OE — When set to 1, this bit directs the system address bit, A0, to be replaced by a higher-order bit. The odd map is then selected when A0 is 1, and the even map when A0 is 0.

GM — This bit controls alphanumeric mode addressing. When set to 1, this bit selects graphics modes, which also disables the character generator latches.

A.3.4.9 Color Don't Care Register

Address 0x03CE; Data 0x03CF; Index 0x07

7	6	5	4	3	2	1	0
—	—	—	—	M3	M2	M1	M0

Color Don't Care Register, Index 0x07

— : Set to 0, Undefined on Read
M3 : Compare Map 3
M2 : Compare Map 2
M1 : Compare Map 1
M0 : Compare Map 0

The register fields are defined as follows:

M* — These bits select whether a map is going to participate in the color compare cycle. When the bit is set to 1, the bits in that map are compared.

A.3.4.10 Bit Mask Register

Address 0x03CE; Data 0x03CF; Index 0x08

7	6	5	4	3	2	1	0
Bit Mask							

When the bit is set to 1, the corresponding bit position in each map can be changed. When the bit set to 0, the bit position in the map is masked to prevent change, provided that the location being written was the last location read by the system microprocessor.

The bit mask applies to write modes 0 and 2. To preserve bits using the bit mask, data must be latched internally by reading the location. When data is written to preserve the bits, the most current data in the latches is written in those positions. The bit mask applies to all maps simultaneously.

A.3.5 Attribute Controller Registers

Each register for the attribute controller has two addresses. Address 0x03C0 is the write address and 0x03C1 is the read address. The individual data registers are selected by writing their index to the Address register (0x03C0).

Table 367. Attribute Controller Registers:

Register Name	Write Address	Read Address	Index
Address	0x03C0	0x03C0	—
Internal Palette	0x03C0	0x03C1	0x0-0x0F
Attribute Mode Control	0x03C0	0x03C1	0x10
Overscan Color	0x03C0	0x03C1	0x11
Color Plane Enable	0x03C0	0x03C1	0x12
Horizontal PEL Panning	0x03C0	0x03C1	0x13
Color Select	0x03C0	0x03C1	0x14

A.3.5.1 Address Register

This read/write register is at address **0x03C0**.

The attribute controller registers do not have an input bit to control selection of the address and data registers. An internal address flip-flop controls this selection. Reading Input Status Register 1 clears the flip-flop and selects the Address register.

After the Address register has been loaded with the index, the next Write operation to 03C0 loads the data register. The flip-flop toggles for each Write operation to address 0x03C0. It does not toggle for Read operations to 03C0 or 03C1.

7	6	5	4	3	2	1	0
—	—	IPAS	Index				

Address Register

— : Set to 0, Undefined on Read
IPAS : Internal Palette Address Source

The register fields are defined as follows:

IPAS — This bit is set to 0 to load color values to the registers in the internal palette. It is set to 1 for normal operation of the attribute controller.
Note: Do not access the internal palette while this bit is set to 1. While this bit is 1, the Type 1 video subsystem disables accesses to the palette; however, the Type 2 does not, and the actual color value addressed cannot be ensured.

Index — These bits contain the index to the data registers in the attribute controller.

A.3.5.2 Internal Palette Registers 0 through F

Write 0x03C0; Read 0x03C1; Index 0x00-0x0F

7	6	5	4	3	2	1	0
—	—	P5	P4	P3	P2	P1	P0

Internal Palette Registers

— : Set to 0, Undefined on Read
P5 to P0 : Palette Data

The register fields are defined as follows:

P5-P0 These 6-bit registers allow a dynamic mapping between the text attribute or graphic color input value and the display color on the CRT screen. When set to 1, this bit selects the appropriate color. The Internal Palette registers should be modified only during the vertical retrace interval to avoid problems with the displayed image. These internal palette values are sent off-chip to the video DAC, where they serve as addresses into the DAC registers.

Bit	*Color*
0	Blue
1	Green
2	Red
3	Secondary Blue
4	Secondary Green
5	Secondary Red

Software Implementation Note: These registers can be accessed only when bit 5 in the Address register is set to 0. When the bit is 1, writes are "don't care" and reads return undefined data.

A.3.5.3 Attribute Mode Control Register

Write 0x03C0; Read 0x03C1; Index 0x10.

7	6	5	4	3	2	1	0
PS	PW	PP	—	EB	ELG	ME	G

Attribute Mode Control Register

— : Set to 0, Undefined on Read
PS : P5, P4 Select
PW : PEL Width
PP : PEL Panning Compatibility
EB : Enable Blink/Select Background Intensity
ELG : Enable Line Graphics Character Code
ME : Mono Emulation
G : Graphics/Alphanumeric Mode

The register fields are defined as follows:

PS This bit selects the source for the P5 and P4 video bits that act as inputs to the video DAC. When this bit is set to 0, P5 and P4 are the outputs of the Internal Palette registers. When this bit is set to 1, P5 and P4 are bits 1 and 0 of the Color Select register.

PW When this bit is set to 1, the video data is sampled so that eight bits are available to select a color in the 256-color mode (0x13). This bit is set to 0 in all other modes.

PP When this bit is set to 1, a successful line-compare in the CRT controller forces the output of the PEL Panning register to 0 until a vertical synchronization occurs, at which time the output returns to its programmed value. This bit allows a selected portion of a screen to be panned.
When this bit is set to 0, line compare has no effect on the output of the PEL Panning register.

The register fields are defined as follows: ***(Continued)***

EB When this bit is set to 0, the most-significant bit of the attribute selects the background intensity (allows 16 colors for background). When set to 1, this bit enables blinking.

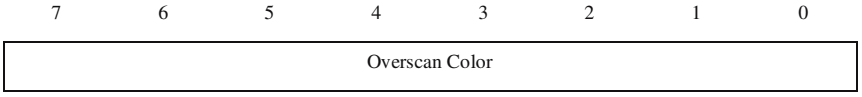
ELG When this bit is set to 0, the ninth dot will be the same as the background. When set to 1, this bit enables the special line-graphics character codes for the monochrome emulation mode. This emulation mode forces the ninth dot of a line graphic character to be identical to the eighth dot of the character. The line-graphics character codes for the monochrome emulation mode are 0xC0 through 0xDF. For character fonts that do not utilize these line-graphics character codes, bit 2 should be set to 0 to prevent unwanted video information from displaying on the CRT screen. BIOS will set this bit, the correct dot clock, and other registers when the 9-dot alphanumeric mode is selected.

ME When this bit is set to 1, monochrome emulation mode is selected. When this bit is set to 0, color emulation mode is selected.

G When set to 1, this bit selects the graphics mode of operation.

A.3.5.4 Overscan Color Register

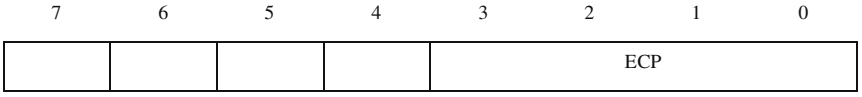
Write 0x03C0; Read 0x03C1; Index 0x11.



These bits select the border color used in the 80-column alphanumeric modes and in the graphics modes other than modes 4, 5, and D. (Selects a color from one of the DAC registers.)

A.3.5.5 Color Plane Enable Register

Write 0x03C0; Read 0x03C1; Index 0x12.



Color Plane Enable Register

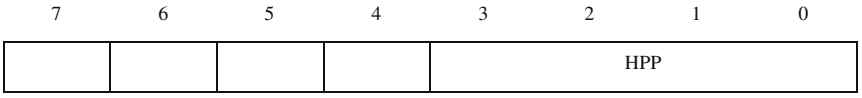
— : Set to 0, Undefined on Read
ECP : Enable Color Plane

The register field is defined as follows:

ECP Setting a bit to 1, enables the corresponding display-memory color plane.

A.3.5.6 Horizontal PEL Panning Register

Write 0x03C0; Read 0x03C1; Index 0x13.



Attribute Mode Control Register

— : Set to 0, Undefined on Read
HPP : Horizontal PEL Panning

The register field is defined as follows:

HPP These bits select the number of pels that the video data is shifted to the left. PEL panning is available in both alphanumeric and graphics modes. The following table shows the number of bits shifted for each mode.

Table 368. Image Shifting

Register Value	Number of Pels Shifted to the Left		
	Mode Hex 13	A/N Modes*	All Other Modes
0	0	1	0
1	—	2	1
2	1	3	2
3	—	4	3
4	2	5	4
5	—	6	5
6	3	7	6
7	—	8	7
8	—	0	—

* Only mode 7 and the A/N modes with 400 scan lines.

A.3.5.7 Color Select Register

Write 0x03C0; Read 0x03C1; Index 0x14.

7	6	5	4	3	2	1	0
—	—	—	—	SC7	SC6	SC5	SC4

Color Select Register

— : Set to 0, Undefined on Read
SC7 : S_color 7
SC6 : S_color 6
SC5 : S_color 5
SC4 : S_color 4

The register fields are defined as follows:

SC7, SC6 In modes other than mode 13 hex, these are the two most-significant bits of the 8-bit digital color value to the video DAC. In mode 13 hex, the 8-bit attribute is the digital color value to the video DAC. These bits are used to rapidly switch between sets of colors in the video DAC.

SC5, SC4 These bits can be used in place of the P4 and P5 bits from the Internal Palette registers to form the 8-bit digital color value to the video DAC. Selecting these bits is done in the Attribute Mode Control register (index 0x10). These bits are used to rapidly switch between colors sets within the video DAC.

A.3.6 Video Digital to Analog Converter

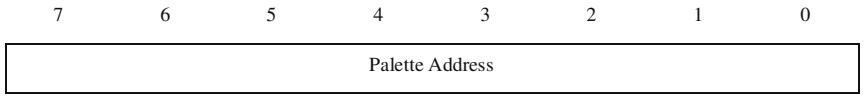
The video digital-to-analog converter (DAC) integrates the function of a color palette and three internal DACs for driving an analog display. The DAC has 256 registers containing 18 bits each to allow the display of up to 256 colors from a possible 256K colors.

Table 369. DAC Registers

Register Name	Read/ Write	Address
Palette Address (Write Mode)	R/W	0x03C8
Palette Address (Read Mode)	W	0x03C7
DAC State	R	0x03C7
Palette Data	R/W	0x03C9
Pel Mask	R	0x03C6

A.3.6.1 Palette Address Write Mode Register

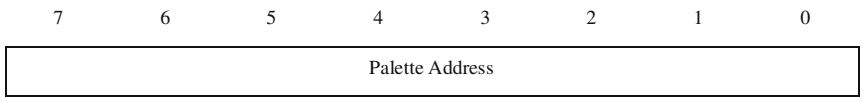
Read/Write Address 0x03C8



This register contains the 8-bit address used to access the 256 color registers during a write operation. Color data from the Palette Data Register is loaded into the palette registers in three separate output cycles per write operation. At the end of the third output to the Palette Data Register, the Palette Address Register will automatically increment.

A.3.6.2 Palette Address Read Mode Register

Write Address 0x03C7



This register contains the 8-bit address used to access the 256 color registers during a read operation. Color data from the palette registers is loaded into the Palette Data Register in three separate output cycles per write operation. At the end of the third output to the Palette Data Register, the Palette Address Register will automatically increment.

A.3.6.3 DAC State Register

Read Address 0x03C7

7	6	5	4	3	2	1	0
—	—	—	—	—	—	STA	

This is a read-only register that returns the last active operation in bits 1 and 0. If the last operation was a read operation, both bits are set to 1. If the last operation was a write, both bits are set to 0.

A.3.6.4 Palette Data Register

Read/Write Address 0x03C9

7	6	5	4	3	2	1	0
Palette Data							

The Palette Data Register is really 18 bits wide to correspond to the three 6-bit RGB representations in the palette registers. Since the system interface is 8 bits, three read/write operations are needed to access this register for each palette register.

A.3.6.5 Pel Mask Register

Read Address 0x03C6

7	6	5	4	3	2	1	0
Pel Mask							

This read only register is initialized by the mode set software and should not be further modified.

A.3.6.6 Device Operation

The palette address (P7 - P0) and the blanking input are sampled on the rising edge of the PEL clock. After three more PEL clock cycles, the video reflects the state of these inputs.

During normal operation the palette address is used as a pointer to one of the 256 data registers in the palette. The value in each data register is converted to an analog signal for each of the three outputs (red, green, blue). The blanking input is used to force the video output to 0 volts. The blanking operation is independent of the palette operation.

Each data register is 18 bits wide: 6 bits each for red, green, and blue. The data registers are accessible through the system interface.

A.3.6.7 Video DAC to System Interface

The Palette Address register holds an 8-bit value that is used to address a location within the video DAC. The Palette Address register responds to two addresses; the address depends on the type of palette access, Read or Write. Once the address is loaded, successive accesses to the data register automatically increment the address register.

For palette Write operations, the address for the Palette Address register is 0x03C8. A write cycle consists of writing three successive bytes to the Data register at address 0x03C9. The six least-significant bits of each byte are concatenated to form the 18-bit palette data. The order is red value first, then green, then blue.

For palette Read operations, the address for the Palette Address register is 0x03C7 (in the read mode, the Palette Address register is write only). A read cycle consists of reading three successive bytes from the Data register at address 0x03C9. The six least-significant bits of each byte contain the corresponding color value. The order is red value first, then green, then blue.

If the Palette Address register is written to during a Read or Write cycle, a new cycle is initialized and the unfinished cycle is terminated. The effects of writing to the Data register during a Read cycle or reading from the Data register during a Write cycle are undefined and can change the palette contents.

The DAC State register is a read-only register at address 0x03C7. Bits 1 and 0 return the last active operation to the DAC. If the last operation was a Read operation, both bits are set to 1. If the last operation was a Write, both bits are set to 0.

Reading the Read Palette Address register at 0x03C8 or the DAC State register at 0x03C7 does not interfere with read or write cycles.

A.3.6.8 Programming Considerations

As explained in “Video DAC to System Interface” on page 232, the effects of writing to the Data register during a read cycle or reading from the Data register during a write cycle are undefined and can change the palette contents. Therefore, the following sequence must be followed to ensure the integrity of the color palette during accesses to it:

1. 1. Disable interrupts.
2. 2. Write the address to PEL Address register.
3. 3. Write or read three bytes of data.
4. 4. Go to Step 2, repeat for the desired number of locations.
5. 5. Enable interrupts.

All accesses to the DAC registers are byte-wide l/O operations.

To prevent "snow" on the screen, an application reading data from or writing data to the DAC registers should ensure that the blank input to the DAC is asserted. This can be accomplished either by restricting data transfers to retrace intervals (use Input Status register 1 to determine when retrace is occurring) or by using the Screen Off bit located in the Clocking Mode register in the sequencer.

Do not write to the PEL Mask register (0x03C6). Palette information can be changed as a result. This register is correctly initialized to 0xFF during a mode set.

A.4 VGA Programming Considerations

The following are some programming considerations for the VGA:

1. The following rules must be followed to guarantee the critical timings necessary to ensure the proper operation of the CRT controller:
    a. The value in the Horizontal Total register must be at least 0x19.

b. The minimum positive pulse width of the 'horizontal synchronization' signal must be four character clock units.

c. The End Horizontal Retrace register must be programmed such that the 'horizontal synchronization' signal goes to 0 at least one character clock time before the 'horizontal display enable' signal goes active.

d. The End Vertical Blanking register must be set to a minimum of one horizontal scan line greater than the line-compare value.

2. When PEL panning compatibility is enabled in the Attribute Mode Control register, a successful line compare in the CRT controller forces the output of the Horizontal PEL Panning register to 0's until a vertical synchronization occurs. When the vertical synchronization occurs, the output returns to the programmed value. This allows the portion of the screen indicated by the Line Compare register to be operated on by the Horizontal PEL Panning register.

3. A write to the Character Map Select register becomes valid on the next whole character line. This will prevent deformed character images when changing character generators in the middle of a character scan line.

4. For mode 13 hex, the attribute controller is configured so that the 8-bit attribute in video memory becomes the 8-bit address (P0-P7) into the video DAC. The user should not modify the contents of the Internal Palette registers

5. The following is the sequence for accessing the attribute data registers:

   • 1. Disable interrupts.

   • 2. Reset the flip-flop for the Attribute Address register.

   • 3. Write the index.

   • 4. Access the data register.

   • 5. Enable interrupts.

6. • The Color Select register in the attribute controller section allows the programmer to rapidly switch color sets in the video DAC. Bit 7 of the Attribute Mode Control register controls the number of bits in the Color Select register used to address the color information in the video DAC (either two or four bits are used). By changing the value in the Color Select register, an application can switch color sets in graphics and alphanumeric modes (mode 13 hex does not use this feature).

   For multiple color sets, the user must load the color values.

7. An application that saves the video state must store the four bytes of information contained in the system microprocessor latches in the graphics controller subsection. These latches are loaded with 32 bits from video memory (8 bits per map) each time the system reads from video memory. The application needs to:

   a. Use write mode 1 to write the values in the latches to a location in video memory that is not part of the display buffer, such as the last location in the address range.

   b. Save the values of the latches by reading them back from video memory.

   If memory addressing is in the chain-4 or odd/even mode, reconfigure the memory as four sequential maps prior to performing the sequence above.

8. The Horizontal PEL Panning register allows programs to control the starting position of the display area on the screen. The display area can be shifted to the left up to eight PEL positions. In single-byte shift modes, to

pan to the PEL position above 8, the CRT controller start address is incremented and the PEL Panning register is reset to 0.

In multiple shift modes, the byte-panning bits (in the Reset Row Scan register) are used as extensions to the Horizontal PEL Panning register. This allows panning across the width of the video output. For example, in the 32-bit shift mode, the byte pan and pel-panning bits provide panning up to 31 bits. To pan from position 31 to 32, the CRT controller start address is incremented and the panning bits, both PEL and byte, are reset to 0.

Further panning can be accomplished by changing the start-address value in the CRT controller registers, Start Address High and Start Address Low. The sequence is:

1. Use the Horizontal PEL Panning register to shift the maximum number of bits to the left.
2. Increment the start address.
3. Set the Horizontal PEL Panning register so that no bits are shifted.

The screen will now be shifted one PEL to the left of the position it was in at the end of Step 1. Step 1 through Step 3 are repeated as often as necessary.

4. When using a split-screen application that scrolls a second screen on top of the first screen and operating in a mode with 200 scan lines, the Line Compare register (CRT Controller register 0x19) must contain an even value. This is a requirement of the double scanning logic in the CRT controller.
5. If the value in the Cursor Start register (CRT Controller register 0x0A) is greater than that in the Cursor End register (CRT Controller register 0x0B), the cursor is not displayed.
6. In 8-dot character modes, the underline attribute produces a solid line across adjacent characters. In 9-dot character modes, the underline across adjacent characters is dashed. In 9-dot modes with the line-graphics characters (C0 - DF character codes), the underline is solid.

A.4.1 Programming the Registers

Each of the video components has an address register and a number of data registers. The data registers have addresses common to all registers for that component. The individual registers are selected by a pointer (index) in its Address register. To write to a data register, the Address register is loaded with the index of the desired data register, then the data register is loaded by writing to the common l/O address.

The general registers do not share a common address; they each have their own I/O address.

See “Video DAC to System Interface” on page 232 for details on programming the video DAC.

For compatibility with the IBM Enhanced Graphics Adapter (EGA), the internal video subsystem palette is programmed the same as the EGA. Using BIOS to program the palette will produce a color compatible to that produced by the EGA. mode 13 hex (256 colors) is programmed so that the first 16 locations in the DAC produce compatible colors.

When BIOS is used to load the color palette for a color mode and a monochrome display is attached, the color palette is changed. The colors are summed to produce shades of gray that allow color applications to produce a readable screen.

Modifying the following bits must be done while the sequencer is held in a synchronous reset through its Reset register:

- Bits 3 and 0 of the Clocking Mode register.
- Bits 3 and 2 of the Miscellaneous Output register.

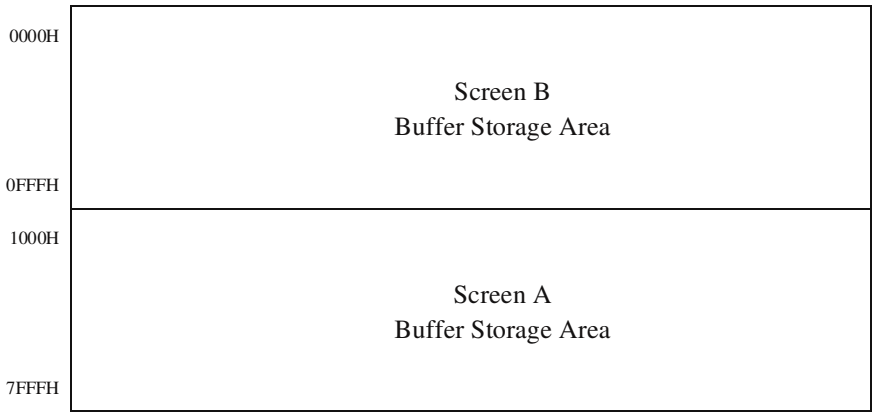
A.4.2 RAM Loadable Character Generator

The character generator is RAM loadable and can support characters up to 32 scan lines high. Three character fonts are stored in BIOS, and one is automatically loaded when an alphanumeric mode is selected. The Character Map Select register can be programmed to redefine the function of bit 3 of the attribute byte to be a character-font switch. This allows the user to select between any two character sets residing in map 2, and effectively gives the user access to 512 characters instead of 256. Character fonts can be loaded off-line, and up to eight fonts can be loaded at any one time. The character generator is in map 2 and must be protected using the map mask function.

A.4.3 Creating a Split Screen

The VGA hardware supports a split screen. The top portion of the screen is designated as screen A, and the bottom portion is designated as screen B.

The following figure shows the screen mapping for a system containing a 32KB alphanumeric storage buffer, such as the VGA. Information displayed on screen A is defined by the Start Address. High and Low registers of the CRT controller. Information displayed on screen B always begins at video address 0x0000.



The Line Compare register of the CRT controller performs the split screen function. The CRT controller has an internal horizontal scan line counter and logic that compares the counter value to the value in the Line Compare register and clears the memory address generator when a comparison occurs. The linear address generator then sequentially addresses the display buffer starting at location 0. Each subsequent row address is determined by the 16-bit addition of the start-of-line latch and the Offset register.

Screen B can be smoothly scrolled onto the display by updating the Line Compare register in synchronization with the 'vertical retrace' signal. Screen B information is not affected by scrolling operations that use the Start Address registers to scroll through the screen A information.

When pel-panning compatibility is enabled (Attribute Mode Control register), a successful line comparison forces the output of the Horizontal PEL Panning register to 0's until vertical synchronization occurs. This feature allows the information on screen B to remain unaffected by pel-panning operations on screen A.

Requirements Summary

B

This appendix lists all the requirements of this architecture. The requirements are collected under the chapter heading. This appendix may be used for a quick reference list of all requirements for building a standard platform. The requirements list follows:

Chapter 1 General Requirements and Information

1–1. The interfaces define in this document must not be accessed by RTAS or marked as *used-by-rtas* with two exception as follows:

a. Any RTAS service that does not return to the operating system may access any interface.

b. The interfaces in Chapter 13, “Graphics,” on page 161 may be accessed by the RTAS *display-character* function, if the graphics interface in the Open Firmware device tree node does not contain the *used-by-rtas* property, but is the device indicated by the *rtas-display-device* property.

1–2. If the implementation of the devices in this document have additional registers which are not part of the programming model given in this document, then these registers must not be required for correct operation of the device by the operating system and they must all be declared in the Open Firmware device tree to prevent other devices from being assigned to their addresses.

1–3. Any controller not contained on standard adaptor cards for their I/O bus must have the *builtin* property.

Chapter 2 ISA DMA Controller

2–1. An ISA DMA Controller must be compatible with the Intel 82378ZB System I/O (SIO), except as noted in the following requirements in this chapter.

2–2. This device (ISA DMA Controller) must be addressed only with ISA I/O addresses.

2–3. This device must be represented by the Open Firmware properties “name=dma-controller, device_type=dma-controller, compatible= chrp,dma”.

2–4. This device must have the Open Firmware properties defined in CHRP ISA DMA Controller Device Binding to: IEEE 1275-1994 Standard for Boot (Initialization, Configuration) Firmware.

2–5. The registers shown in Table 2 on page 5, Register Map for ISA DMA Device, must all be present in the device defined in this chapter.

2–6. All the reg property fields specified in Table 2 on page 5 must comply with the format given in the ISA Bus Binding to: IEEE 1275-1994 Standard for Boot (Initialization, Configuration) Firmware.

2–7. This device must be represented as an ISA device and a child of an ISA bridge node in the Open Firmware device tree.

2–8. For ISA DMA, the DACK# signal must be active low.

2–9. For ISA DMA, the DREQ signal must be active high.

2–10. For ISA DMA, software must assume the terminal count (TC) is an output from the DMA controller. This means some ISA devices may not work if they depend on TC being an input to the DMA controller.

2–11. For ISA DMA, a hardware design must not produce an error if 0b0 is written to bit 6 of DCEM1 or DCEM2.

2–12. For operations involving ISA DMA slave devices, software must assume the hardware only allows addressing modes as follows:

a. Channels 0-3 are 8-bit count by bytes addressing mode. The DMA address, count, and page registers must be programmed accordingly.

b. Channels 5-7 are 16-bit count by word (2 bytes) addressing mode. The DMA address, count, and page registers must be programmed accordingly.

c. 16-bit count by bytes addressing mode is not supported.

2–13. For ISA DMA, a hardware design must not produce an error if 0b00 is written to bits [3:2] of DCEM1, or if 0b01 is written to bits [3:2] of DCEM2.

2–14. Software must assume the DMA Write All Mask Bits Registers (WAM1, WAM2) are write-only.

2–15. During Scatter/Gather operations, the High Page Registers (described in Section 2.4.13, “Current High Page Register 0 (CHIP0),” on page 13 through Section 2.4.14, “Current High Page Register 1-7 (CHIP1-CHIP7),” on page 14) and Low Page Registers (described in Section 2.4.9, “Current Low Page Register 0 (CLOP0),” on page 12 through Section 2.4.10, “Current Low Page Register 1-7 (CLOP1-CLOP7),” on page 13) must increment appropriately as part of the full 32 bit transfer address.

2–16. Scatter/Gather Descriptor: The Memory Buffer Starting Physical Address field in the SGD must be programmed as follows:

a. For 8-bit count by byte operations, this field is programmed with the byte address of the buffer to be transferred. The address may be even or odd.

b. For 16-bit count by word operations, this field is programmed with the even byte address of the buffer to be transferred. Thus, bit 0 in this field will be 0.

2–17. IRQ 13 must be reserved for the ISA DMA device.

2–18. The Scatter/Gather Descriptor Table Pointer Registers must be programmed with a single 32-bit write.

2–19. The ISA DMA Controller must not preclude an ISA bus master from using any of the seven available DMA channels, regardless of addressing mode.

2–20. The ISA DMA hardware must provide at least 24-bit addressing to system memory for ISA bus masters.

Chapter 3 Floppy Disk Controller

3–1. A Floppy Disk Controller that is compliant with the interfaces as defined in this section must be represented in the Open Firmware device tree for the auto-eject version of the device as identified by the following properties; name=fdc, device_type=fdc, and compatible=chrp,fdc. For the non-auto-eject version of the device, it is represented by the following properties; name=fdc, device_type=fdc, and compatible=pnpPNP,700.

3–2. The firmware must configure the Floppy Disk Controller in the PS/2™ mode, with enhanced auto-eject or without auto-eject.

3–3. This device must be addressed purely with ISA I/O addresses which can be aliased or non-aliased.

3–4. DMA transfers for the floppy disk controller must be 8-bit DMA slave and must support at least the ISA Compatible Timing mode.

3–5. Interrupts generated by the floppy disk controller must be high true edge sensitive.

3–6. The floppy disk controller must support 3.5 inch media with an unformatted capacity of 2 megabytes and a formatted capacity of 1.44 megabytes.

3–7. The floppy disk controller must support the 3.5 inch media with an unformatted capacity of 4 megabytes and a formatted capacity of 2.88 megabytes for purposes of floppy disk port tape backup units.

3–8. The three sets of registers shown in Table 52 on page 33 must be present in the Open Firmware device tree for the auto-eject version of the device as identified by the following properties; name=fdc, device_type=fdc, and compatible=chrp,fdc.

3–9. The first two sets of registers shown in Table 52 on page 33 must be present in the Open Firmware device tree for the non-auto-eject version of the device as identified by the following properties; name=fdc, device_type=fdc, and compatible=pnpPNP,700.

3–10. This device must be represented as an ISA device and a child of an ISA bridge node in the Open Firmware device tree.

3–11. A floppy disk controller must be reported in the Open Firmware device tree as specified in the *CHRP ISA Floppy Device Binding* [16].

3–12. For the device described herein, all the registers defined in Section 3.4.1, “Status Register A (SRA),” on page 33 through Section 3.4.10, “Autoeject Register (AEJ),” on page 41 must be implemented as described herein.

3–13. For the device described herein, the commands described in Section 3.5.1.1, “Configure Command,” on page 43 through Section 3.5.1.22, “Write Deleted Data Command,” on page 55 must be implemented as specified.

3–14. For the device described herein, the Command Status Registers described in Section 3.5.2.1, “Status Register 0 (ST0),” on page 56 through Section 3.5.2.4, “Status Register 3 (ST3),” on

page 58 must be implemented as specified.

3–15. The disk drive must be configured in a manner such that the Disk in Place signal, on the drive, is functionally a state indicator of the presence of media. The signal must assert an active level when a disk is present in the drive and an inactive level when there is no disk in the drive.

Chapter 4 Legacy Interrupt Controller

4–1. The Legacy Interrupt Controller must be represented by the Open Firmware properties “name=interrupt-controller, device_type=interrupt-controller, compatible=chrp,iic”.

4–2. The Legacy Interrupt Controller registers specified herein must be implemented precisely as defined, including the following characteristics and bit definitions:

a. Table 129, “Initialization Command Word 1 Register (ICW1) Characteristics,” on page 68

b. Table 130, “Initialization Command Word 1 Register (ICW1),” on page 68

c. Table 131, “Initialization Command Word 2 Register (ICW2) Characteristics,” on page 68

d. Table 132, “Initialization Command Word 2 Register (ICW2),” on page 68

e. Table 133, “Initialization Command Word 3 Register (ICW3) Characteristics,” on page 69.

f. Table 134, “Initialization Command Word 3 Register (ICW3),” on page 69

g. Table 135, “Initialization Command Word 4 Register (ICW4) Characteristics,” on page 69

h. Table 136, “Initialization Command Word 4 Register (ICW4),” on page 70

i. Table 137, “Operation Command Word 1 Register (OCW1) Characteristics,” on page 70

j. Table 138, “Operation Command Word 1 Register (OCW1),” on page 70

k. Table 140, “Operation Command Word 2 Register (OCW2) Characteristics,” on page 71

l. Table 141, “Operation Command Word 2 Register (OCW2),” on page 71

m. Table 144, “Operation Command Word 3Register (OCW3) Characteristics,” on page 72

n. Table 145, “Operation Command Word 3 Register (OCW3),” on page 73

o. Table 149, “Interrupt Request Register (IRR) Characteristics,” on page 74

p. Table 143, “Interrupt Level Select Encodings,” on page 72

q. Table 151, “In-Service Register (ISR) Characteristics,” on page 75

r. Table 143, “Interrupt Level Select Encodings,” on page 72

s. Table 153, “Edge/Level Interrupt Control Register 1 (ELI1) Characteristics,” on page 76

t. Table 154, “Edge/Level Interrupt Control 1 Register (ELI1),” on page 76

u. Table 155, “Edge/Level Interrupt Control 2 Register (ELI2) Characteristics,” on page 76

v. Table 156, “Edge/Level Interrupt Control 2 Register (ELI2),” on page 76

4–3. This device must be addressed purely with ISA I/O addresses.

4–4. The registers shown in Table 125 on page 63 must all be present in the device defined in this chapter. The Open Firmware “reg” property information given in Table 125 on page 63 must be provided in the Open Firmware device tree for this device.

4–5. This device must be represented as an ISA device and a child of the ISA bridge node in the Open Firmware device tree.

4–6. This device must be represented in the OF device tree as specified in the *CHRP ISA Interrupt Controller Device Binding to: IEEE 1275-1994 Standard for Boot (Initialization, Configuration) Firmware*.

4–7. All the reg property fields specified in table Table 125 on page 63 must comply with the format given in the ISA (or PCI) bus binding to IEEE Std 1275-1994, as modified by the CHRP binding to IEEE Std 1275-1994.

4–8. The following events must occur during the OF initialization sequence:

a. The level sense circuit must be set to the Edge-sensitive mode (following the initialization procedure, interrupts are generated by a low-to-high transition high level on the interrupt request input.)

b. IRQ7 must be assigned priority 7.

c. PIC2 (slave) mode address must be set to 7.

d. The Special Mask mode must be cleared and the controller must be set to read the Interrupt Request register.

e. All Legacy Interrupt Controller registers must be initialized to the values indicated in the “Value in Register When OF Passes Control to OS” column of the characteristics table corresponding to the specific register.

4–9. PIC1 (master) interrupt controller must be initialized before PIC2 (slave) interrupt controller. Failure to do so will cause unexpected results.

4–10. The Interrupt Controller must respond to a Non-specific EOI command by resetting the highest in-service bit of those set.

4–11. The Interrupt Controller must respond to a Specific EOI command by resetting the in-service bit that corresponds to the one specified by the command, except that an in-service bit masked by an Interrupt Mask register bit cannot be reset by a Non-specific EOI command when in the Special Mask mode.

4–12. The Interrupt Vector which is returned by the Interrupt Controller on an Interrupt Acknowledge cycle must be as defined in Table 126 on page 65.

4–13. For the device described herein, all the registers defined in sections 4.4.1.1 through 4.4.1.4 must be implemented precisely as described herein.

4–14. During the initialization sequence ICW1 must be written before writing ICW2, ICW2 must be written before writing ICW3, and ICW3 must be written before writing ICW4.

4–15. For the device described herein, all the registers defined in sections 4.4.2.1 through 4.4.4 must be implemented precisely as described herein.

4–16. Writing of OCW1 must be performed before the OCW2 and OCW3 registers are accessed, and

following the ICW1-4 initialization sequence.

4–17. When an in-service bit is set to 1, the Interrupt Controller must inhibit all other interrupts with the same or lower priority.

4–18. When an in-service bit is set to 1, the Interrupt Controller must allow interrupts with a higher priority to cause an interrupt.

4–19. For the device described herein, all the registers defined in sections 4.4.5.1 through 4.4.5.2 must be implemented precisely as described herein.

Chapter 5 Parallel Port Controller

5–1. The Parallel Port Controller must be represented by the Open Firmware properties “name=parallel, device_type=parallel, compatible=chrp,ecp”. The compatible property may also specify the compatible property for the National Semiconductor 87308 in addition to “chrp,ecp”.

5–2. The Parallel Port Controller must protect against damage to the Parallel Port Controller electronics in the case where the device attached to the Parallel Port has its power on and the Parallel Port Controller has its power off.

5–3. The device described herein must support the following modes as defined by the IEEE 1284 Specification: Compatibility mode, Nibble mode (Compatibility mode along with Nibble mode are called Standard Parallel Port, SPP, in this specification), Byte mode (called SPP Extended mode in this specification), Enhanced Parallel Port (EPP) mode (version 1.9), and Extended Capabilities Port (ECP) mode.

5–4. The device described herein must also support the following modes as described herein: Parallel Port FIFO mode and Test FIFO mode.

5–5. This device must be addressed purely with ISA I/O addresses which can be aliased or non-aliased.

5–6. DMA transfers fro the Parallel Port Controller must be 8-bit DMA slave and must support at least the ISA Compatible Timing mode.

5–7. Interrupts generated by the Parallel Port Controller must be high true edge sensitive.

5–8. The Parallel Port Controller registers specified herein must be implemented precisely as defined, including the following characteristics and bit definitions:

a. Table 160, “Data Register (DTR) Characteristics,” on page 84

b. Table 161, “Data Register (DTR),” on page 84

c. Table 162, “Status Register (STR or DSR) Characteristics,” on page 85

d. Table 163, “Status Register (STR or DSR),” on page 85

e. Table 164, “Control Register (CTR or DCR) Characteristics,” on page 85

f. Table 165, “Control Register (CTR or DCR),” on page 86

g. Table 166, “Extended Control Register (ECR) Characteristics,” on page 87

h. Table 167, “Extended Control Register (ECR),” on page 87

i. Table 168, “EPP Address Register (ADDR) Characteristics,” on page 88

j. Table 169, “EPP Data Port 0 Register (DTR) Characteristics,” on page 88

k. Table 170, “EPP Data Port 1 Register (DATA1) Characteristics,” on page 88

l. Table 171, “EPP Data Port 2 Register (DATA2) Characteristics,” on page 89

m. Table 172, “EPP Data Port 3 Register (DATA1) Characteristics,” on page 89

n. Table 173, “ECP Address FIFO (AFIFO) Characteristics,” on page 90

o. Table 174, “Address FIFO Register (AFIFO),” on page 90

p. Table 175, “ECP Data FIFO Register (DFIFO) Characteristics,” on page 90

q. Table 176, “ECP Data FIFO Register (DFIFO),” on page 90

r. Table 177, “Parallel Port FIFO Register (CFIFO) Characteristics,” on page 91

s. Table 178, “Parallel Port FIFO Register (CFIFO),” on page 91

t. Table 179, “Test FIFO Register (TFIFO) Characteristics,” on page 91

u. Table 180, “Test FIFO Register, TFIFO,” on page 91

5–9. The Open Firmware “reg” property information given in Table 159 on page 83 must be provided in the Open Firmware device tree for this device.

5–10. This device must be represented as an ISA device and a child of an ISA bridge node in the Open Firmware device tree.

5–11. This device must be represented in the OF device tree as specified in the *CHRP ISA Parallel Port Device Binding to: IEEE 1275-1994 Standard for Boot (Initialization, Configuration) Firmware*. [23]

5–12. All the reg property fields specified in Table 159 on page 83 must comply with the format given in the ISA (or PCI) bus binding to IEEE Std 1275-1994, as modified by the CHRP binding to IEEE Std 1275-1994.

5–13. The following events must occur during the OF initialization sequence:

a. The Parallel Port Controller must be enabled and activated.

b. If there is I/O Range checking of the Parallel Port Controller it must be disabled.

c. The parallel Port Controller address space must be set up, enabled.

d. The interrupt level and type (edge versus level) must be setup, enabled, and reported in the OF device tree.

e. Eight-bit DMA for the Parallel Port Controller must be: set up, enabled, and reported in the OF device tree.

f. Demand DMA must be enabled

g. Both ECP and EPP 1.9 modes must be enabled.

h. The Parallel Port Clock must be enabled (that is, clock must be operational).

i. The Parallel Port Controller must be set up to disable its outputs when the Parallel Port is disabled.

j. All Parallel Port Controller registers must be initialized to the values indicated in the “Value in Register When OF Passes Control to OS” column of the characteristics table corresponding to the specific

register.

5–14. Writing a Parallel Port device address to this register must initiate an EPP device/register selection operation.

5–15. Accesses to this register must initiate device read or write operations to bits 7-0 of the 32 bit word to be transferred.

5–16. Accesses to this register must initiate device read or write operations to bits 15-8 of the 32 bit word to be transferred.

5–17. Accesses to this register must initiate device read or write operations to bits 23-16 of the 32 bit word to be transferred.

5–18. Accesses to this register must initiate device read or write operations to bits 31-24 of the 32 bit word to be transferred.

Chapter 6 UART Controller

6–1. The UART serial device must be NS16550 compatible, as specified in this chapter.

6–2. The UART serial device must be addressed only with ISA I/O addresses.

6–3. The UART serial device must generate only low-to-high edge sensitive interrupts.

6–4. The Open Firmware device tree node for the UART serial device described in this chapter must follow the specifications given in the CHRP ISA Serial Port Device Binding [28].

6–5. The Open Firmware “reg” property information given in Table 182, “UART Registers,” on page 94 must be provided in the Open Firmware device tree for the UART serial device.

6–6. The UART serial device must be represented as an ISA device and a child of an ISA bridge node in the Open Firmware device tree.

6–7. All the “reg” property fields specified in Table 182, “UART Registers,” on page 94 must comply with the format given in the ISA/EISA/ISA-PnP binding to IEEE Std 1275-1994, as modified by the CHRP binding to IEEE Std 1275-1994.

6–8. The UART serial device must precisely implement the registers defined in Section 6.3, “UART Registers,” on page 94.

6–9. The selection of the TDR/IER versus the DLL/DLH for the UART serial device must be exclusively controlled by the DLAB bit (bit 7) in the Line Control Register.

6–10. The UART serial device must contain a baud rate generator that generates baud rates (within acceptable RS232 tolerances) according to the following relationship: (baud rate x 16) = (frequency input) / DLH|DLL(decimal).

6–11. The UART serial device must have the “clock-frequency” property specified in its device node in the Open Firmware device tree. This property value corresponds to the “frequency input” term in Requirement 6–10, and is not necessarily the oscillator input to the chip that implements the serial device.

Chapter 7 ISA Keyboard/Mouse Controller

7–1. All requirements given in this chapter are specifically for the device represented by the Open Firmware properties “name=8042, device_type=8042, compatible = “chrp,8042”.

7–2. The keyboard and the mouse/auxiliary device are assigned separate interrupts for indicating to the system when user input is available. These interrupts must be separate edge-triggered interrupts.

7–3. A controller that is compatible with the controller described herein must be represented in the device tree as specified in the *CHRP ISA Bus Binding to: IEEE Standard 1275-1994* [15].

7–4. This device must be addressed purely with ISA I/O addresses which can be aliased or non-aliased.

7–5. The registers shown in Table 209 must all be present in the device defined in this chapter. The Open Firmware “reg” property information given in Table 209 must be provided in the Open Firmware device tree for this device.

7–6. All the reg property fields specified in Table 209 must comply with the format given in the *CHRP ISA Bus Binding to: IEEE Standard 1275-1994* [15].

7–7. For the device described herein, all the registers defined in sections 7.3.1 through 7.3.4 must be implemented precisely as described herein.

Chapter 8 Audio

8–1. The audio subsystem must provide one stereo analog audio input, which may typically be a line level or microphone input attached to an external jack.

8–2. The audio subsystem must provide one stereo analog audio output, which may typically be a jack for the attachment of speakers or a headphone.

8–3. An analog-to-digital converter (ADC) must be provided to digitize the stereo analog audio input for delivery at the stereo digital audio output.

8–4. An digital-to-analog converter (DAC) must be provided to convert the stereo digital audio input for delivery at the stereo analog audio output.

8–5. The audio subsystem must be capable of simultaneous recording and playback with 16 bit sample resolution at frequencies of 22.05 and 44.1 KHz.

8–6. The audio subsystem must be configured to a passive state (inputs muted, interrupt and DMA requests disabled) when control is passed to the operating system. System resets must also return the audio subsystem to a passive state.

Chapter 9 ESCC

9–1. When control of the ESCC Controller defined herein is passed from Open Firmware to the Operating System, it must be configured as described herein.

9–2. In the Open Firmware device tree “escc” and “escc-legacy” device nodes, Open Firmware must define the “clock-frequency” property to indicate the frequency of the configured ESCC PCLK. This clock source must be selectable from either the RTxC input or the ESCC’s PCLK. When it is selected from the RTxC input it must have a nominal frequency of 3.684 MHz.

9–3. A controller that is compatible with the controller described herein must be represented in the device tree as specified in the CHRP ESCC Device Binding.

9–4. The registers shown in Table 222 on page 117 must all be present in the device with name= “escc”, device_type= “escc” and compatible= “chrp,es0”.

9–5. The Open Firmware “reg” property information given in Table 222 on page 117 must be provided in the Open Firmware device tree for this device.

9–6. The “escc” device node must have a “ranges” property, along with corresponding “#address-cells”=1 and “#size-cells”=1 properties consistent with the representation of its parent bus. The “ranges” must map the address ranges described by the child node’s “reg” entries.

9–7. All the reg property fields specified in Table 222 on page 117 must comply with the format for the parent bus as defined by the *Mac I/O* [14] binding .

9–8. The registers shown in Table 223 on page 118 must all be present in the device with name= “escc-legacy”, device-type= “escc-legacy” and compatible= “chrp,es1”.

9–9. The Open Firmware “reg” property information given in Table 223 on page 118 must be provided in the Open Firmware device tree for this device.

9–10. The “escc-legacy” device node must have a “ranges” property, along with corresponding “#address-cells”=1 and “#size-cells”=1 properties consistent with the representation of its parent bus. The “ranges” must map the address ranges described by the child node’s “reg” entries.

9–11. All the reg property fields specified in Table 223 on page 118 must comply with the format for the parent bus as defined by the *Mac I/O* [14] binding for Open Firmware.

9–12. The registers shown in Table 224 on page 118 must all be present in the device with name= “ch-a”, device_type= “serial”, and compatible= “chrp,es2”.

9–13. The registers shown in Table 224 on page 118 must all be present in the device with name= “ch-b”, device_type= “serial”, and compatible= “chrp,es3”.

9–14. The Open Firmware “reg” property information given in Table 224 on page 118 must be provided in the Open Firmware device tree for this device.

9–15. The “reg” property fields for the child nodes of the “escc” device must conform to the order described in Table 224 on page 118.

9–16. The registers shown in Table 225 on page 119 must all be present in the device with name= “ch-a”, device-type= “legacy” , and compatible=”chrp,es4”.

9–17. The registers shown in Table 225 on page 119 must all be present in the device with name= “ch-b”, device-type= “legacy” , and compatible=”chrp,es5”.

9–18. The Open Firmware “reg” property information given in Table 225 on page 119 must be provided in the Open Firmware device tree for this device.

9–19. The “reg” property fields for the child nodes of the “escc-legacy” device must conform to the order described in Table 225 on page 119.

9–20. For the device described herein, all the registers defined in sections 9.7.1, “Command Register,” on page 120 through Section 9.7.3, “Enhancement Register,” on page 120 must be implemented precisely as described herein.

9–21. For the device described herein, all the registers defined in sections 9.8.1, “SCC Recovery Count Register,” on page 121 through Section 9.8.10, “Receive Mask,” on page 126 must be implemented precisely as described herein.

Chapter 10 Apple Desktop Bus Controller

10–1. If an ADB is present in the system, it must perform as defined in this chapter.

10–2. The ADB controller device must be represented by the Open Firmware properties “name=adb, device_type=adb, compatible= chrp-adb0”.

10–3. The registers shown in Table 253 on page 130 must all be present in the device defined in this chapter.

10–4. The Open Firmware “reg” property information given in Table 253 on page 130 must be provided in the Open Firmware device tree for this device.

10–5. The interrupt property must be provided for this device in the Open Firmware device tree.

10–6. The Open Firmware device tree node for the ADB serial device described in this chapter must follow the specification given in the Open Firmware *Mac I/O* [14] binding.

10–7. The ADB controller must be implemented with all the registers as defined in :

a. Table 254, “Interrupt Status Register (IST) Characteristics,” on page 131

b. Table 255, “Interrupt Status Register,” on page 131

c. Table 256, “Command/Data Register File (CDRF) Characteristics,” on page 132

d. Table 257, “Interrupt Source Enable (ISE) Characteristics,” on page 132

e. Table 258, “Interrupt Source Enable,” on page 132

f. Table 259, “Data Type Count (DTC) Characteristics,” on page 132

g. Table 260, “Data Type Count Register,” on page 133

h. Table 261, “Error Status Register (ESR) Characteristics,” on page 133

i. Table 262, “Error Status Register,” on page 133

j. Table 263, “Control (CTL) Characteristics,” on page 134

k. Table 264, “Control Register,” on page 134

l. Table 265, “Autopoll Control (ACL) Characteristics,” on page 134

m. Table 266, “Autopoll Control Register,” on page 134

n. Table 267, “Active Device Address low and High (ACD) Characteristics,” on page 135

o. Table 268, “Active Device High and Low Registers,” on page 135

10–8. When system software wants to transfer data to the ADB, the following process must be used:

a. System software must set the Transfer Access Request (TAR) bit to 1.

b. If TAG_IS_EN is set to 1 polls the ADB controller Interrupt Status register, system software must wait for an interrupt from the ADB controller. Otherwise system software must poll the the Interrupt Status

register to determine when the TAG bit is set.

c. When the system software is granted access to the CDRF, the ADB controller must not write to the CDRF.

d. When the TAG bit is set, system software must first write the command byte and then from 0 to 8 bytes of data to the CDRF.

e. Once the command and data bytes are written, system software must set the How Many Bytes (HMB)field of the the Data Type Count register, with the number of bytes written to the command plus data byte registers in the CDRF.

f. If system software expects a response to the command, it must set the Command Response Expected (CRE) bit in the Control register.

g. System software must set the DTB (Data To Bus) bit in the Control register.

h. System software must first clear the TAR bit in the Control register and then clear the TAG bit in the the Interrupt Status register in that order.

i. When the TAG bit is cleared, the ADB controller must sample the DTB bit

j. If the DTB is set, the ADB controller must send the data to the ADB and then clear the DTB.

k. The ADB controller must read any response data and post it to the CDRF.

10–9. When the ADB controller acquires data the following process must be used:

a. The ADB shall fill the 9-byte CDRF with either the command that was sent from the System or the Talk command that caused a response to an autopoll and 0 to 8 data bytes.

b. The ADB controller must set the How Many Bytes, HMB[3:0], bits in the status register to the number of data bytes plus one for the command byte.

c. The ADB controller must set any error status bits (see Section 10.5, “Error Handling,” on page 136) in the Error Status register.

d. The ADB controller must set the DFB interrupt source bit in the Interrupt Status Register.

e. If the data was from an autopoll or SRQ autopoll action, the ADB controller must set the Autopoll Data (APD) bit.

f. System software must clear APD and DFB.

g. The ADB controller must wait until the system software clears the DFB bit before resuming autopolling.

10–10. After processing the commands and data in the CDRF, the ADB controller must resume autopolling, if autopolling is enabled.

10–11. The ADB controller must recognize the error conditions listed in Table 269 on page 136.

10–12. The ADB controller must terminate the read from the bus, append zeros to any undetected bits of an unfinished byte ,post the data that was received and set the DLE error bit in the status register, along with the DFB bit.

10–13. If the interrupt is enabled (DFB_IS_EN), the controller must generate an IRQ to the System.

10–14. The ADB controller shall load the command that was sent into the CDRF, shall set the NRE error in the sta-

tus register, and shall set the DFB bit in the interrupt source register.

10–15. The ADB controller performing the SRQ autopolling sequence must attempt to resolve SRQ autopolling error.

10–16. If the ADB master cannot resolve the error it must return to the main idle loop and resume autopolling, if autopolling is enabled.

10–17. The ADB controller must set the DFB bit to inform the system software that there is data in the CDRF.

10–18. The address of the device must be reflected in the command byte that is stored in the CDRF.

10–19. The ADB controller must present these resets to the platform in manners compliant with the other CHRP requirements.

10–20. The ADB controller must scan incoming commands and data (CDRF) from the keyboard at the default logical device two.

- •When the Control (0x0036), Command (0x0037), and the Power (0x7F7F) keys are detected together (with or without other keys) by the ADB controller, the ADB controller must initiate a power-on system reset (called “hard reset” in the PowerPC processor user manuals).
- •When the Command (0x0037) and the Power (0x7F7F) keys are detected together (with or without other keys) by the ADB controller, the ADB controller must assert the nonmaskable interrupt (NMI).
- •After the NMI is asserted when either the Command (0x0037) or the Power (0x7F7F) key are detected together by the ADB controller as released, the ADB controller must deassert the nonmaskable interrupt (NMI).

10–21. When the ADB controller finds the ADB_RST bit set, it must perform a bus reset as follows:

a. It must drive the ADB Data Out line low for a minimum of 3 ms, then release the line.

b. It must pull up the open collector output to the deassertion state.

c. It must wait the programmed holdoff time, then clear the ADB_RST bit in the Control register, signalling that the ADB bus reset sequence is complete.

10–22. System software must not attempt to use the ADB controller while the ADB_RST bit in the Control register is set to 1 other than to poll the Control register for a change in the ADB_RST bit to a 0.

10–23. The ADB controller must implement timing values for the ADB as specified in Table 270 on page 139.

Chapter 11 IDE Drive Controller

11–1. An IDE Drive Controller that is compliant with the interfaces as defined in this section must be represented in the Open Firmware device tree with a node that has a name property of name=ide, a device_type property of device_type=ide, and a compatible property of compatible=chrp,ide.

11–2. The IDE Bus Master Controller defined herein must supply “Native-PCI” capabilities and PCI bus mastering capabilities concurrently.

11–3. All IDE devices connected to the IDE Bus master Controller defined herein must have DMA capability.

11–4. When control of the IDE Bus master Controller defined herein is passed from Open Firmware to the Operating System, it must be configured as a PCI device as described herein.

11–5. This device must be addressed only with PCI I/O addresses.

11–6. The registers shown in Table 272 must all be present in the device defined in this chapter. The Open Firmware “reg” property information given in Table 272 must be provided in the Open Firmware device tree for this device.

11–7. This device must be represented as an PCI device and a child of a PCI bridge node in the Open Firmware device tree.

11–8. All the reg property fields specified in Table 272 on page 142 must comply with the format given in the ISA (or PCI) bus binding to IEEE Std 1275-1994, as modified by the CHRP binding to IEEE Std 1275-1994.

11–9. For the device described herein, the Physical Region Descriptor, PRD, must be implemented as defined in this section.

11–10. PRD format in memory must be Little-Endian.

11–11. For the device described herein, all the registers defined in sections Table 11.5.1 through Table 11.5.6 must be implemented as described herein.

Chapter 12 SCSI

12–1. All requirements given in this chapter are specifically for the device represented by the Open Firmware properties “name=scsi, device_type=scsi, compatible = chrp,mesh0” unless otherwise noted.

12–2. The registers shown in Table 283 on page 148 must all be present in the device defined in this chapter.

12–3. A MESH SCSI device must make use of properties defined in the *MAC-IO* [14] Open Firmware bindings document.

12–4. The interrupt from the MESH SCSI section of the controller and the DBDMA section of the controller must be wire-OR’ed together so that one interrupt is presented to the system for the complete MESH SCSI controller defined herein.

12–5. For the device described herein, all the registers defined in Section 12.5.1, “Transfer Count 0 Register 0 (XferCount0),” on page 149 through Section 12.5.16, “Selection TimeOut Register (SelTO),” on page 158 must be implemented precisely as described.

12–6. For the device described herein, all the registers and their associated function defined in Appendix A must be implemented precisely as described.

Chapter 13 Graphics

13–1. Graphics controllers must provide both a Big Endian (BE) and a Little Endian (LE) interface to