82C480
8514/A-COMPATIBLE
GRAPHICS CONTROLLER

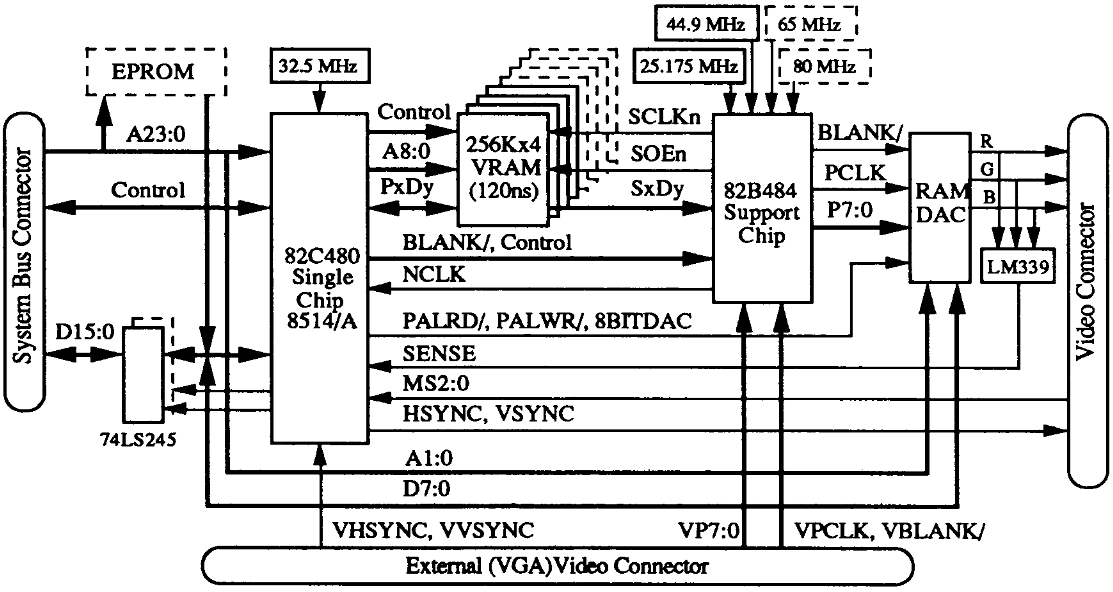
- Fully Compatible with IBM® 8514/A at both register and software ('Adapter Interface' or AI) level
- Runs Windows and Presentation Manager for the 8514/A without special drivers
- Single Chip Solution: 160-Pin Plastic Flatpack
- Integrated Micro Channel, Industry Standard Architecture, and EISA Bus Interfaces
- Autoconfigurable for 8-bit or 16-bit System Interface
- Demultiplexed bus interface resulting in lower chip count. Total of 9 chips required for a complete 8514/A implementation including memory:

  1 82C480 Graphics Controller
  1 74LS245 Bus Transceiver
  1 RAMDAC
  1 LM339 Comparator
  4 256Kx4 VRAMs (+4 for 256 colors)
  1 82B484 Support Chip
  +1 EPROM (optional)

- Supports both 256K and 1M VRAMs

	16 256K	32 256K	4 1M	8 1M
640x480x16-color	x	x	x	x
640x480x256-color	x	x	x	x
1024x768x16-color	x	x	x	x
1024x768x256-color		x		x

- Hardware Graphics Functions:
  - Bit Blt
  - Line Draw
  - Color Mixing
  - Polygon Fill
  - Pattern Fill
  - Scissoring
- External palette DAC support for up to 16 million colors (Autoconfigurable for 6-bit or 8-bit RAMDACs)
- Resolutions supported up to 2540x2048 (either interlaced or non-interlaced) with video rates up to 300 MHz
- Software transparent 1024x768 non-interlaced monitor support
- Low-Power CMOS process



82C480 System Block Diagram

Revision History

Revision	Date	By	Comment
1.3	5/20/90	DR	Document updated to reflect Rev.1 silicon (internal use only) - Register definition added - VESA register and bitfield naming scheme incorporated
1.4	6/13/90	ST	Change document format to Chips Standard Data Sheet Format
1.5	7/11/90	DR	Document updated to reflect Rev. 2 silicon

Table of Contents

Section Page

List of Figures and Tables

Figure Page

Table Page

Introduction

The 82C480 is a 100% register-level IBM 8514/A compatible controller. The 82C480 provides memory interface logic, video control logic, and interface logic to the ISA (PC/AT™) Bus, EISA Bus, and Micro Channel (MCA).

With 512 KBytes of display memory, the 82C480 supports 16 and 256 displayable colors in 640x480 resolution and 16 with 1024x768 resolution. With 1Mbyte of display memory, it supports 256 displayable colors with 1024x768 resolution. An external RAMDAC supports 256K (6-bit DAC) or 16 million (8-bit DAC) total colors.

The 82C480 supports the following modes selectable via software:

- VGA mode (pass through video from VGA subsystem)
- Advanced Function mode (8514/A 1024x768 or 640x480)

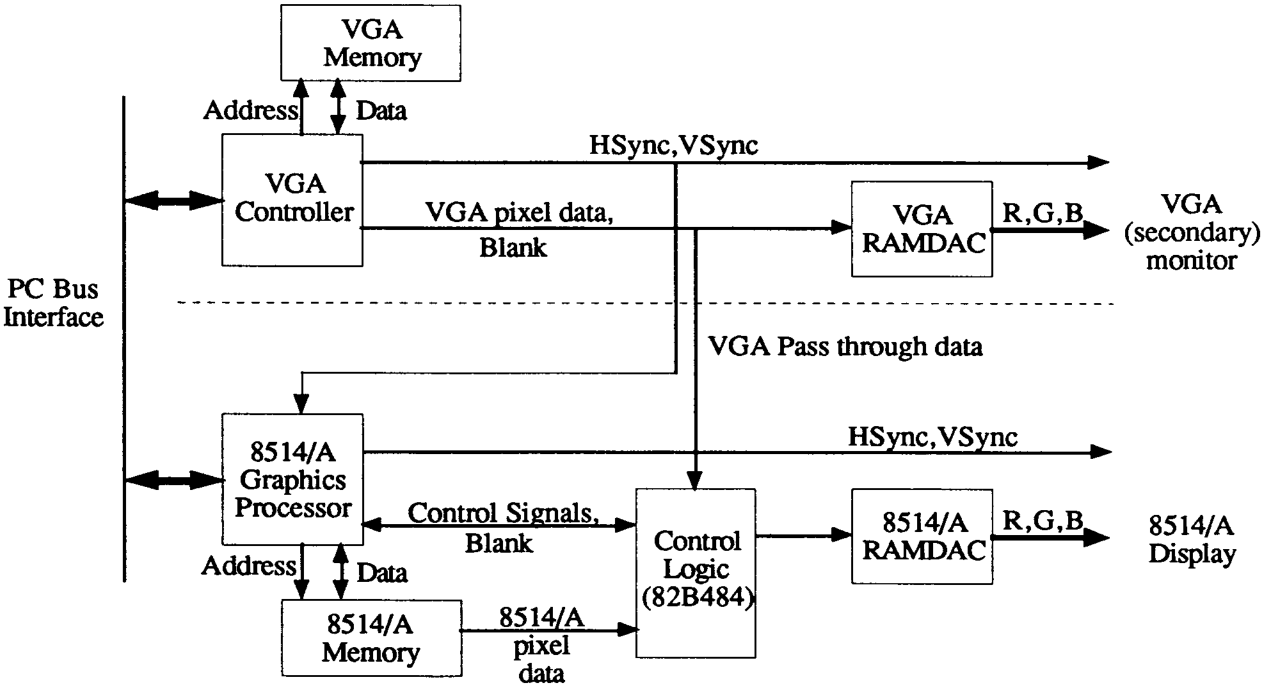
The 82C480 defaults to VGA mode on reset. In the Advanced Function mode, the 82C480 supports 640x480 or 1024x768 resolution as well as providing more complex hardware functions such as BitBlt, line drawing, polygon fill, patterns, raster operations, and scissoring (post-clipping).

Software support for the 82C480 is provided via the Chips and Technologies Adapter Interface (AI) and drivers for major software applications. Any software which supports the IBM 8514/A will also support a compatible 82C480 implementation.

The 82C480 is supplied in a 160-pin PFP package.

Bus Interface

The 82C480 allows selection of either ISA (PC/AT) or MCA (Micro Channel) Bus Interface by detecting a strapping option during reset. All control signals



VGA /8514 Pass -Through Interface

for both interface types are integrated onto the single 8514/A-compatible chip.

The 82C480 supports both 8-bit and 16-bit CPU interfaces.

ROM Interface

The 82C480 supports an optional 8-bit ROM for Power-On-Self-Test (POST) code which may be implemented in AT-bus systems without using additional external components except the ROM chip. The ROM address is decoded and the ROMCS/ pin is asserted to enable ROM data onto the data bus. A 16-bit ROM may be implemented with two ROM chips and one additional external PAL.

Video initialization code and support functions may be included in the POST ROM, incorporated into the system BIOS, loaded from disk at system initialization as a TSR, or incorporated directly into drivers.

VRAM Interface

Like the IBM 8514/A, the 82C480 supports 256K (64Kx4) VRAMs. In addition, the 82C480 also supports 1M (256Kx4) VRAMs, allowing a lower cost, lower chip-count 8514/A implementation. The 82C480 supports a faster memory clock (40 MHz) to maximize use of VRAM bandwidth.

The 82C480 supports five memory cycle types:

1. Read (page mode)
2. Write (page mode)
3. Read-Modify-Write (page mode)
4. RAS-only refresh
5. Data Transfer Cycle

Extensions and Extended Registers

The 82C480 chip provides enhanced modes of operation which are not available on the IBM 8514/A. These extended features are controlled via bits in extension registers.

The 82C480 has extended the 8514/A architecture to make virtually all of the registers readable. This improves the testability of the chip and makes state saving easy to accomplish.

External Color Palette

The 82C480 supports the programming of an external color palette DAC (RAMDAC) by decoding the CPU addresses and generating the read and write signals for the external palette.

Normally, each RAMDAC analog output provides 6-bit resolution (64 shades of color on each of the analog R, G, and B outputs). However, a pin is provided by the 82C480 for selecting 8-bit-per-color mode for the DAC (e.g., if using an INMOS IMSG-178 or BT478) which provides 256 shades of color on each RGB output. If this capability is not required, the 8BITDAC pin may instead be used as a general purpose output.

Performance

The 82C480 provides up to 2 times the performance of the IBM 8514/A hardware. Projected performance for major graphics operations are listed below (Mp/s = million pixels per second):

Function	82C480	8514/A	Relative Performance
Clear	28.6 ms	33.3 ms	1.2x
Horizontal Solid Line	14.8 Mp/s	7.3 Mp/s	2.0x
Vertical Line	3.7 Mp/s	2.9 Mp/s	1.3x
BitBlt	14.8 Mp/s	9.6 Mp/s	1.5x

Bit-Block Transfer (BITBLT)

The 82C480 uses an internal 32/40 pixel register file for BitBlts (8 words x 32/40 bits). This allows reading a block of pixels in page mode, then writing them using page mode RMW cycles. The 82C480 has a 32/40-bit internal data path, and its faster clock rate results in faster BitBlt performance.

Monitor Support

The 82C480 supports the following analog monitors:

- IBM Monochrome Display 8503 or compatible
- IBM High Resolution Monochrome Display 8507 or compatible
- IBM Color Display 8512 or compatible
- IBM Color Display 8513 or compatible
- Interlaced monitors supporting high resolution (IBM 8514, 8515, or compatible)
- Multisync and compatible monitors
- Any interlaced or non-interlaced monitor with a resolution of up to 2560x2048 and supporting video rates of up to 300 MHz

Line Draw

Line draw functions (straight lines only) are handled in hardware by the 82C480 chip. Arcs, circles, and other higher-level shapes must be decomposed to a series of short straight-line segments. The 82C480 uses the Bresenham algorithm for line drawing.

Text and Alphanumeric Support

The 82C480 chip (like the 8514/A) does not support a separate text mode in hardware. Text display is supported via graphics modes, so text may be any size.

The AI supports 3 standard text modes at 1024x768 resolution:

- Mode 0: 85x38 (12x20 character cell)
- Mode 2: 128x54 (8x14 character cell)
- Mode 3: 146x51 (7x15 character cell)

The AI also supports an 80x34 text mode (mode 1) at 640x480 resolution (8x14 character cell).

Video Subsystem Total Chip Count

Using the 82C480, a complete 8514/A-compatible 8-bit video subsystem can be built with just 9 ICs (10 ICs for a 16-bit system), including display memory, as shown in the table below:

Qty	Chip Type
1	82C480 Graphics Controller Chip
1	74LS245 Transceiver (2 for 16-bit interface)
1	IMSG176-65 RAMDAC
1	LM339 Comparator
4	256Kx4 VRAM (120 ns) (+4 for 8 planes)
1	82B484 Support Chip
9	Total
+1	27256 POST ROM (optional)

Additional components required would be 25.175, 32.5, 44.900 and 65.000 MHz oscillators, 15-pin video connector, LM334 Current reference, 1N4148 diode, and various resistors and capacitors. The indicated configuration (4 256K x 4 VRAM's) supports 640x480 non-interlaced (25.175 MHz.) 16-color and 256-color modes and 1024x768 interlaced (44.900 MHz.) and non-interlaced (65.000 MHz.) 16-color mode.

Adding 256-color support for 1024x768 would require four additional 256Kx4 VRAM chips (for a total of 8 256K x 4 VRAM's).

If a 'Power-On-Self-Test' (POST) ROM is implemented in a Micro Channel bus configuration, two additional 8-bit address latches (74LS373 or equivalent) are also required.

Package

The 82C480 is available in a 160-pin plastic flat pack (PFP).

Complete descriptions of all 82C480 pins are included in this document starting on the next page. The pins are separated into the following logical groups for discussion: Bus Interface, Display memory, Video, Clock, Power, and Ground.

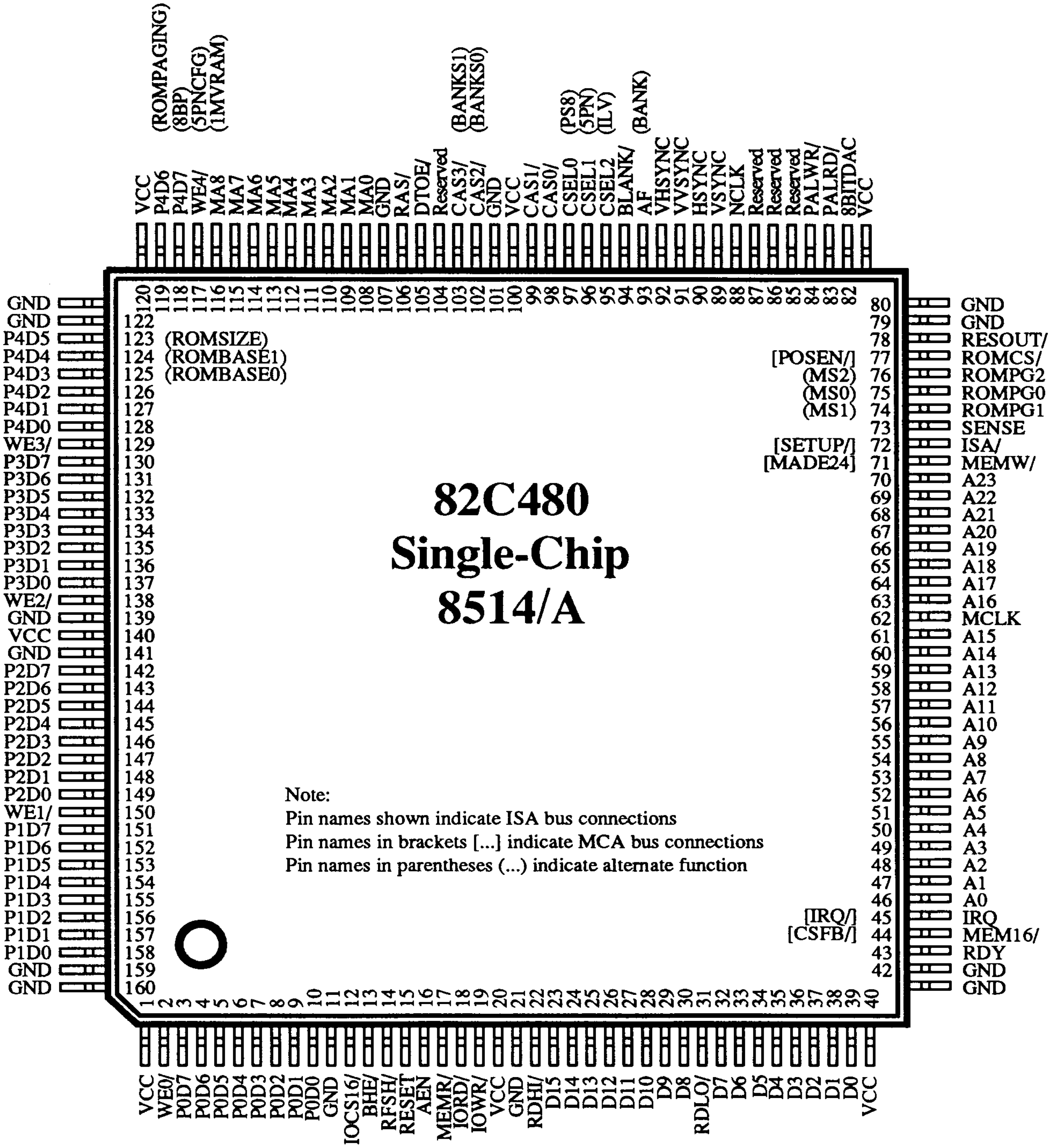
82C480 Pin Usage Summary	
Bus Interface:	60
Display Memory:	61
Video:	9
Clock:	6
Test:	3
Power:	7
Ground:	14
Total:	160

82C480 System Diagrams

Included at the end of this document are schematic examples of the following:

1. 8/16-bit ISA (PC/AT) Bus Interface
   16-bit Micro Channel (MCA) Bus Interface
2. Memory Interfacing (40-bit Configuration A)
   Memory Interfacing (80-bit Configuration B)
3. Video/Clock Logic (82B484 - IMSG176/8)
   Video/Clock Logic (82B484 - BT471/8)

82C480 Pinouts



Note:
Pin names shown indicate ISA bus connections
Pin names in brackets [...] indicate MCA bus connections
Pin names in parentheses (...) indicate alternate function

82C480 Pinouts

82C480 PIN DESCRIPTIONS

System Bus Interface

Pin #	Pin Name	Type	Active	Description
23	D15	I/O	High	System Upper Data Bus All byte steering is done internally, so the 82C480 may be used without data transceivers in motherboard applications; 74LS245 or equivalent transceivers are recommended for adapter cards to meet the 24 mA drive requirement.
24	D14	I/O	High	
25	D13	I/O	High	
26	D12	I/O	High	
27	D11	I/O	High	
28	D10	I/O	High	
29	D9	I/O	High	
30	D8	I/O	High	
32	D7	I/O	High	System Lower Data Bus
33	D6	I/O	High	
34	D5	I/O	High	
35	D4	I/O	High	
36	D3	I/O	High	
37	D2	I/O	High	
38	D1	I/O	High	
39	D0	I/O	High	
70	A23	In	High	System Address Bus (latched internally for the MCA bus on the falling edge of CMD/) In MCA designs, connect A23:0 to A23:0. In ISA (PC/AT) Bus designs, connect A19:0 to SA19:0 and A23:20 to ground. In MCA designs, A14:0 must be latched externally by ADL/ or CMD/ for the ROM. In ISA bus designs, the address inputs need not be latched.
69	A22	In	High	
68	A21	In	High	
67	A20	In	High	
66	A19	In	High	
65	A18	In	High	
64	A17	In	High	
63	A16	In	High	
61	A15	In	High	
60	A14	In	High	
59	A13	In	High	
58	A12	In	High	
57	A11	In	High	
56	A10	In	High	
55	A9	In	High	
54	A8	In	High	
53	A7	In	High	
52	A6	In	High	
51	A5	In	High	
50	A4	In	High	
49	A3	In	High	
48	A2	In	High	
47	A1	In	High	
46	A0	In	High	
72	ISA/ [SETUP/]	In	Low	The primary function of this pin is to indicate whether the 82C480 should be configured for the MCA or ISA (PC/AT) bus interface. This pin may be tied to ground or RESOUT/ in ISA Bus configurations. The MCA interface also drives this pin to select the 82C480 during POS (if low, the POS registers at 100-102h are not accessible and all others are if the enable bit in register 102h is set).

82C480 PIN DESCRIPTIONS

System Bus Interface

<table>
<tr><th>Pin #</th><th>Pin Name</th><th>Type</th><th>Active</th><th>Description</th></tr>
<tr><td>15</td><td>RESET</td><td>In</td><td>High</td><td>RESET=1 resets the 82C480 and tristates all output pins except RESOUT/. Also puts internal counters in a known state for chip test.</td></tr>
<tr><td>78</td><td>RESOUT/</td><td>Tri-O</td><td>Low</td><td>RESOUT/ is an inverted and delayed RESET. It is used to enable any external tristate buffers required to drive programming options on the 82C480 pins; these options are latched on the falling edge of RESET. RESOUT/ is guaranteed to stay low long enough to meet the hold time requirement from RESET for any programming option. Some options, such as on the MA8 pin, may be driven by RESOUT/ directly.</td></tr>
<tr><td>13</td><td>BHE/</td><td>In</td><td>Low</td><td>Byte High Enable. Low indicates that the high order byte at the current word address is being accessed. Along with A0, indicates which bytes are transferred over the bus (all byte steering is done internally):
<table>
<tr><th>BHE/</th><th>A0</th><th>Effect</th><th></th></tr>
<tr><td>0</td><td>0</td><td>Both bytes on D15:0</td><td>(16-bit slot only)</td></tr>
<tr><td>0</td><td>1</td><td>High byte on D15:8</td><td>(16-bit slot only)</td></tr>
<tr><td>1</td><td>0</td><td>Low byte on D7:0</td><td>(8- or 16-bit slot)</td></tr>
<tr><td>1</td><td>1</td><td>High byte on D7:0</td><td>(8-bit slot only)</td></tr>
</table>
The 82C480 configures itself to 8- or 16-bit slots. BHE/ is located on the 16-bit bus extension in PC/AT systems, so BHE/ has an internal pullup to hold it inactive in case the card is installed in an 8-bit slot.</td></tr>
<tr><td>22
31</td><td>RDHI/
RDLO/</td><td>Tri-O
Tri-O</td><td>Low
Low</td><td>Direction control for high and low external data bus transceivers. These outputs are tristate when RESET is active, so have internal pullups to prevent driving the bus.

0 = Read data from 82C480
1 = Write data into 82C480</td></tr>
<tr><td>71</td><td>MEMW/ [MADE24]</td><td>In</td><td>Both</td><td>In ISA (PC/AT) bus interface, currently unused, but should be connected to SMEMW/ for compatibility with future products. In MCA interface, MADE24=1 indicates only 24 bits of address are being decoded; the 82C480 will not respond to any cycle in MCA mode unless this pin is high. MADE24 may also be used as an active-high chip select to decode 32-bit addresses.</td></tr>
</table>

<table>
<tr><th>Pin #</th><th>Pin Name</th><th></th><th>Type</th><th>Active</th><th>Description</th></tr>
<tr><td>17</td><td>MEMR/</td><td>[CMD/]</td><td>In</td><td>Low</td><td>In MCA, indicates a command cycle (valid data on the bus). Driven by CMD/ from MCA, VGACMD/ from CHIPS/250. In ISA bus, indicates a memory read cycle.</td></tr>
<tr><td>19</td><td>IOWR/</td><td>[S0/]</td><td>In</td><td>Low</td><td>In ISA (PC/AT) bus interface, indicates an I/O Write cycle. In MCA bus interfaces, indicates Status bit 0 (S0/).</td></tr>
<tr><td>18</td><td>IORD/</td><td>[S1/]</td><td>In</td><td>Low</td><td>In ISA (PC/AT) bus interface, indicates an I/O Read cycle. In MCA bus interfaces, indicates Status bit 1 (S1/).</td></tr>
<tr><td>16</td><td>AEN</td><td>[MIO/]</td><td>In</td><td>Both</td><td>In ISA (PC/AT) bus interface, defines valid I/O address: 0 = valid I/O address, 1 = Invalid I/O address (DMA cycle). If single-cycle DMA is used, memory addresses will be on the bus at the same time that IORD/ or IOWR/ is active. The 82C480 will not respond to IORD/ or IOWR/ while AEN=1. In MCA bus interface, indicates memory or I/O cycle: 1 = memory cycle, 0 = I/O cycle.
<table>
<tr><th>MIO/</th><th>S1/</th><th>S0/</th><th>Cycle Type</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>-reserved-</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>I/O Read</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>I/O Write</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>-reserved-</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>-reserved-</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Memory Read</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Memory Write (not used on 480)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>-reserved-</td></tr>
</table></td></tr>
<tr><td>14</td><td>RFSH/</td><td></td><td>In</td><td>Low</td><td>Active low signal indicating Refresh cycle. When this pin is low, the chip is not accessible. This pin and MA-DE24 are effectively active-high chip selects.</td></tr>
<tr><td>43</td><td>RDY</td><td></td><td>Tri-O</td><td>High</td><td>Ready. Driven low to indicate that the current cycle should be extended with wait states. Driven high at the end of cycle to indicate 'ready', then tristated. This signal is normally tristated and is only driven low if the 82C480 cannot respond immediately to I/O requests (or when the ROM is addressed in MCA) to insert the necessary wait states. Another way to look at this pin is as an active-low wait-state request line, instead of an active-high ready line.</td></tr>
</table>

82C480 PIN DESCRIPTIONS

Pin #	Pin Name		Type	Active	Description
44	MEM16/	[CSFB/]	Tri-O	Low	In ISA (PC/AT) bus configurations, this pin is currently unused. It is reserved to indicate 16-bit memory cycle capability. For compatibility with future products, this pin should be connected to MEMCS16/ in ISA bus configurations. In MCA bus configurations, this pin is called 'Card Select Feedback'; it indicates any valid access to the 82C480. It is an unlatched decode of A23:0, MIO/, and MADE24.
12	IOCS16/	[DS16/]	Tri-O	Low	In ISA (PC/AT) bus interface, indicates 16-bit I/O cycle. In MCA Interface, indicates 16-bit Memory or I/O cycle. Asserted by the 82C480 to indicate that the chip is capable of transferring 16 bits over the bus at the requested address. In the MCA bus, DS16/ is active for all 82C480 accesses except those to the RAMDAC, POS, and ROM.
45	IRQ	[IRQ/]	Tri-O	Both	In the ISA (PC/AT) bus, this pin is tristated when interrupts are not enabled, low when interrupts are enabled but no interrupt is pending, and high when interrupts are enabled and an interrupt is pending. In the MCA bus, this pin functions as an active-low open-collector output. The interrupt request pin is normally connected to IRQ9. IRQ9/ may be shared by multiple controllers on the MCA bus; in the ISA bus, only one controller at a time may have IRQ9 enabled. This pin has high drive capability (IOL = -16mA). While this is still not enough to meet the MCA bus specification of 24mA, the board designer may still want to connect IRQ/ directly to the bus, eliminating the need for a 74LS125 driver.
77	ROMCS/	[POSEN/]	I/O	Low	In ISA (PC/AT) bus configurations, this pin indicates access to ROM space. In MCA configurations, this pin indicates access to ROM space or to the POS ID registers (I/O address 100-101h and SETUP/ active). In 82C480-based designs with ROM, the POS registers map into the ROM. In systems without ROM, the same information may be provided from external TTL drivers activated by A0 and POSEN/.

Note: Since ROMCS/ is in input mode during reset, an internal pullup resistor ensures the ROM is held in a quiescent state.

Pin #	Pin Name		Type	Active	Description
76	ROMPG2	(MS2)	I/O	High	ROM Page (ROM Address msb) outputs / Monitor ID inputs. During reset, these pins are inputs sampled on the falling edge of RESET (these pins have internal 50K pullups) to latch the state of the Monitor ID inputs (MS2:0) from the video connector, which may then be read from the Subsystem Status register. If a ROM is connected to ROMPG2:0, MS2:0 are normally driven onto these pins with a tri-state buffer enabled by RESOUT/. However, if P4D6 is low at the falling edge of RESET, these pins remain inputs so MS2:0 may be connected to these pins directly. If P4D6 is high at the falling edge of RESET, these pins then become outputs driven by ROM_PAGE_SEL[2:0]. If ROM paging capability is desired, these pins connect to ROM address inputs A14:12 (32K) or A12:10 (8K).
74	ROMPG1	(MS1)	I/O	High	
75	ROMPG0	(MS0)	I/O	High	

MS2	MS1	MS0	Monitor Type	
0	0	0	reserved	
0	0	1	8507 (Mono)	(31.5/35.5 KHz)
0	1	0	8514 (Color)	(31.5/35.5 KHz)
0	1	1	reserved	
1	0	0	reserved	
1	0	1	8503 (Mono)	(31.5 KHz)
1	1	0	8512/13 (Color)	(31.5 KHz)
1	1	1	reserved	

Note: The ROMPG2:0 outputs are forced high during SETUP mode, independent of the ROM Page register contents.

Note: ROM_PG_SEL [2:0] register bits are set to 1 on RESET, so that the ROM page defaults to ROM page 7.

Note: In MCA systems, address bits A11:0 must be latched for the ROM. Addresses may be latched by 74LS373 or equivalent transparent latches on the falling edge of CMD/ or by 74LS374 or equivalent registers on the rising edge of ADL/. In ISA systems, ROM addresses A11:0 are held valid during the entire bus cycle and need not be latched.

Note: In MCA systems, A1:0 may be connected directly to the RAMDAC. IMSG176 or equivalent RAMDACs latch their register select inputs internally on the leading edge of PALRD/ or PALWR/ and the MCA bus holds addresses valid long enough to meet hold time requirements. If addresses are latched for a ROM however, use the latched address hence minimizing the bus signal fanout.

Pin #	Pin Name		Type	Active
118	P4D7	(8PLANE)	I/O	High
119	P4D6	(ROM PAGING)	I/O	High
123	P4D5	(ROMSIZE)	I/O	High
124	P4D4	(ROMBASE1)	I/O	High
125	P4D3	(ROMBASE0)	I/O	High
126	P4D2	(reserved)	I/O	High
127	P4D1	(reserved)	I/O	High
128	P4D0	(reserved)	I/O	High
130	P3D7		I/O	High
131	P3D6		I/O	High
132	P3D5		I/O	High
133	P3D4		I/O	High
134	P3D3		I/O	High
135	P3D2		I/O	High
136	P3D1		I/O	High
137	P3D0		I/O	High
142	P2D7		I/O	High
143	P2D6		I/O	High
144	P2D5		I/O	High
145	P2D4		I/O	High
146	P2D3		I/O	High
147	P2D2		I/O	High
148	P2D1		I/O	High
149	P2D0		I/O	High
151	P1D7		I/O	High
152	P1D6		I/O	High
153	P1D5		I/O	High
154	P1D4		I/O	High
155	P1D3		I/O	High
156	P1D2		I/O	High
157	P1D1		I/O	High
158	P1D0		I/O	High
3	P0D7		I/O	High
4	P0D6		I/O	High
5	P0D5		I/O	High
6	P0D4		I/O	High
7	P0D3		I/O	High
8	P0D2		I/O	High
9	P0D1		I/O	High
10	P0D0		I/O	High

Description

VRAM Data Bus. The first digit in the signal name indicates the position of the pixel within the 'nugget'. The second indicates the bit position ('plane') within the pixel.

For horizontal resolutions of 1024 or less, four-pixel nuggets may be used and the minimum number of 256Kx4 VRAMs required is four (connected to P3:0Dn with P4Dn unconnected). For 1280 horizontal resolution, the minimum number of 256Kx4 VRAMs required becomes five (connected to P4:0Dn).

The P4Dn pins are also used for hardware configuration options. On the falling edge of RESET, the state of the P4Dn pins are latched internally. These 8 bits contain internal pull-ups which cause the options to default to the correct values for 8514/A compatibility. An external LS244 buffer (enabled by RESOUT/) may be used to drive non-default ('0') configuration values on any subset of the P4Dn pins. The inputs of the buffer may be strapped to ground or tied to jumpers for user-configurable options. Alternatively, P4Dn inputs may be connected to 100Ω pulldown resistors in 32-bit pixel data bus configurations. Software may write over any configuration options latched from P4Dn which eliminates the need for most strapping options.

P4D7 is saved at reset as a read-only bit in Subsystem Status register (42E8) bit-7 (SUBSYS_STAT[7]). It is also accessible (read/write) in Extended Configuration register #2 (5AE8h) bit-7 (EC2[7]). It is interpreted as the **8PLANE** ('8 Bit Planes') bit:

0 = 4-Plane/16-color configuration
1 = 8-Plane/256-color configuration (default)

P4D6 is saved in EC2[6] as **ROM Paging**:

0 = ROMPG2:0 are input pins readable at SUBSYS_STAT[6:4] (Monitor Sense pins).
1 = ROMPG2:0 output ROM_PAGE_SEL[2:0] (Monitor Sense bits are latched from ROMPG2:0 on the falling edge of RESET only)

P4D5 is saved in EC2[5] as **ROM Size**:

P4D5: 0=32K, 1=8K ROM space in system memory

P4D4:3 are saved in EC2[4:3] at reset. These are interpreted by hardware as **ROMBase**:

P4D4:3	8K MCA	8K ISA	32K
00	C8000h	C6000h	D0000h
01	D8000h	D8000h	D8000h
10	C0000h	C0000h	C0000h
11 (default)	C6000h	C8000h	C8000h

82C480 PIN DESCRIPTIONS

Display Memory Interface

Pin #	Pin Name		Type	Active	Description
116	MA8	(1MVRAM)	I/O	High	VRAM Address. On the falling edge of RESET, the state of MA8 is latched into extension register EC2[10]. If 0, 256Kb VRAMs are being used (same as the 8514/A); if 1, 1Mb VRAMs are being used, and the 82C480 reinterprets VRTCFG register bits to maintain register compatibility with the 8514/A. MA8 (only) has an internal 50K pull-up resistor, so the default is 1. To select 256K VRAMs, drive MA8 with RESOUT/.
115	MA7		Tri-O	High	
114	MA6		Tri-O	High	
113	MA5		Tri-O	High	
112	MA4		Tri-O	High	
111	MA3		Tri-O	High	
110	MA2		Tri-O	High	
109	MA1		Tri-O	High	
108	MA0		Tri-O	High	
106	RAS/		Tri-O	Low	VRAM Row Address Strobe. An internal pullup keeps the VRAMs in a quiescent state during reset. During normal operation, RAS/ will be observed active (low) except when a RAS precharge cycle is necessary.
103	CAS3/	(BANKS1)	I/O	Low	VRAM Column Address Strobes for memory banks 0–3. CAS3:1/ are unused in single-bank configurations (such as the basic four 1M-VRAM minimum configuration). CAS3:2/ have internal pullups and are sampled at reset and saved in EC2[9:8] as the number of VRAM banks installed: 00=1, 01=2, and 11=4 banks.
102	CAS2/	(BANKS0)	I/O	Low	
99	CAS1/		Tri-O	Low	
98	CAS0/		Tri-O	Low	
117	WE4/	(5PN)	I/O	Low	VRAM Write Enables for pixels 4:0 of the nugget, respectively. WE4/ is only used in 5-pixel nugget configurations (it is unused in 4-pixel nugget configurations, such as the basic four 1M-VRAM minimum configuration). WE4/ has an internal pullup and should be connected to RESOUT/ or GND to configure the 82C480 for 4-pixel nuggets. The state of WE4/ at reset (called the 5PN or '5-Pixel Nugget' bit) is saved in Extension Register EC2[11] (read/write) and in the Memory Control register MEM_CNTL[0] as part of the horizontal configuration (HORCFG) bitfield.
129	WE3/		Tri-O	Low	
138	WE2/		Tri-O	Low	
150	WE1/		Tri-O	Low	
2	WE0/		Tri-O	Low	
105	DTOE/		Tri-O	Low	Data Transfer / Output Enable for all VRAM chips
104	Reserved		n/c	n/a	This pin is reserved for future use and should be left unconnected.

82C480 PIN DESCRIPTIONS

Pin #	Pin Name		Type	Active	Description
62	MCLK		In	High	Memory Clock. MCLK clocks everything except the CRT control logic (which uses NCLK). The MCLK frequency is selected to match the speed of the RAMs used:

MCLK	VRAM Speed
25 MHz	150 ns
32 MHz	120 ns
40 MHz	100 ns (Max MCLK frequency)

Pin #	Pin Name		Type	Active	Description
88	NCLK		In	High	Double Nugget Clock. Used to generate timing inside the 82C480 chip.

Note: The NCLK frequency is 1/8 or 1/10 of the selected video clock frequency (depending on whether the memory array is configured for 4- or 5-pixel nuggets). CLKSEL2:0 are used to select the video clock from up to eight frequencies (see the CLKSEL selection table below). The selected video frequency is used to clock the external video shift registers and RAMDAC. 82C480 internal logic is clocked on the rising edge of NCLK. HSYNC is delayed by one half NCLK cycle and is clocked on the falling edge of NCLK.

Pin #	Pin Name		Type	Active	Description
95	CSEL2	(INTERLEAVE)	Tri-O	High	Video Clock Select. If BLANK/=0 (blank active), used to select one of up to eight frequencies using external clock selection logic (only the first two frequencies are required for IBM 8514/A compatibility).
96	CSEL1	(5PN)	Tri-O	High	
97	CSEL0	(PS8)	Tri-O	High	

CSEL 2	CSEL 1	CSEL 0	Frequency Selected	Resolution	
0	0	0	25.175 MHz	640x480	non-interlaced
0	0	1	44.900MHz	1024x768	interlaced
0	1	0	65.000MHz	1024x768	non-interlaced
0	1	1	80.000MHz	1280x1024	interlaced
1	0	0	User-specified	User-specified	
1	0	1	User-specified	User-specified	
1	1	0	User-specified	User-specified	
1	1	1	User-specified	User-specified	

Note: The selected frequency is used to clock the external video shift registers and RAMDAC. It is divided by eight or ten (depending on whether the memory array is configured for 4-pixel or 5-pixel nuggets) for input on NCLK. CLKSEL2:0 are controlled by EC3[10:8] (CLKSEL0 is duplicated in ADVFUNC_CNTL[2]).

If BLANK/=1 (blank inactive), these pins output INTERLEAVE, 5PN (5-pixel nugget), and PS8 (Pseudo-8-plane mode) bits. 5PN is MEM_CNTL[0]. PS8 mode is selected by setting VRTCFG=00 (MEM_CNTL[3:2]) (normal setting is 01).

Pin #	Pin Name		Type	Active	Description
93	AF	(BANK)	Tri-O	High	If BLANK/=0, indicates Advanced Function (ADVFUNC_CNTL[0]); 1=82C480 drives the video connector, 0=external logic (usually a VGA) drives the video connector. If BLANK/=1, indicates BANK, used by external logic to select which serial output enables (SOE3:0/) to activate for a particular scan line in 4-bank configurations: BANK=0 selects SOE1:0/; BANK=1 selects SOE3:2/. In two-bank configurations, BANK=0 selects SOE0/ and BANK=1 selects SOE1/.

82C480 PIN DESCRIPTIONS

Test, Power, and Ground

Pin #	Pin Name	Type	Active	Description
83	PALRD/	Tri-O	Low	Connected to the Read input of the Palette DAC (IMSG176, BT471, or compatible). Asserted when the 82C480 is enabled and an I/O Read occurs from addresses 2EAh-2EDh. Tristate during reset. Internal 50K pullup.
84	PALWR/	Tri-O	Low	Connected to the Write input of the Palette DAC (IMSG176, BT471, or compatible). Asserted when the 82C480 is enabled and an I/O Write occurs to addresses 2EAh-2EDh (or 3C6h-3C9h in VGA pass-through mode). Tristate during reset.
82	8BITDAC	I/O	High	Used to control the '6/8-Bit-Analog-Output' pin of the Palette DAC (0=6-bit, 1=8-bit) if using an IMSG178, BT478, or compatible. This pin is sampled at reset, the state is loaded into EC2[13], then this pin becomes an output driven by EC2[13]. If EC2[13] is written by software, the chip attempts to drive the pin to the programmed level; if EC2[13] is read, the value read is the actual state of the pin, not the register bit. Thus this pin can be used to control the DAC output resolution (with default to either 6-bit or 8-bit operation depending on whether a 10K resistor is connected to the pin as a pullup or a pulldown). If it is desired to use a fixed size RAMDAC, this pin should be connected to ground via a 5.1Ω (max) resistor for 6-bit or to +5V via a 5.1Ω (max) resistor for 8-bit operation. (Automatic DAC Size Detection: Patent Pending)
90 89 94	HSYNC VSYNC BLANK/	Tri-O Tri-O Tri-O	Both Both Low	Horizontal and Vertical Sync signals for the monitor and Blanking signal for the external palette DAC. Sync polarities are programmable. Tristated during reset.
92 91	VHSYNC VVSYNC	In In	Both Both	Horizontal and Vertical Sync signals from external VGA. When not in Advanced Function (AF) mode (i.e., 82C480 not enabled), these signals are used to drive the HSYNC and VSYNC outputs instead of the 82C480 H/V Sync.
73	SENSE	In	High	Sense Input for analog output and monochrome/color monitor connection testing.

For compatibility, this pin should be connected to the outputs of an LM339 comparator. The comparator 'plus' inputs are connected to a 0.34V comparison voltage (corresponding to approximately half brightness) and the 'minus' inputs are connected to analog R, G, and B (maximum analog output voltage is 0.7V). If the R, G, or B output voltage is below the comparison voltage, the corresponding comparator output will be inactive (tristated/high); if above the comparison voltage, the comparator output will be driven low. Therefore, SENSE will be high if R, G, and B are all below half brightness. SENSE will be low if any one of R, G, or B is higher than half brightness. The SENSE input may be read via software as bit-0 of the Display Status register at 02E8h.

82C480 PIN DESCRIPTIONS

Test, Power, and Ground

Pin #	Pin Name	Type	Active	Description
87	Reserved	n/c	n/a	This pin is reserved for future use and should be left unconnected for normal operation.
85	Reserved	n/c	n/a	This pin is reserved for future use and should be left unconnected for normal operation.
86	Reserved (GND)	n/c	n/a	This pin is reserved for future use and should be grounded for compatibility with future chip revisions.
1	VCC	P	-	Power Pins
20	VCC	P	-	
40	VCC	P	-	
81	VCC	P	-	
100	VCC	P	-	
120	VCC	P	-	
140	VCC	P	-	
11	GND	P	-	Ground Pins
21	GND	P	-	
41	GND	P	-	
42	GND	P	-	
79	GND	P	-	
80	GND	P	-	
101	GND	P	-	
107	GND	P	-	
121	GND	P	-	
122	GND	P	-	
139	GND	P	-	
141	GND	P	-	
159	GND	P	-	
160	GND	P	-	

82C480 Registers

Register Mnemonic	Register Name		Read Access	Write Access	Bits	Reg Type	Page
POS_2	Setup Control		0102	0102	1	POS	21
DISP_STAT	Display Status		02E8*	N/A	3	8514/A	21
H_TOTAL	Horizontal Total		26E8*	02E8*	9	8514/A	22
DAC_MASK	DAC Mask		02EA	02EA†	8	RAMDAC	22
DAC_R_INDEX	DAC Read Index		02EB	02EB†	8	RAMDAC	23
DAC_W_INDEX	DAC Write Index		02EC	02EC†	8	RAMDAC	23
DAC_DATA	DAC Data		02ED	02ED†	8	RAMDAC	24
H_DISP	Horizontal Displayed		06E8*	06E8*	8	8514/A	24
H_SYNC_STRT	Horizontal Sync Start		0AE8*	0AE8*	8	8514/A	25
H_SYNC_WID	Horizontal Sync Width		0EE8*	0EE8*	6	8514/A	25
V_TOTAL	Vertical Total		12E8*	12E8*	12	8514/A	26
V_DISP	Vertical Displayed		16E8*	16E8*	12	8514/A	26
V_SYNC_STRT	Vertical Sync Start		1AE8*	1AE8*	12	8514/A	27
V_SYNC_WID	Vertical Sync Width		1EE8*	1EE8*	6	8514/A	27
DISP_CNTL	Display Control		22E8*	22E8*	7	8514/A	28
SUBSYS_STAT	Subsystem Status		42E8*	N/A	16	8524/A	29
SUBSYS_CNTL	Subsystem Control		2EE8*	42E8*	12	8514/A	30
ROM_PAGE_SEL	ROM Page Select		46E8*	46E8*	3	8514/A	31
ADVFUNC_CNTL	Advanced Function Control		4AE8*	4AE8*	2	8514/A	32
EC0	Extended Configuration Register 0		52E8*	52E8*	0	CHIPS	33
EC1	Extended Configuration Register 1		56E8*	56E8*	0	CHIPS	33
EC2	Extended Configuration Register 2		5AE8*	5AE8*	13	CHIPS	34
EC3	Extended Configuration Register 3		5EE8*	5EE8*	12	CHIPS	36
CUR_Y	Current Y Position		82E8*	82E8*	12	8514/A	37
CUR_X	Current X Position		86E8*	86E8*	12	8514/A	38
DESTY_AXSTP	Destination Y Position/Axial Step Constant		8AE8*	8AE8*	13	8514/A	38
DESTX_DIASTP	Destination X Position/Diagonal Step Constant		8EE8*	8EE8*	13	8514/A	39
ERR_TERM	Error Term		92E8*	92E8*	16	8514/A	39
MAJ_AXIS_PCNT	Major Axis Pixel Count/Rectangle Width		96E8*	96E8*	11	8514/A	40
GP_STAT	Graphics Processor Status		9AE8*	N/A	10	8514/A	41
CMD	Command		N/A	9AE8*	16	8514/A	42
SHORT_STROKE	Short Stroke Vector Transfer		9EE8*	9EE8*	16	8514/A	44
BKGD_COLOR	Background Color		A2E8*	A2E8*	8	8514/A	45
FRGD_COLOR	Foreground Color		A6E8*	A6E8*	8	8514/A	45
WRT_MASK	Write Mask		AAE8*	AAE8*	8	8514/A	46
RD_MASK	Read Mask		AEE8*	AEE8*	8	8514/A	46
COLOR_CMP	Color Compare		B2E8*	B2E8*	8	8514/A	47
BKGD_MIX	Background Mix		B6E8*	B6E8*	8	8514/A	48
FRGD_MIX	Foreground Mix		BAE8*	BAE8*	8	8514/A	49
MULTIFUNC_CNTL	Multi-function Control Index (MFC)		N/A	BEE8*	4	8514/A	50
MIN_AXIS_PCNT	Minor Axis Pixel Count	MFC[0]	BEE8*	BEE8*	11	8514/A	50
SCISSORS_T	Top Scissors	MFC[1]	BEE8*	BEE8*	12	8514/A	51
SCISSORS_L	Left Scissors	MFC[2]	BEE8*	BEE8*	12	8514/A	51
SCISSORS_B	Bottom Scissors	MFC[3]	BEE8*	BEE8*	12	8514/A	52
SCISSORS_R	Right Scissors	MFC[4]	BEE8*	BEE8*	12	8514/A	52
MEM_CNTL	Memory Control	MFC[5]	BEE8*	BEE8*	5	8514/A	53
PATTERN_L	Fixed Pattern Low	MFC[8]	BEE8*	BEE8*	5	8514/A	54
PATTERN_H	Fixed Pattern High	MFC[9]	BEE8*	BEE8*	5	8514/A	54
PIX_CNTL	Pixel Control	MFC[A]	BEE8*	BEE8*	7	8514/A	55
PIX_TRANS	Pixel Data Transfer		E2E8*	E2E8*	8/16	8514/A	56

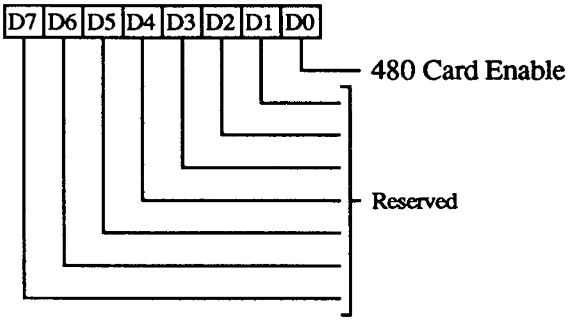
* Indicates 16-bit read/write access at port addresses xxx8/xxx9.

† When in VGA Pass-through mode, write accesses to 3C6–3C9h are mirrored in the 8514/A RAMDAC.

Register	I/O	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISP_STAT	02E8	0	0	0	0	0	0	0	0	0	0	0	0	0	TOG	VBK	SENS
H_TOTAL	26E8	0	0	0	0	0	0	0	H_T0TAL + 1								
DAC_MASK	02EA	X	X	X	X	X	X	X	X	PALETTE MASK							
DAC_R_INDEX	02EB	X	X	X	X	X	X	X	X	READ INDEX							
DAC_W_INDEX	02EC	X	X	X	X	X	X	X	X	WRITE INDEX							
DAC_DATA	02ED	X	X	X	X	X	X	X	X	RAMDAC DATA REGISTER							
H_DISP	06E8	0	0	0	0	0	0	0	0	HORIZONTAL DISPLAYED							
H_SYNC_START	0AE8	0	0	0	0	0	0	0	0	HORIZONTAL SYNC START							
H_SYNC_WID	0EE8	0	0	0	0	0	0	0	0	0	0	POL	H SYNC WIDTH				
V_TOTAL	12E8	0	0	0	0	VTB									VTADJ		
V_DISP	16E8	0	0	0	0	VTD									VDADJ		
V_SYNC_STRT	1AE8	0	0	0	0	VSB									VSADJ		
V_SYNC_WID	1EE8	0	0	0	0	0	0	0	0	0	0	POL	V SYNC WIDTH				
DISP_CNTL	22E8	0	0	0	0	0	0	0	0	0	DISPEN		INL	DBL	MEMCFG		ODD
SUBSYS_STAT	42E8R	CHIP ID				CHIP REV				8P	MON ID			GPI	IIO	PF	VBF
SUBSYS_CNTL	42E8W	GPCTRL		CHPTST		IGPI	IIIO	IPF	IVBF	0	0	0	0	RGPI	RIIO	RPF	RVBF
ROM_PAGE_SEL	46E8	0	0	0	0	0	0	0	0	0	0	0	VGA	VGA	ROM PAGE		
ADVFUNC_CNTL	4AE8	0	0	0	0	0	0	0	0	0	0	0	0	0	CLK	1	AF
EC2	5AE8	0	0	DAC	0	5PN	1M	BANKS		8BP	RPG	RSIZ	ROMBASE		1	1	1
EC3	5EE8	0	0	0	OVR	0	CLK2	CLK1	CLK0	ALE	AHE	ALS	AHS	MFC INDEX			
CUR_Y	82E8	0	0	0	0	0	Y–POSITION										
CUR_X	86E8	0	0	0	0	X–POSITION											
DESTY_AXSTP	8AE8	0	0	0	SGN	Y–DESTINATION / AXIAL STEP CONSTANT											
DESTX_DIASTP	8EE8	0	0	0	SGN	X–DESTINATION / DIAGONAL STEP CONSTANT											
ERR_TERM	92E8	R/W	R/W	R/W	INITIAL ERROR TERM												
MAJ_AXIS_PCNT	96E8	0	0	0	0	0	MAJOR AXIS PIXEL COUNT / RECTANGLE WIDTH										
GP_STAT	9AE8R	0	0	0	0	0	0	BSY	RDY	QUEUE STATE							
CMD	9AE8W	COMMAND			BSQ	0	0	16B	PCD	INCY	MAJ	INCX	DRW	TYP	LAST	AP	WRT
SHORT_STROKE	9EE8	ANGLE			DRW	LENGTH				ANGLE			DRW	LENGTH			
BKGD_COLOR	A2E8	0	0	0	0	0	0	0	0	BACKGROUND COLOR							
FRGD_COLOR	A6E8	0	0	0	0	0	0	0	0	FOREGROUND COLOR							
WRT_MASK	AAE8	0	0	0	0	0	0	0	0	WRITE MASK							
RD_MASK	AEE8	0	0	0	0	0	0	0	0	READ MASK							
COLOR_CMP	B2E8	0	0	0	0	0	0	0	0	COLOR COMPARE							
BKGD_MIX	B6E8	0	0	0	0	0	0	0	0	0	BSS		BACKMIX				
FRGD_MIX	BAE8	0	0	0	0	0	0	0	0	0	FSS		FOREMIX				
MULTIFUNC_CNTL	BEE8	INDEX				X	X	X	X	X	X	X	X	X	X	X	X
MIN_AXIS_PCNT	BEE8	0	0	0	0	0	MINOR AXIS PIXEL COUNT/ RECT HEIGHT										
SCISSORS_T	BEE8	0	0	0	1	TOP SCISSORS											
SCISSORS_L	BEE8	0	0	1	0	LEFT SCISSORS											
SCISSORS_B	BEE8	0	0	0	1	BOTTOM SCISSORS											
SCISSORS_R	BEE8	0	1	0	0	RIGHT SCISSORS											
MEM_CNTL	BEE8	0	1	0	1	0	0	0	0	0	0	0	SWP	VRTCFG		HORCFG	
PATTERN_L	BEE8	1	0	0	0	0	0	0	0	0	0	0	MASK LOW				
PATTERN_H	BEE8	1	0	0	1	0	0	0	0	0	0	0	MASK HIGH				
PIX_CNTL	BEE8	1	0	1	0	0	0	0	0	MIXSEL		COLCMPOP			PMODE		0
PIX_TRANS	E2E8	PIXEL/PLANE DATA								PIXEL/PLANE DATA							

SETUP CONTROL REGISTER (POS_2)

Read at I/O Address 102h
Write at I/O Address 102h
Byte Accessible only



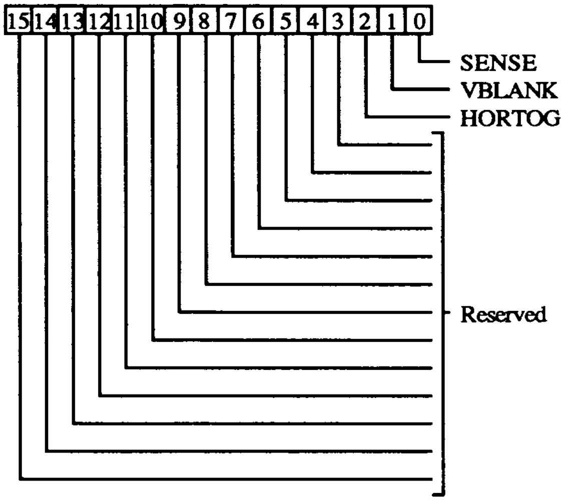
0 480 Card Enable

1 = Card enabled
0 = Card disabled (all registers and memory invisible to system except POS Setup registers).

7–1 Reserved = 0

DISPLAY STATUS REGISTER (DISP_STAT)

Read only at I/O Address 02E8h
Byte or Word Accessible



0 SENSE is the result of a wired-OR of comparators on each of the RGB video signals. This sense input may be used to determine if a color, monochrome or no monitor is connected to the analog connector. The RAMDAC expects to be driving a 75Ω matched impedance cable. The comparator reference voltage is generally chosen to be approximately half of the full scale DAC output. By programming the RAMDAC for various voltage outputs and patterns and then reading SENSE, the monitor type can be deduced.

1 VBLANK Vertical Blank State

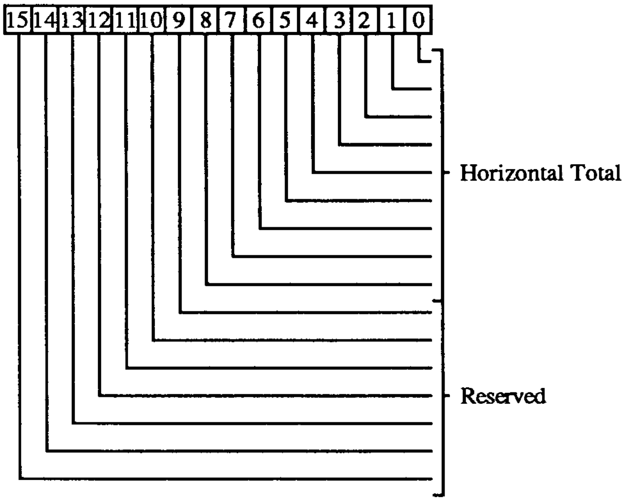
0 = Vertical Blank inactive
1 = Vertical Blank active

2 HORTOG Horizontal Toggle reads the state of a flip-flop which is clocked by HSYNC; the beginning of the sync pulse (which generally occurs after the end of displayed data) can be detected by polling this bit and waiting for a change of state.

15–3 Reserved = 0

HORIZONTAL TOTAL REGISTER (H_TOTAL)

Read at I/O Address 26E8h
Write at I/O Address 02E8h
Byte or Word Accessible



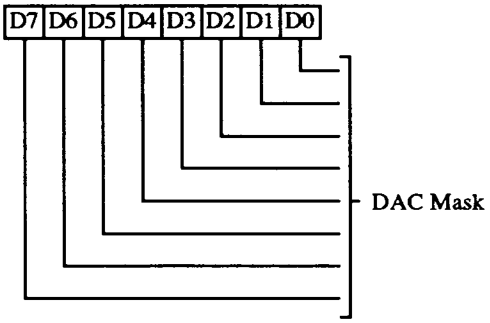
8–0 Horizontal Total defines the total horizontal scan line width including the display, blank, and sync times. All horizontal timings are measured in terms of double nuggets (8 or 10 pixels depending upon the state of MEM_CNTL[0]). By definition, H_TOTAL is measured from the start of the first displayed pixel on the scan line. Therefore, the total time per scan line is:

(H_TOTAL+1) double nugget periods.

15–9 Reserved = 0

DAC MASK REGISTER (DAC_MASK)

Read at I/O Address 02EAh
Write at I/O Address 02EAh
Write at I/O Address 03C6h (VGA Mode Only)
Byte Accessible only

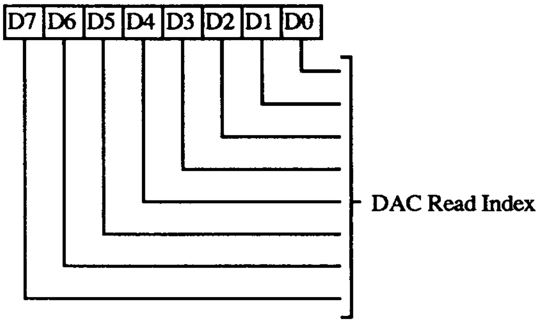


7–0 DAC Mask. This register is in the RAMDAC Color Palette. Pixel data bits are ANDed with the DAC Mask before going to the palette. This can be used to split the display buffer into multiple buffers by alternately enabling different planes.

The 8514/A palette registers may be written at their equivalent VGA I/O addresses when the 8514/A is in VGA pass-through mode. This allows the 8514/A palette to mirror the contents of the VGA palette hence displaying the correct colors when outputing VGA pass-through video. The 8514/A never responds to I/O read accesses to the VGA palette.

DAC READ INDEX REGISTER (DAC_R_INDEX)

Read at I/O Address 02EBh
Write at I/O Address 02EBh
Write at I/O Address 03C7h (VGA Mode Only)
Byte Accessible only



7–0 DAC Read Index. This register is in the RAMDAC Color Palette.

The DAC Read Index is used to point to the location in the RAM lookup table which will be read by the current sequence of I/O reads to the DAC_DATA register. To read the contents of a location in the RAM palette:

1. Write the RAM address for that location to the DAC Read Index register. The data in the RAMDAC at that address is read into the DAC_DATA register. The DAC Read Index register auto-increments to point to the next RAMDAC address location. Address location 0FFh wraps around to 0.

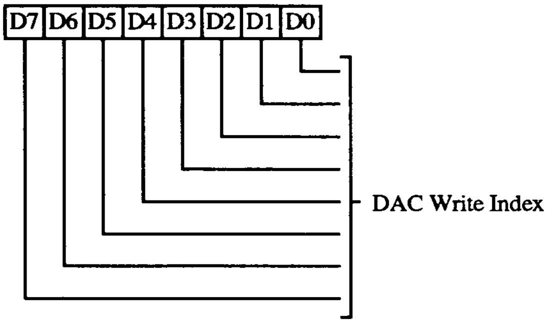
2. The palette data for that location may then be read by three consecutive accesses to the DAC_DATA register; one each for Red, Green, and Blue palette data (6 or 8 bits).

   Upon completion of the third read to the DAC_DATA register, the 18/24 bits of data for the next address location is available to be read from the DAC_DATA register.

The IBM 8514/A uses an 18-bit wide color palette (6 bits each for Red, Green, and Blue). The 82C480 supports both 6 and 8 bit RAMDACs.

DAC WRITE INDEX REGISTER (DAC_W_INDEX)

Read at I/O Address 02ECh
Write at I/O Address 02ECh
Write at I/O Address 03C8h (VGA Mode Only)
Byte Accessible only



7–0 DAC Write Index. This register is in the RAMDAC Color Palette.

The DAC Write Index is used to point to the location in the RAM lookup table which will be loaded following the completion of a triple write sequence to the DAC_DATA register. To update a location in the RAM palette:

1. Write the RAM address for that location to the DAC Write Index register.

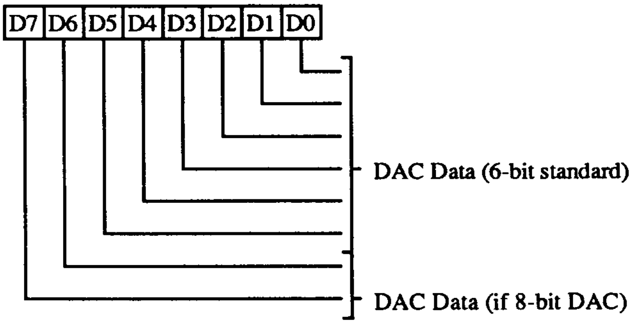
2. In three consecutive writes to the DAC_DATA register, output Red, Green, and Blue palette data (6 or 8 bits).

   Upon completion of the third write to the DAC_DATA register, the 18/24 bits of palette data is transferred to the RAMDAC lookup table. The DAC Write Index is then incremented to point to the next RAMDAC address location. Address location 0FFh wraps around to 0.

The IBM 8514/A uses an 18-bit wide color palette (6 bits each for Red, Green, and Blue). The 82C480 supports both 6 and 8 bit RAMDACs.

DAC DATA REGISTER (DAC_DATA)

Read at I/O Address 02EDh
Write at I/O Address 02EDh
Write at I/O Address 03C9h (VGA Mode Only)
Byte Accessible only



7–0 DAC Data. This register is in the RAMDAC Color Palette.

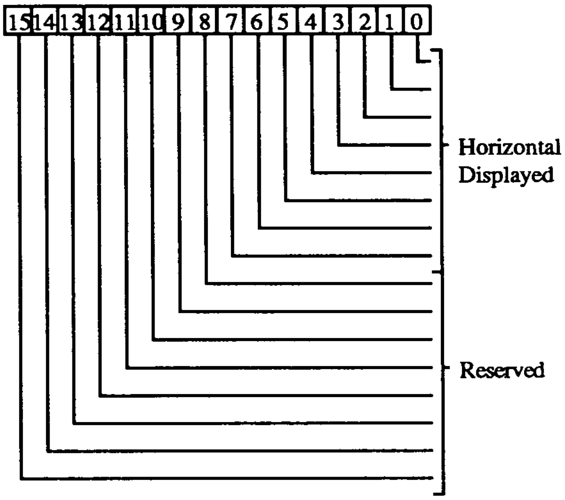
To write a new color for a pixel, first write its address to DAC_W_INDEX, then write the red, green, and blue values consecutively to DAC_DATA. After the blue value is written, all three color values are simultaneously transferred to the palette, and DAC_W_INDEX is incremented to point to the next palette word. This allows writing the whole palette with only one write to the DAC_W_INDEX register.

To read a color value, first write its address to DAC_R_INDEX; this causes that locations value to be copied into a temporary register (18 or 24-bit) and DAC_R_INDEX to be incremented. Then the red, green, and blue values (6 or 8-bit) may be read consecutively by reading the DAC_DATA register. As soon as the blue value is read, the next locations color data is transferred to the temporary register and DAC_R_INDEX is again incremented. This allows reading of the whole palette with only one write of the DAC_R_INDEX register.

The IBM 8514/A uses an 18-bit wide color palette (6 bits each for Red, Green, and Blue). The 82C480 supports both 6 and 8 bit RAMDACs.

HORIZONTAL DISPLAYED REGISTER (H_DISP)

Read at I/O Address 06E8h (Chips Extension)
Write at I/O Address 06E8h
Byte or Word Accessible



7–0 Horizontal Displayed defines both the number of displayed pixels per scan line and the start of the horizontal blanking. All horizontal timings are performed in terms of double nuggets. The start of the horizontal blanking signal occurs at:

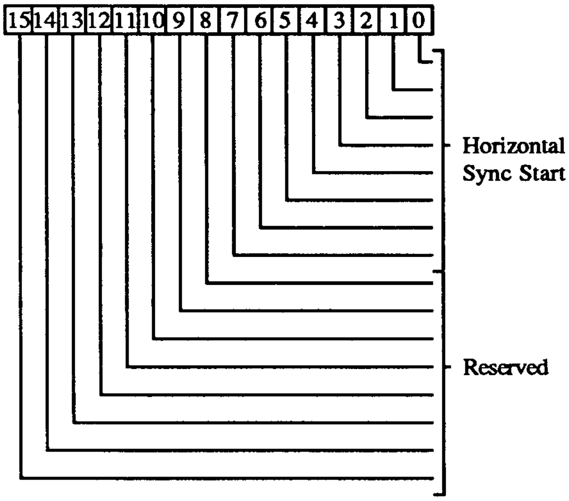
(H_DISP+1) double nugget periods

after the start of the scan line.

15–8 Reserved = 0

HORIZONTAL SYNC START REGISTER (H_SYNC_STRT)

Read at I/O Address 0AE8h (Chips Extension)
Write at I/O Address 0AE8h
Byte or Word Accessible



7–0 Horizontal Sync Start defines the timing delay of the horizontal sync signal from the start of a scan line. For HSYNC to occur, H_SYNC_STRT must be less than H_TOTAL. The horizontal sync pulse begins at:

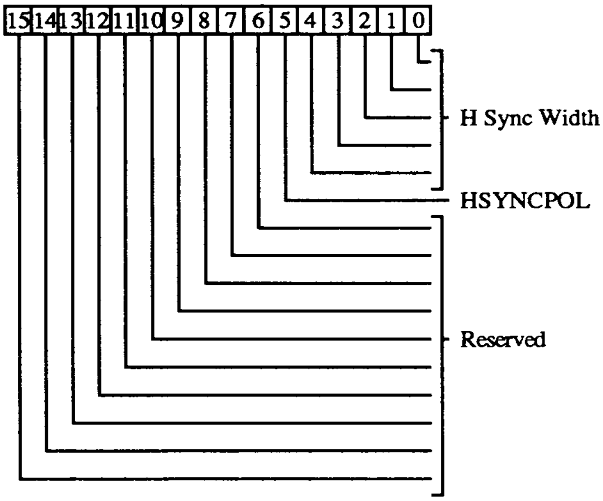
(H_SYNC_STRT+1) double nugget periods

after the beginning of a scan line.

15–8 Reserved = 0

HORIZONTAL SYNC WIDTH REGISTER (H_SYNC_WID)

Read at I/O Address 0EE8h (Chips Extension)
Write at I/O Address 0EE8h
Byte or Word Accessible



4–0 Horizontal Sync Width defines the width (in double nugget periods) of the horizontal sync pulse. The horizontal sync pulse is independent of Horizontal Total in that the end of a scan line does not terminate the HSYNC signal.

5 HSYNCPOL (Horizontal Sync Polarity)

1 = Negative (–)
0 = Positive (+)

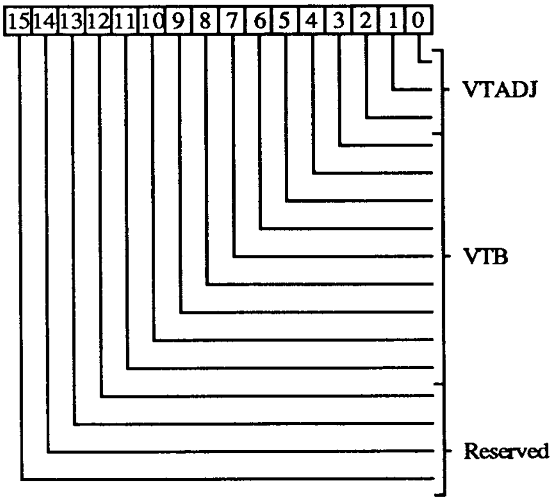
The horizontal and vertical sync polarities are used to indicate the horizontal and vertical frame times to some IBM monitors as described below:

Horizontal Sync Polarity	Vertical Sync Polarity	Horizontal Scan Freq (KHz)	Vertical Scan Freq (Hz)
–	–	31.47	59.94
–	+	31.47	70.08
+	–	31.47	70.08
+	+	35.52	43.48

15–6 Reserved = 0

VERTICAL TOTAL REGISTER (V_TOTAL)

Read at I/O Address 12E8h (Chips Extension)
Write at I/O Address 12E8h
Byte or Word Accessible



2–0 VTADJ = Vertical Total Adjust

11–3 VTB = Vertical Total Base

The Vertical Total value is calculated from VTB, VTADJ, and the Scan Modulus. The internal vertical counter register is incremented at the end of every scan line when in non-interlaced mode. When in interlaced mode, the vertical counter increments every half scan line (ie. when the horizontal counter reaches H_TOTAL÷2 and H_TOTAL). For true interlacing to occur, Vertical Total must be programmed to an odd number of half scan lines (ie. VTADJ is even).

Vertical Total = (Modulus * VTB) + VTADJ + 1

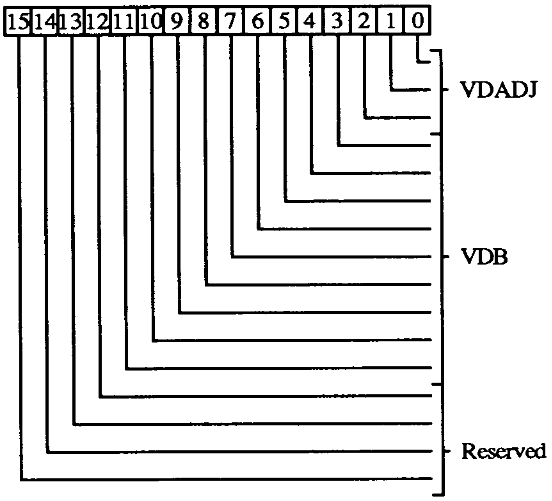
The Scan Modulus is calculated from the DBLSCAN and MEMCFG bits of the DISP_CNTL register as follows:

DBLSCAN	MEMCFG	Scan Modulus
0	0 0	2
0	0 1	4
0	1 0	6
0	1 1	8
1	0 0	4
1	0 1	8
1	1 0	12
1	1 1	16

15–12 Reserved = 0

VERTICAL DISPLAYED REGISTER (V_DISP)

Read at I/O Address 16E8h (Chips Extension)
Write at I/O Address 16E8h
Byte or Word Accessible



2–0 VDADJ = Vertical Displayed Adjust

11–3 VDB = Vertical Displayed Base

The Vertical Displayed value is calculated from VDB, VDADJ, and the Scan Modulus (opposite). The internal vertical counter is incremented at the end of every scan line when in non-interlaced mode; every half scan line (ie. when the horizontal counter reaches H_TOTAL÷2 and H_TOTAL) when in interlaced mode. The Vertical Displayed value also determines the starting location for the Vertical Blank signal. The Vertical Blank signal is synchronized to the end of a scan line (full H_TOTAL).

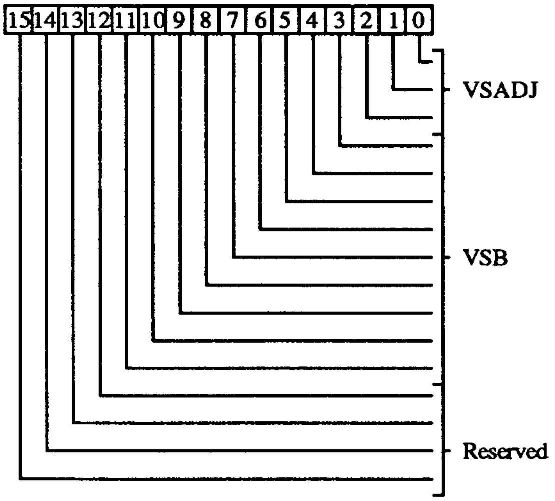
Vertical Disp = (Modulus * VDB) + VDADJ + 1

Since Vertical Total must be odd when in interlaced mode, scan lines have an alternating odd/even relationship with the half scan line count in alternate frames. (In even frames, the vertical counter begins counting at the start of a full scan line and ends on a half scan line. In the odd frame which follows, the vertical counter begins on a half scan line and ends on a full scan line.) Because Vertical Blank is synchronized to full H_TOTAL, the Vertical Blank pulse for even frames is delayed by H_TOTAL÷2 and is shorter by this amount in odd frames.

15–12 Reserved = 0

VERTICAL SYNC START REGISTER (V_SYNC_STRT)

Read at I/O Address 1AE8h (Chips Extension)
Write at I/O Address 1AE8h
Byte or Word Accessible



2–0 VSADJ = Vertical Sync Start Adjust

11–3 VSB = Vertical Sync Start Base

The Vertical Sync Start value is calculated from VSB, VSADJ, and the Scan Modulus. The internal vertical counter register is incremented at the end of every scan line when in non-interlaced mode. When in interlaced mode, the vertical counter increments every half scan line (i.e. when the horizontal counter reaches H_TOTAL÷2 and H_TOTAL). The Vertical Sync signal is not synchronized to the end of a scan line (full H_TOTAL) as is the Vertical Blank.

VSYNC START = (VSB * Modulus) + VSADJ + 1

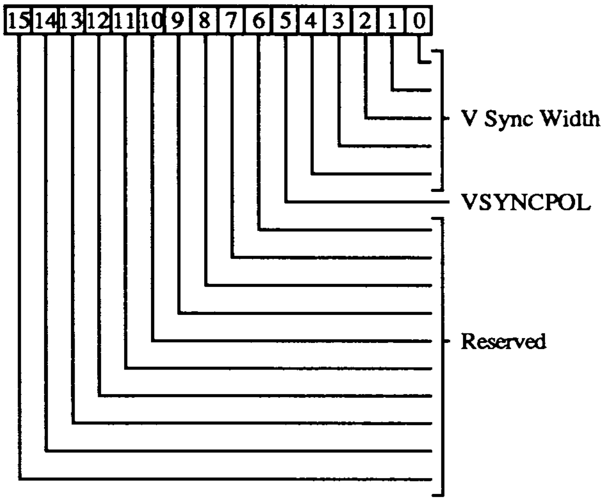
The Scan Modulus is calculated from the DBLSCAN and MEMCFG bits of the DISP_CNTL register as follows:

DBLSCAN	MEMCFG	Scan Modulus
0	0 0	2
0	0 1	4
0	1 0	6
0	1 1	8
1	0 0	4
1	0 1	8
1	1 0	12
1	1 1	16

15–12 Reserved = 0

VERTICAL SYNC WIDTH REGISTER (V_SYNC_WID)

Read at I/O Address 1EE8h (Chips Extension)
Write at I/O Address 1EE8h
Byte or Word Accessible



4–0 Vertical Sync Width defines the width of the vertical sync pulse in scan lines. The vertical sync pulse is independent of Vertical Total so that when the vertical counter reaches Vertical Total, the VSYNC signal is not terminated. When in interlaced mode, the pulse width is counted in half scan lines.

5 VSYNCPOL (Vertical Sync Polarity)

1 = Negative (–)
0 = Positive (+)

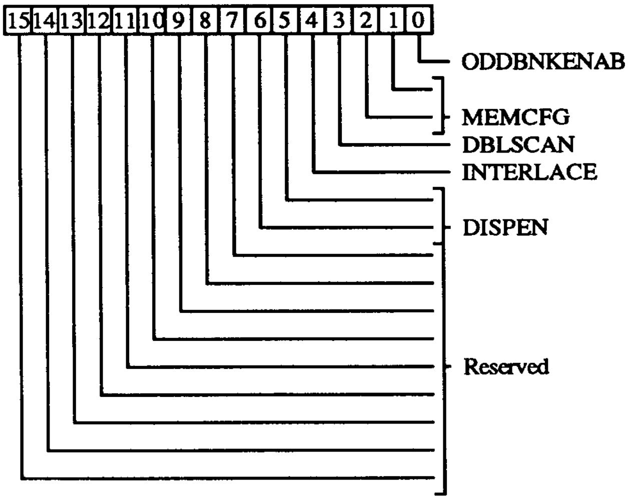
The horizontal and vertical sync polarities are used to indicate the horizontal and vertical frame times to some IBM monitors as described below:

Horizontal Sync Polarity	Vertical Sync Polarity	Horizontal Scan Freq (KHz)	Vertical Scan Freq (Hz)
–	–	31.47	59.94
–	+	31.47	70.08
+	–	31.47	70.08
+	+	35.52	43.48

15–6 Reserved = 0

DISPLAY CONTROL REGISTER (DISP_CNTL)

Read at I/O Address 22E8h (Chips Extension)
Write at I/O Address 22E8h
Byte or Word Accessible



0 ODDBNKENAB (Odd Bank Enable). This bit is not used internally by the 480.

0 = No Change
1 = Horizontally Interleave banks (normal 8514/A mode).

2–1 MEMCFG (Memory Configuration)

2	1	Function
0	0	NCLK (PS8 Mode)
0	1	NCLK÷ 2 (Normal 8514/A Mode)
1	0	NCLK÷ 3
1	1	NCLK÷ 4

3 DBLSCAN (Double Scan).

0 = Single scan
1 = Double scanning enabled

4 INTERLACE

0 = Non-interlace
1 = Interlace

6–5 DISPEN (Display Enable)

6	5	Function
0	0	No effect
0	1	Enable Hsync, Vsync, Blank, data transfer cycles, and refresh cycles
1	x	Disable Hsync, Vsync, Blank, data transfer cycles, and refresh cycles

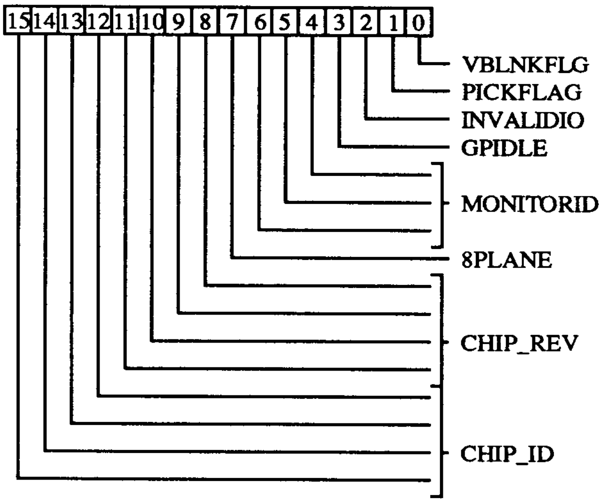
Bits 6:5 control an R-S latch. The output of this latch is read back in bit 5; bit 6=0.

15–7 Reserved = 0

NOTE: INTERLACE, DBLSCAN, and MEMCFG bits are duplicated in each of the alternate video register sets.

SUBSYSTEM STATUS REGISTER (SUBSYS_STAT)

Read at I/O Address 42E8h
Byte or Word Accessible



3–0 Interrupt requests. These bits show the state of internal interrupt requests; these will only activate the IRQ pin if the corresponding Interrupt Enable bit in the SUBSYS_CNTL register is set. Interrupts may only be reset by writing a 1 to the corresponding Interrupt Clear bit in the SUBSYS_CNTL register. The interrupt conditions supported are:

0 VBLNKFLG (Vertical Blank Flag) - Vertical blank signal has become active

1 PICKFLAG - The current XY position went inside the scissor while executing a line, outline, SSV, bitblt or rectangle command.

2 INVALIDIO - The host attempted to write to the queue when no words were available or read from PIX_TRANS when no data was ready.

3 GPIDLE - The graphics drawing engine is idle.

6–4 MONITORID - This field contains the monitor ID bits from pins MS2:0. These bits are driven by some monitors to indicate monitor type. The following monitor IDs are returned by IBM and compatible monitors:

Monitor ID			IBM Monitor Type
0	0	0	Reserved
0	0	1	8507 (1024x768) Monochrome
0	1	0	8514 (1024x768) Color
0	1	1	Reserved
1	0	0	Reserved
1	0	1	8503 (640x480) Monochrome
1	1	0	8512/8513 (640x480) Color
1	1	1	Reserved

7 8PLANE - 8 bit planes
Latched on reset from P4D7, 8PLANE emulates the function of the jumper on the IBM 8514/A board.

0 = 4 planes
1 = 8 planes

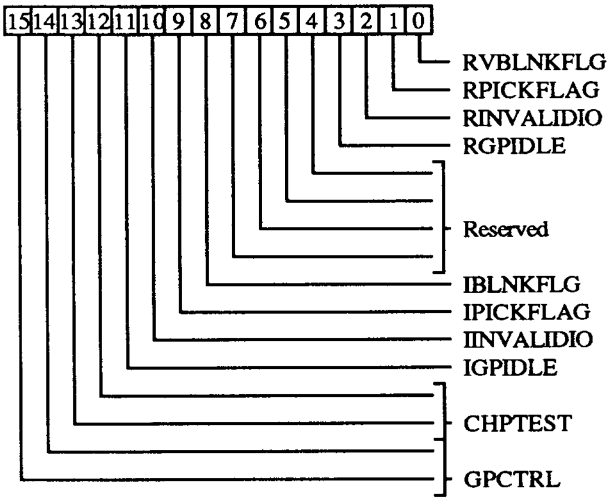
11–8 CHIP_REV - Revision number of the silicon. This number, in combination with the CHIP_ID will allow programmers to determine which advanced capabilities are available on this 8514/A.

15–12 CHIP_ID - This field contains the CHIPS ID for this 8514/A implementation.

0 = Chips and Technologies 82C480

SUBSYSTEM CONTROL REGISTER (SUBSYS_CNTL)

Read at I/O Address 2EE8h (Chips Extension)
Write at I/O Address 42E8h
Byte or Word Accessible



3–0 Interrupt Reset. These bits are used to reset active interrupt flags. They may be reset individually or in combination.

0 = Don't reset flag
1 = Reset flag

The reset bits are as follows:

0 RVBLNKFLG Reset Vertical Blank Flag
1 RPICKFLAG Reset PICK Flag
2 RINVALIDIO Reset Queue overflow / Data underflow flag
3 RGPIDLE Reset Graphics Engine Idle flag

Note:Bits 3:0 always read back = 0.

7–4 Reserved = 0

11–8 Interrupt Enables. These bits are used to permit a particular interrupt condition to generate an external IRQ.

0 = Disable interrupts for this condition
1 = Enable interrupts for this condition

8 IBLNKFLG Enable Vertical Blank interrupt
9 IPICKFLAG Enable PICK interrupt
10 IINVALIDIO Enable Queue Overflow / Data Underflow interrupt
11 IGPIDLE Enable Graphics Engine Idle interrupt

13–12 CHPTEST Chip test bits used by IBM for testing their two chip set. Program to 01 to enable normal function.

On read back, bit D13 = 1 if 13:12 were programmed to 1x; otherwise = 0. On reset, initialized to 0 (Normal Operation). Bit D12 always reads back as 0.

15–14 GPCTRL Graphics Processor Control

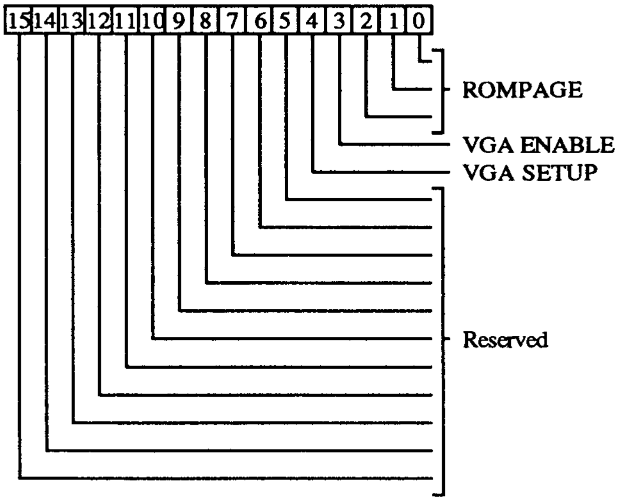
15	14	Effect on 82C480
0	0	No effect
0	1	Enable Chip
1	0	Reset Chip. Stop all VRAM cycles, flush queue, clear GPBUSY and Queue Status flags. Don't reset counters or tristate pins.
1	1	Reset Chip. Stop all VRAM cycles, flush queue, clear GPBUSY and Queue Status flags. Don't reset counters or tristate pins. (Same as 1 0).

Bits 15:14 control an R-S type latch. The status of this software reset latch is read back in bit D15; bit D14 = 0.

On reset bit D15 = 0 (enabled).

ROM PAGE SELECT REGISTER (ROM_PAGE_SEL)

Read at I/O Address 46E8h (Chips Extension)
Write at I/O Address 46E8h
Byte or Word Accessible



2–0 ROMPAGE. ROMPAGE allows mapping one of 8 different 4K pages of on-board ROM to the same 4K memory in address space.

All of the standard IBM 8514/A address ranges except CA000h–CA7FFh conflict with most extended VGA BIOS. To prevent a conflict from occurring, we suggest the ROM (if installed) be relocated to another address range using the ROM relocation bits in Extended Configuration Register EC3. EC3 also contains bits to disable ROM paging and to select the size of the ROM address space.

3 VGA ENABLE This bit is not physically present on the 82C480, but is standard on ISA VGA cards. It is included here for completeness.

0 = Disable VGA
1 = Enable VGA

4 VGA SETUP This bit is not physically present on an 8514/A card but is standard on ISA VGA cards. It is included here for completeness.

0 = Normal operation
1 = Setup mode

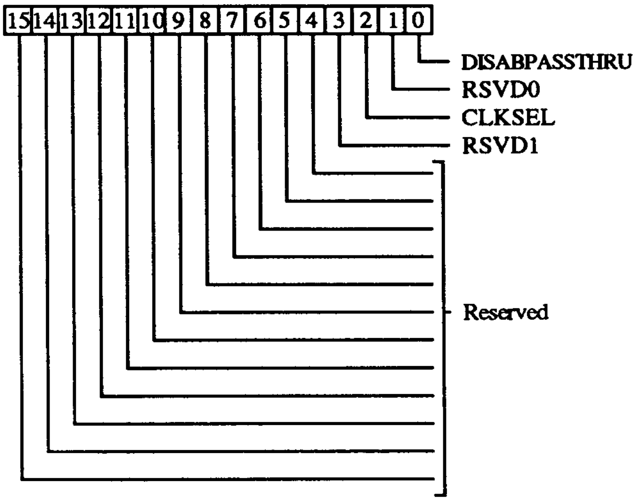
15–5 Reserved = 0

ISA Bus Address	MCA Bus Address	ROMPAGE	32K x 8 ROM	
(20-bit)	(24-bit)	2 1 0	Address	Comments
C6800–C6FFF	0C6800–0C6FFF	x x x	7800–7FFF	Fixed ROM
C7000–C7FFF	0C7000–0C7FFF	0 0 0	0000–0FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	0 0 1	1000–1FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	0 1 0	2000–2FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	0 1 1	3000–3FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	1 0 0	4000–4FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	1 0 1	5000–5FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	1 1 0	6000–6FFF	Paged ROM
C7000–C7FFF	0C7000–0C7FFF	1 1 1	7000–7FFF	Paged ROM
CA000–CA7FF	0CA000–0CA7FF	x x x	7000–77FF	Fixed ROM

IBM 8514/A Standard ROM Address Mapping

ADVANCED FUNCTION CONTROL REGISTER (ADVFUNC_CNTL)

Read at I/O Address 4AE8h (Chips Extension)
Write at I/O Address 4AE8h
Byte or Word Accessible



0 DISABPASSTHRU This bit controls the VGA pass-through feature of the 8514/A.

- 0 = VGA video is passed through the 8514/A RAMDAC to the display.
- 1 = Advanced Function. 8514/A video is passed through the RAMDAC to the display.

1 RSDV0 Reserved bit 0 = 1

2 CLKSEL (Clock Select) The IBM 8514/A supports only two clock frequencies; 0=25.175MHz (640x480), and 1=44.900MHz (1024x768 interlaced). The 480 supports 8 clock frequencies through its extended clock select bits in register EC3. This bit also controls the selection of the alternate video register sets if they are enabled. Alternate Video Register sets are an 82C480 extension of the 8514/A standard. They are used to support display timing parameters other than the IBM standard (eg. 1024x768 non-interlaced).

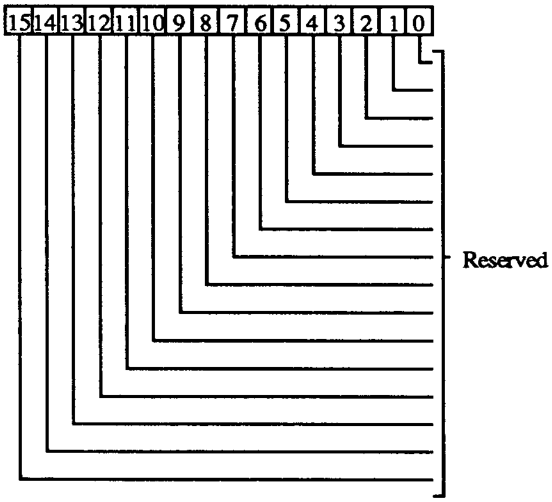
3 RSDV1 Reserved bit 1 = 0

15–4 Reserved = 0

WARNING: *All write accesses to this register will reset the Alternate Register Set OVERRIDE bit (EC3[12]). This means that a non-standard resolution driver must rewrite register EC3 each time a write to ADVFUNC_CNTL is performed.*

EXTENDED CONFIGURATION REGISTER 0 (EC0)

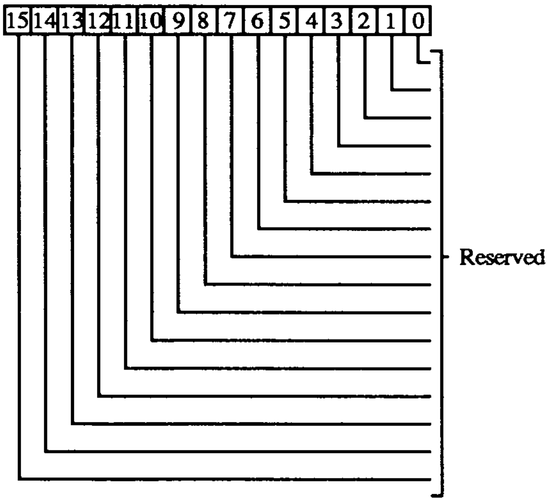
Read at I/O Address 52E8h (Chips Extension)
Write at I/O Address 52E8h (Chips Extension)
Byte or Word Accessible



15–0 Reserved for future use by CHIPS.

EXTENDED CONFIGURATION REGISTER 1 (EC1)

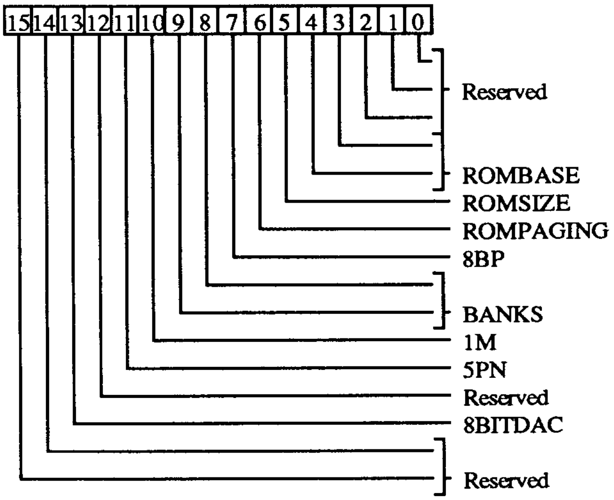
Read at I/O Address 56E8h (Chips Extension)
Write at I/O Address 56E8h (Chips Extension)
Byte or Word Accessible



15–0 Reserved for future use by CHIPS.

EXTENDED CONFIGURATION REGISTER 2 (EC2)

Read at I/O Address 5AE8h (Chips Extension)
Write at I/O Address 5AE8h (Chips Extension)
Byte or Word Accessible



2–0 Reserved = 1. These bits are latched on reset from P4D2:0 and as yet have no defined usage. They each have internal pull–up resistors so they default to 1.

4–3 ROMBASE ROM Base defines the start of the 8514/A ROM address space in system memory (default = 11). These bits are latched on reset from P4D4:3. ROM address mapping is also dependant on ROMSIZE (below) and the type of system interface.

EC2		8K		32K	
[4]	[3]	MCA	ISA	MCA	ISA
0	0	0C8000	C6000	0D0000	D0000
0	1	0D8000	D8000	0D8000	D8000
1	0	0C0000	C0000	0C0000	C0000
1	1	0C6000	C8000	0C8000	C8000

ROM page 0 always maps to the first 4K of address space.

NOTE: The VGA ROM takes up 32K bytes at C0000–C7FFFh which conflicts with the space used by the MCA bus 8514/A (C6000–C7FFFh). Also, CA000h as used by IBM conflicts with other cards in typical ISA bus systems. For this reason, it is recommended that the ROM space in 480-based systems be configured for the default addresses: C6000–C7FFFh in MCA and C8000–C9FFFh in the ISA bus. For specific ROM mapping tables consult the tables on the following page.

5 ROMSIZE Indicates the size of the address space allocated in memory (starting at the base address indicated by ROMBASE above).

0 = 32K ROM address space
1 = 8K ROM address space (default)

This bit is latched on reset from P4D5.

If a 32K physical ROM is implemented in an 8K address space, then the ROMPG2:0 pins should be connected to ROM address inputs A14:12. The upper 4K of the ROM space may then be paged from any one of the eight 4K pages in the ROM. This is done via the ROM_PAGE_SEL register. If an 8K physical ROM is connected, ROMSIZE should be selected as 8K and ROMPG2:0 do not need to be connected. These bits may then be used as general purpose outputs or may be selected by the ROM PAGING bit as transparent inputs for the MONITOR_ID bits.

6 ROMPAGING This bit is latched on reset from P4D6.

0 = ROMPG pins are inputs
1 = ROMPG pins are outputs

If ROMPAGING = 1, after latching the MONITOR_ID data on reset, the ROMPG pins then become outputs. If ROMSIZE is also latched low on reset, then ROMPG pins always output the contents of ROM_PG_SEL[2:0]. If ROMSIZE = 1, then ROMPG pins output either address or ROM_PG_SEL[2:0] as demonstrated in the paging table on the following page.

If ROMPAGING = 0, then the ROMPG pins become transparent inputs readable at SUBSYS_STAT[6:4]. This allows the detection of a monitor which was connected after power was switched on.

7 8BP (8 Bit Planes). This bit is a copy of the read-only 8PLANE bit in SUBSYS_STAT[7]. It is latched from P4D7 on reset. Configuration drivers written specifically for the 82C480 may write this bit, eliminating the need for a jumper.

0 = 4-bit planes
1 = 8-bit planes (default)

9–8 BANKS Indicates the number of VRAM banks connected to the 82C480.

00 = 1 bank
01 = 2 banks
10 = 3 banks
11 = 4 banks (default)

These bits are latched from pins CAS3/ and CAS2/ on reset.

10 1MB Indicates type of VRAMs used:

0 = 64Kx4
1 = 256Kx4 (default)

This bit is latched from MA8 on reset.

11 5PN 5-Pixel Nuggets. This bit is latched from WE4/ on reset. It determines whether the VRAM bank configuration is 4 or 5 pixels deep. It is necessary to program MEM_CNTL[0] in order to set the 82C480 into 5-pixel nugget mode.

12 Reserved = 0

13 8BITDAC 8-Bit DAC Control. Latched from 8BITDAC on reset.

0 = 6-bit
1 = 8-bit

15–14 Reserved = 0

Note: The 82C480 mapping table shown below demonstrates the address space mapping for a default 480 MCA configuration. Paging in the C7000–C7FFFh range is chosen by n=ROM_PAGE_SEL[2:0]. The value of 'n' is determined by the ROMPAGING bits. '0L' refers to the lower 4KB of the 8KB ROM page 0; '0H' refers to the upper 4KB of that page.

System Address	IBM MCA 8514/A ROM Address	480 MCA 8K ROM Address	ROM Page
C6000–C67FF	N/A	0000–07FF	0L
C6800–C6FFF	7800–7FFF	0800–0FFF	7H/0H
C7000–C7FFF	n000–nFFF	n000–nFFF	n
CA000–CA7FF	7000–77FF	N/A	7L

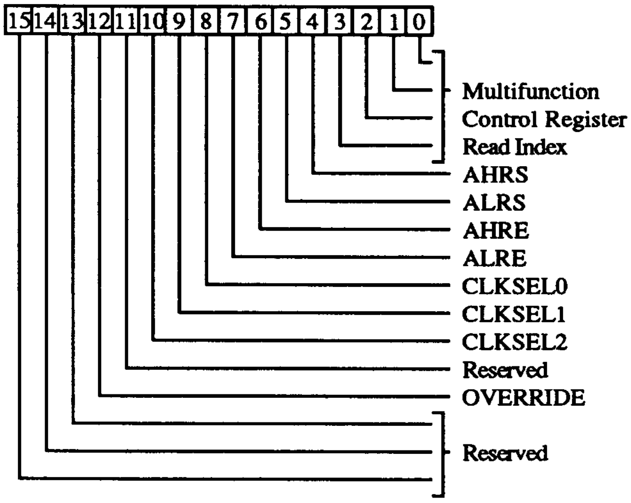
8514/A and 480 ROM Map
(MCA Bus)

System Address	IBM ISA VGA ROM Address	Typical ISA VGA ROM Address	ROM Page
C0000–C5FFF	0–5FFF	0000–5FFF	0–5
C6000–C67FF	N/A	6000–67FF	6L
C6800–C6FFF	7800–7FFF	6800–6FFF	7H/6H
C7000–C7FFF	n000–nFFF	7000–7FFF	n / 7
CA000–CA7FF	7000–77FF	N/A	7L

VGA ROM Map
(ISA Bus)

EXTENDED CONFIGURATION REGISTER 3 (EC3)

Read at I/O Address 5EE8h (Chips Extension)
Write at I/O Address 5EE8h (Chips Extension)
Byte or Word Accessible



3–0 Multifunction Control Register Read Index. These bits indicate which of the Multifunction Control [MFC] index registers will be read by a read access to BEE8h. When a read is performed at BEE8h, the index bits [15:12] are undefined.

4 AHRS Alternate High Register Select. This bit controls write access to the Alternate High group of Video Timing Parameter Registers.

0 = Not selected (default)
1 = Selected

This bit is set to initialize the High-Resolution video parameters (1024x768). This would generally only be done once following power-up under the control of the BIOS or a device driver. It may be used to customize the video timing for a specific country or monitor type (eg. non-interlaced vs. interlaced).

Warning: This bit disables write access to the normal video registers. Do not enable select bits AHRS and ALRS simultaneously. Do not leave this bit set after programming the desired video timing registers.

5 ALRS Alternate Low Register Select. This bit controls write access to the Alternate Low group of Video Timing Parameter Registers.

0 = Not selected (default)
1 = Selected

This bit is set to initialize the Low-Resolution video parameters (640x480). This would generally only be done once following power-up under the control of the BIOS or device driver. It may be used to customize the video timing for a specific country or monitor type (eg. 72Hz. refresh).

Warning: This bit disables write access to the normal video registers. Do not enable select bits AHRS and ALRS simultaneously. Do not leave this bit set after programming the desired video timing registers.

6 AHRE Alternate High Register Enable. This bit enables the alternate High-resolution video timing register set. When this bit is set, a write to the ADVFUNC_CNTL register with CLKSEL0=1, will select the alternate high registers for use when generating video.

0 = Use normal video registers (default)
1 = Use Alternate High video registers

7 ALRE Alternate Low Register Enable. This bit enables the alternate Low-resolution video timing register set. When this bit is set, a write to the ADVFUNC_CNTL register with CLKSEL0=0, will select the alternate low registers for use when generating video.

0 = Use normal video registers (default)
1 = Use Alternate Low video registers

10–8 CLKSEL2:0 Clock Select 2–0. There are three sets of Clock Select bits maintained internally by the 480. One set each for Alternate High, Alternate Low, and normal video register sets. When one of the Alternate Register Select bits is set, then the CLKSEL bits corresponding to that alternate set is visible (read/write) at this location. The default on reset is the normal set. The CLKSEL0 bit of the normal video register

Warning: *All writes to this register should be 8-bit. Due to the nature of the control bits in the lower byte, simultaneous writes to the alternate CLKSEL bits would be indeterminate.*

set is always read/write accessible in the ADVFUNC_CNTL register.

11 Reserved = 0

12 OVERRIDE (default = 1)

0 = Do not override AHRE,ALRE
1 = Force normal video register set

This bit allows an 82C480 custom driver to force the 480 to use the normal video register set. That is, it overrides the Alternate Register Enable bits since it is using a video mode other than 640x480 or 1024x768.

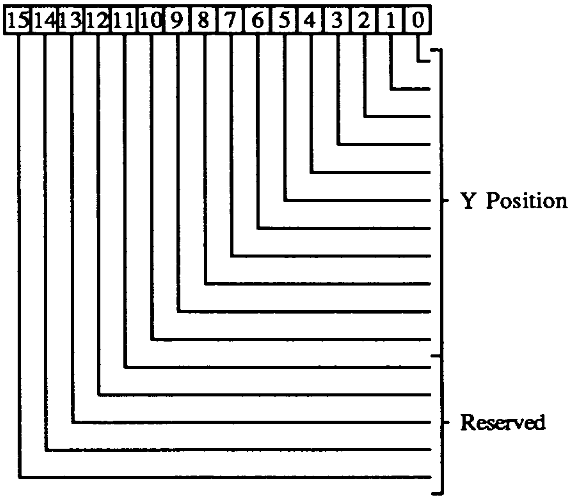
This bit is reset on any write to the ADVFUNC_CNTL register. This is necessary to avoid improper recovery when a non-standard exit from an application driver occurs. The 480 must always force an unsuspecting program (not a CHIPS 480 driver) to use the Alternate Video Register Set if one is enabled.

The implications of this reset is that if a high resolution device driver which uses OVERRIDE should need to write ADVFUNC_CNTL, then it must immediately write EC3 to again override the Alternate Register Enable bits.

15–13 Reserved = 0

CURRENT Y POSITION REGISTER (CUR_Y)

Read at I/O Address 82E8h
Write at I/O Address 82E8h
Byte or Word Accessible

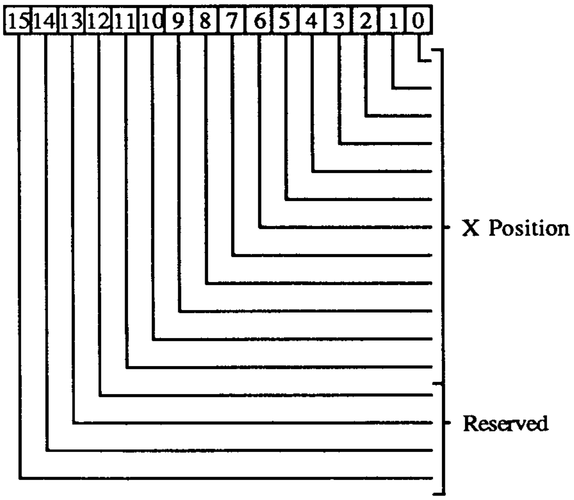


11–0 Y Position = Y coordinate of current position (in pixels). The XY position is relative to the top left corner of the display. Each scan line displays X-coordinates 0 through XMAX – 1. Each screen displays Y-coordinates 0 through YMAX – 1.

15–12 Reserved = 0

CURRENT X POSITION REGISTER (CUR_X)

Read at I/O Address 86E8h
Write at I/O Address 86E8h
Byte or Word Accessible



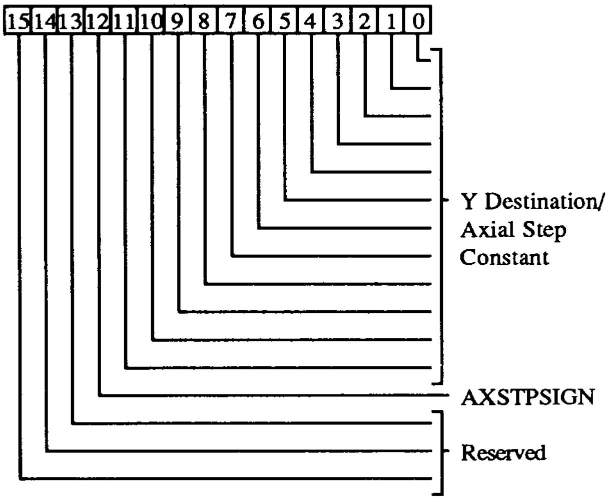
11–0 X Position = X coordinate of current position (in pixels). The XY position is relative to the top left comer of the display. Each scan line displays X-coordinates 0 through XMAX – 1. Each screen displays Y-coordinates 0 through YMAX – 1.

Warning: In 5PN mode, bits 11, 1 and 0 are remainder bits; 10:2 are the current X position modulus 5.

15–12 Reserved = 0

DESTINATION Y POSITION REGISTER AXIAL STEP CONSTANT REGISTER (DESTY_AXSTP)

Read at I/O Address 8AE8h (Chips Extension)
Write at I/O Address 8AE8h
Byte or Word Accessible



11–0 Y-Destination. During BITBLT operations this register is programmed to the destination Y address (measured in pixels).

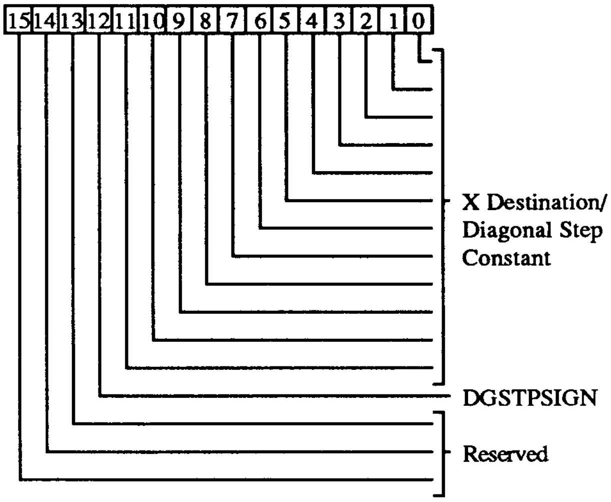
11–0 Axial Step Constant. During line drawing, this register is programmed as the Bresenham constant 2*dminor (dminor is the length of the line projected onto the minor or dependant axis).

12 AXSTPSIGN Sign flag for Axial Step Constant (should be set to 0 during BITBLT operations).

15–13 Reserved = 0

DESTINATION X POSITION REGISTER DIAGONAL STEP CONSTANT REGISTER (DESTX_DIASTP)

Read at I/O Address 8EE8h (Chips Extension)
Write at I/O Address 8EE8h
Byte or Word Accessible



11–0 X-Destination. During BITBLT operations this register is programmed to the destination X address (measured in pixels).

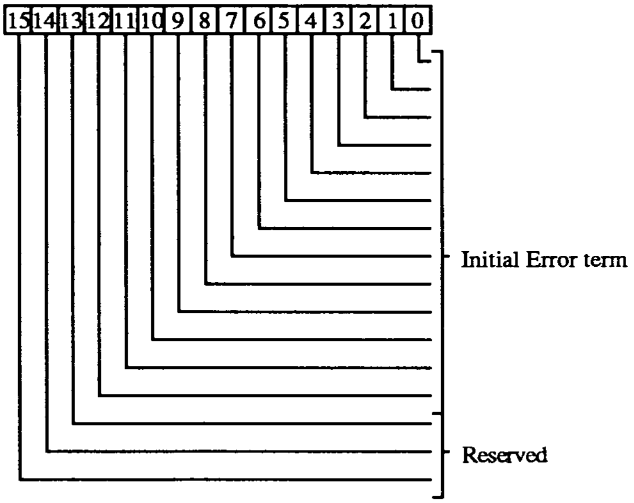
11–0 Diagonal Step Constant. During line drawing, this register is programmed as the Bresenham constant 2*dminor – 2*dmajor (dminor is the length of the line projected onto the minor or dependant axis; dmajor is the length of the line projected onto the major or independent axis).

12 DGSTPSIGN Sign flag for Diagonal Step Constant (should be set to 0 for BITBLT operations).

15–13 Reserved = 0

ERROR TERM REGISTER (ERR_TERM)

Read at I/O Address 92E8h
Write at I/O Address 92E8h
Byte or Word Accessible

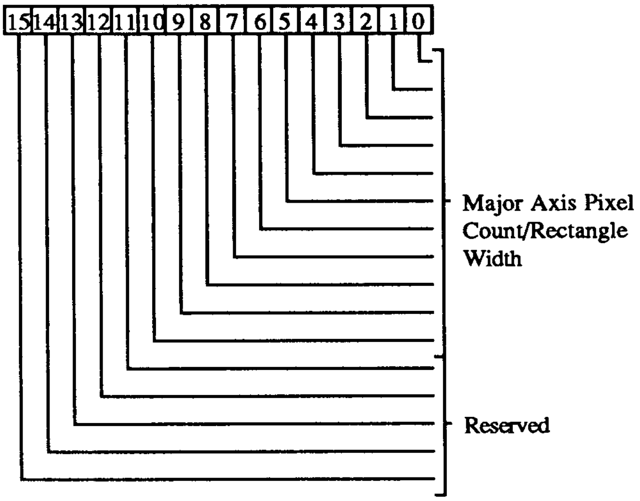


12–0 Initial Error Term. This register is programmed to the Bresenham initial error term before a line drawing command is issued. Its initial value is set to 2*dminor – dmajor (dminor is the length of the line projected onto the minor or dependant axis; dmajor is the length of the line projected onto the major or independent axis). During the line draw, this register holds the current state of the error term. When bit 12=1, it causes the Bresenham logic to increment the dependant axis (make a diagonal step).

15–13 Reserved (R/W)
These bits are read/write but do not participate in the calculation of the error term.

MAJOR AXIS PIXEL COUNT REGISTER RECTANGLE WIDTH REGISTER (MAJ_AXIS_PCNT)

Read at I/O Address 96E8h (Chips Extension)
Write at I/O Address 96E8h
Byte or Word Accessible



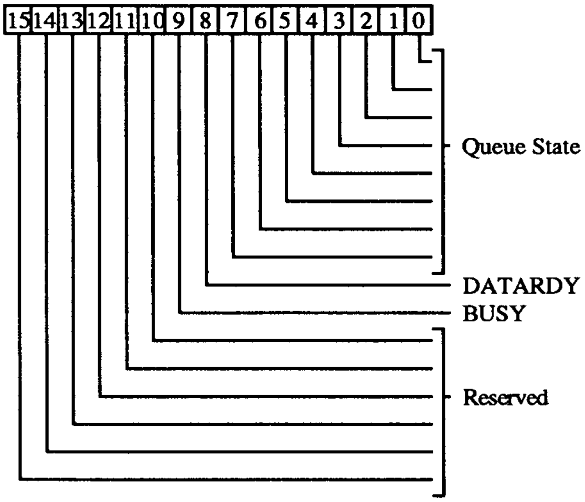
10–0 Major Axis Pixel Count holds the constant term dmajor for Bresenham line drawing (dmajor is the length of the projection of the line to be drawn along the independent axis). This must be a positive number.

10–0 Rectangle Width holds the width of the rectangle (in pixels) for BITBLT and rectangle drawing commands.

15–11 Reserved = 0

GRAPHICS PROCESSOR STATUS REGISTER (GP_STAT)

Read at I/O Address 9AE8h
Byte or Word Accessible



7–0 Queue State indicates the number of positions available in the queue. The standard 8514/A command queue is eight bytes deep. Queue State is nicknamed the thermometer register. It shows the number of bytes available encoded linearly. Queue State = 0 if the queue is empty; this does not imply that the 82C480 is idle-it may have consumed all of the data in the command queue but still may be executing a drawing command. Always test BUSY to determine if the drawing engine is idle.

Queue State 76543210		
00000000	=	8 words available-queue empty
00000001	=	7 words available
00000011	=	6 words available
00000111	=	5 words available
00001111	=	4 words available
00011111	=	3 words available
00111111	=	2 words available
01111111	=	1 word available
11111111	=	0 words available-queue full

8 DATARDY indicates availability of variable data from the PIX_TRANS register.

0 = No new data ready
1 = Data is ready to be read from the PIX_TRANS register. This will occur if a drawing command is executed with WRTDATA=0 and PCDATA=1. It indicates that the next pixel has been read from the display buffer and may be read from the PIX_TRANS register.

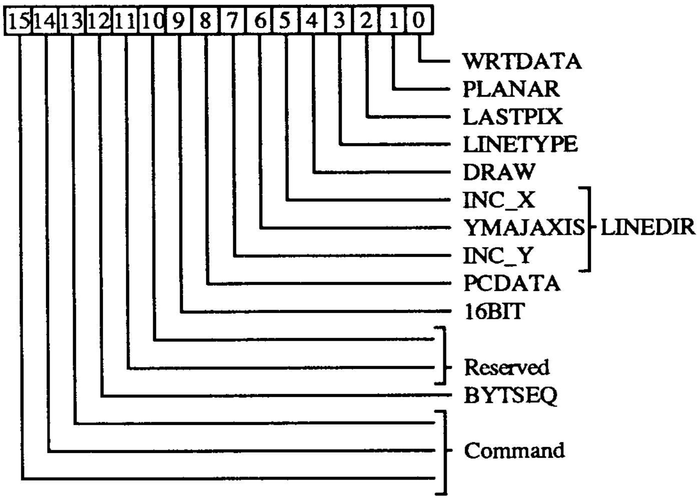
9 GPBUSY (Graphics Processor Busy)

0 = Not Busy
1 = Busy (currently executing a drawing command).

15–10 Reserved = 0

COMMAND REGISTER (CMD)

Write at I/O Address 9AE8h
Byte or Word Accessible



0 WRTDATA = Pixel write operation

0 = Read operation (line, SSV, or rectangle commands). Commands execute normally except that no writes to memory are performed. This can be used with rectangle commands to copy rectangular blocks of the display bitmap into system memory.
1 = Write operation. VRAM writes are enabled for drawing command.

1 PLANAR defines the orientation of the display bitmap for most drawing operations.

0 = Through plane mode
1 = Across plane mode

Some rectangle drawing commands disregard this bit and force Across plane mode for increased speed.

2 LASTPIX = Last Pixel Null

0 = Last pixel is drawn
1 = Last pixel is not drawn

For line commands (CMD_LINE, SSV, or CMD_LINEAF) the current position (CUR_X, CUR_Y) is moved to the end of the line but the last pixel is not drawn. This is used for those mixes (such as XOR) which would produce the wrong color if the same pixel were drawn twice (e.g. at the end of one polygon segment and at the start of the next segment).

For rectangles, this bit has different effects depending on which rectangle command is issued and also the state of the INC_X and INC_Y bits as listed below:

CMD_RECT
INC_X = 0 Leftmost column not drawn
INC_X = 1 Rightmost column not drawn

CMD_RECTV1
INC_Y = 0 Top row not drawn
INC_Y = 1 Bottom row not drawn

CMD_RECTV2
No effect

CMD_BITBLT
INC_X = 0 Leftmost column not drawn
INC_X = 1 Rightmost column not drawn

3 LINETYPE (Vector Enable).

0 = Enable Bresenham line drawing
1 = Enable vector line drawing

When vector line drawing is enabled (LINETYPE = 1) and a CMD_NOP issued, writes to the SHORT_STROKE register allow the specified vector to be drawn at the current position. When LINETYPE = 1, and a draw line command (CMD_LINE) is issued, a vector of length MAJ_AXIS_PCNT is drawn in the direction specified by LINEDIR. Lines drawn with LINETYPE = 1 are restricted to 45° angles. Lines of any slope may be drawn using the Bresenham drawing registers (LINETYPE = 0).

4 DRAW This bit is the equivalent of a pen up/pen down flag on a plotter.

0 = Marking disabled for line and BitBlts. Current position is moved, but no pixels are modified.
1 = Marking enabled.

Note: This bit should be set when attempting to read or write bitmap data.

5 INC_X (X positive) This bit along with INC_Y (below) determine which quadrant the slope of a line lies within. They also determine the orientation for rectangle draw commands. The upper left corner of the screen is the origin (x,y = 0,0).

0 =	Lines are drawn in negative X direction (right to left).
1 =	Lines are drawn in positive X direction (left to right).

6 YMAJAXIS For Bresenham line drawing commands, this bit determines which axis is the independent or major axis. Bits INC_X, and INC_Y define which quadrant the slope falls within; this bit further defines the slope to within an octant.

0 =	X is major (independent) axis.
1 =	Y is major (independent) axis.

7 INC_Y (Y positive) This bit along with INC_X (above) determine which quadrant the slope of a line lies within. They also determine the orientation for rectangle draw commands. The upper left corner of the screen is the origin (x,y = 0,0).

0 =	Lines are drawn in the negative Y direction (up the screen).
1 =	Lines are drawn in the positive Y direction (down the screen).

7–5 LINEDIR When a line draw command (CMD_LINE) is issued with LINETYPE = 1, LINEDIR bits specify the angular direction in which the line is to be drawn (counterclockwise relative to the positive x-axis). The length of the line is determined by the MAJ_AXIS_PCNT register.

LINEDIR	Angular Direction
0 0 0	000°
0 0 1	045°
0 1 0	090°
0 1 1	135°
1 0 0	180°
1 0 1	225°
1 1 0	270°
1 1 1	315°

8 PCDATA (Pixel Data Enable)
This bit is used to indicate that data is to be read/written through the PIX_TRANS register during the draw / SSV command.

0 =	Pixel data transfer disabled
1 =	Pixel data transfer enabled. The drawing engine waits for read/write of the PIX_TRANS register for each pixel during a draw operation. Depending on the foreground and background source select bits of the command issued, this data is not necessarily used in determining the actual pixel drawn.

9 16BIT 16-bit operation.

0 =	PIX_TRANS register is accessed as an 8-bit register.
1 =	PIX_TRANS register is processed internally as two bytes in the order specified by BYTSEQ (below).

Note: The IBM 8514/A does not handle 8-bit data transfer cycles properly to the PIX_TRANS register (IN AL,DX or OUT DX,AL). When developing programs which must execute on any IBM 8514/A or compatible, only 16-bit I/O operations should be performed (IN AX,DX or OUT DX,AX) to this register.

11–10 Reserved = 0

12 BYTSEQ Byte Sequence affects both reads and writes of SHORT_STROKE and PIX_TRANS registers when 16BIT = 1.

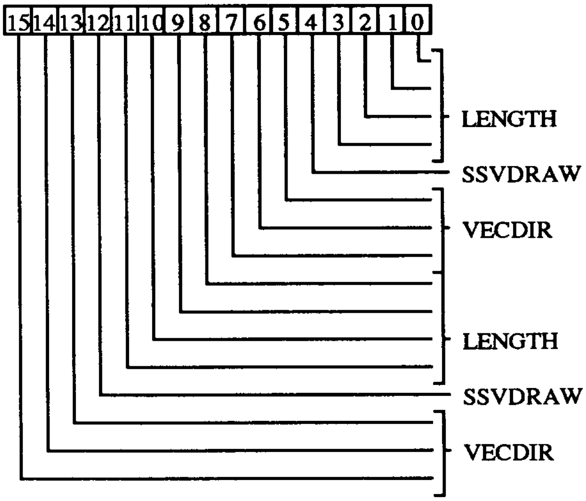
0 =	Take high byte first
1 =	Take low byte first

15–13 CMD Draw Command

111 =	Illegal
110 =	CMD_BITBLT (Copy rectangle)
101 =	CMD_LINEAF (outline)
100 =	CMD_RECTV2 (Fast filled Y direction rectangle)
011 =	CMD_RECTV1 (Fill rectangle in Y direction)
010 =	CMD_RECT (Fill rectangle in X direction)
001 =	CMD_LINE
000 =	CMD_NOP (This should be used when drawing SSVs)

SHORT STROKE VECTOR TRANSFER REGISTER (SHORT_STROKE)

Read at I/O Address 9EE8h (Chips Extension)
Write at I/O Address 9EE8h
Byte Accessible (16BIT=0)
Byte or Word Accessible (16BIT=1)



3–0 LENGTH = Length of vector projected onto major axis (this is also the number of pixels drawn). The current position is always moved LENGTH pixels from its starting value. A LENGTH of 0 leaves the current position unchanged.

4 SSVDRAW

0 = Don't write pixels
1 = Write pixels

7–5 VECDIR Vector Direction indicates the angle (measured counter-clockwise from horizontal right) at which the line is drawn.

VECDIR	Angular Direction
0 0 0	000°
0 0 1	045°
0 1 0	090°
0 1 1	135°
1 0 0	180°
1 0 1	225°
1 1 0	270°
1 1 1	315°

15–8 The lower 8 bits are duplicated in the upper 8 bits of the register so that two SSV commands may be issued simultaneously. These vectors are then executed in the order specified by BYTSEQ (CMD[12]).

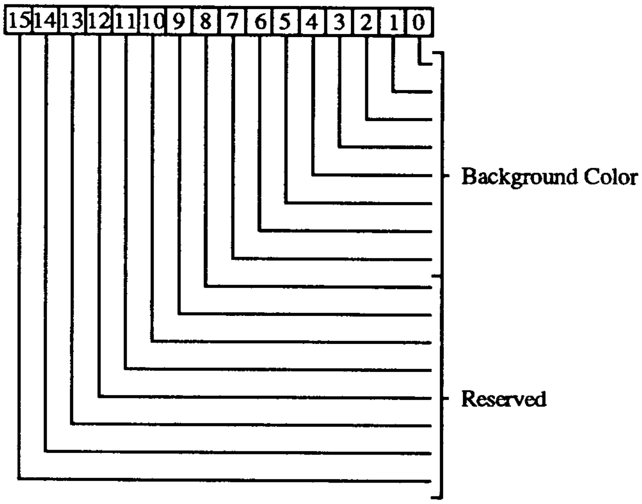
WARNING: The high byte of SHORT_STROKE must always be written in order for the command to execute. This means that if a byte write is performed to 9EE8h, it must be followed by a byte write to 9EE9h before the drawing engine will start. This register functions independent of the state of the 16BIT bit.

This is identical to the way in which the IBM 8514/A functions and is duplicated in the 82C480 for compatibility. The IBM Adapter Interface (AI) appears to always communicate with the 8514/A SHORT_STROKE register using 16–bit I/O instructions.

It is always possible to pad a single SSV instruction with a NUL byte. The drawing engine recognizes an SSV with LENGTH = 0 and SSVDRAW = 0 as a NOP.

BACKGROUND COLOR REGISTER (BKGD_COLOR)

Read at I/O Address A2E8h (Chips Extension)
Write at I/O Address A2E8h
Byte or Word Accessible



7–0 Background Color. This is the color used for writing pixels where the Foreground Color Mix is selected and FSS=00, or the Background Color Mix is selected and BSS=00.

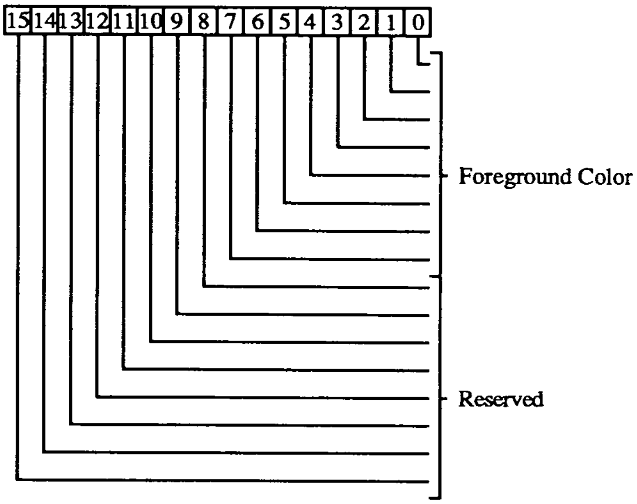
8-15 Reserved = 0

Warning: The BKGD_COLOR register is readable at A2E8h only when GPBUSY = DATARDY = 0. Otherwise, during drawing commands, A2E8h returns PIX_TRANS register data and perfoms as if a read were made to the PIX_TRANS register. This is required for IBM compatibility.

The BKGD_COLOR register is writable at A2E8h whenever GPBUSY = DATARDY = 0 or when PCDATA = 0. Otherwise, during drawing commands with PCDATA = 1, a write to A2E8h functions as a write to the PIX_TRANS register.

FOREGROUND COLOR REGISTER (FRGD_COLOR)

Read at I/O Address A6E8h (Chips Extension)
Write at I/O Address A6E8h
Byte or Word Accessible



7–0 Foreground Color. This is the color used for writing pixels where the Foreground Color Mix is selected and FSS=01, or the Background Color Mix is selected and BSS=01.

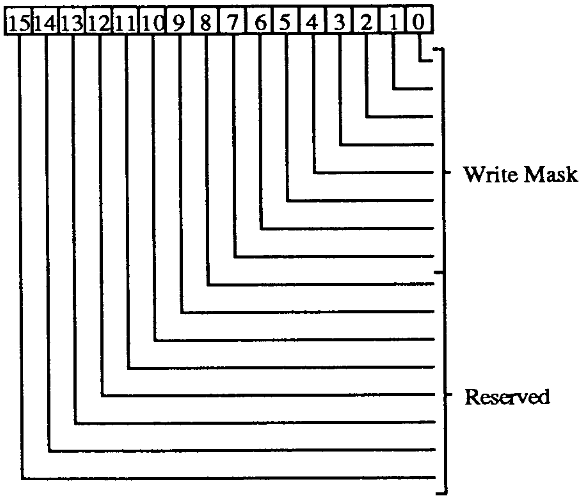
8-15 Reserved = 0

Warning: The FRGD_COLOR register is readable at A6E8h only when GPBUSY = DATARDY = 0. Otherwise, during drawing commands, A6E8h returns PIX_TRANS register data and performs as if a read were made to the PIX_TRANS register. This is required for IBM compatibility.

The FRGD_COLOR register is writable at A6E8h whenever GPBUSY = DATARDY = 0 or when PCDATA = 0. Otherwise, during drawing commands with PCDATA = 1, a write to A6E8h functions as a write to the PIX_TRANS register.

WRITE MASK REGISTER (WRT_MASK)

Read at I/O Address AAE8h (Chips Extension)
Write at I/O Address AAE8h
Byte or Word Accessible



7–0 Write Mask is used to prevent data from being modified in specific planes.

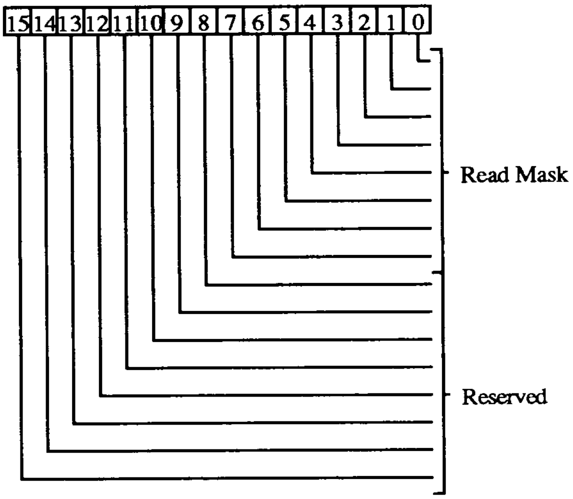
0 = Disable modifying plane
1 = Enable modifying plane

15–8 Reserved = 0

Note: WRT_MASK has some special functions during area fill operations. Refer to the discussion of the PLANEMODE bits with respect to area fill described in the Drawing Operations section of this manual.

READ MASK REGISTER (RD_MASK)

Read at I/O Address AEE8h (Chips Extension)
Write at I/O Address AEE8h
Byte or Word Accessible



7–0 Read Mask is primarily used when performing bounded fill operations (PIX_CNTL[2]=1). RD_MASK affects the following commands:

- Fill operations using CMD_RECT
- Fill operations using CMD_BITBLT
- Reading data in Across Plane Mode

RD_MASK is used to allow bit planes to participate in any logical or arithmetic mix:

0 = Allow plane to be read
1 = Prevent plane from being read

The effect of setting Read Mask = 0 for non-contiguous ranges of bits when an arithmetic mix is selected is undefined. This is because carry bits may propagate to bits (planes) which are masked by Read Mask.

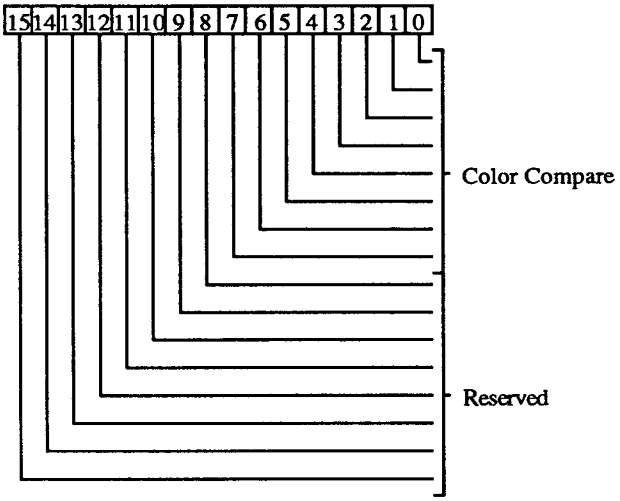
Warning: When performing a fill operation with PLANEMODE = 11, this register must contain a non-zero value even though it is not directly used for either the Boundary Mask or the Plane Mask.

15–8 Reserved = 0

Note: RD_MASK has some special functions during area fill operations. Refer to the discussion of the PLANEMODE bits with respect to area fill described in the Drawing Operations section of this manual.

COLOR COMPARE REGISTER (COLOR_CMP)

Read at I/O Address B2E8h (Chips Extension)
Write at I/O Address B2E8h
Byte or Word Accessible



7–0 Color Compare defines an 8-bit color which is compared to the destination data during BitBlts. The arithmetic comparison to be used (<,>,=,true,false, etc..) is specified by the COLCMPOP bits of the PIX_CNTL register (MFC[A][5:3]).

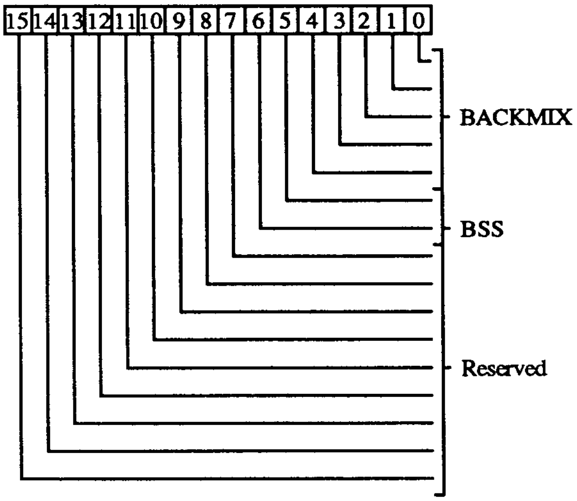
If the result of the comparison is true, the destination data is left unchanged. This is like the transparency concept used by Microsoft Windows, but with two differences:

1. The comparison is generalized to include more than just an equality test.
2. The comparison is with the destination data, not the source. IBM refers to this type of comparison as underpaint. PLANEMODE and DRAW can affect the Color Compare.

15–8 Reserved = 0

BACKGROUND MIX REGISTER (BKGD_MIX)

Read at I/O Address B6E8h (Chips Extension)
Write at I/O Address B6E8h
Byte or Word Accessible



4–0 BACKMIX The background mix defines the mix or raster op to be used in drawing operations. It is analogous to the Foreground Color Mix. One of these two mixes is selected for each pixel in the current nugget based on the PIX_CNTL register. All 16 logical mixes, and five arithmetic mixes (sum, difference, min, max, and average) are supported. Options also exist for sum, average, and difference mixes to include saturation (no sum can be greater than 0FFh, and no difference can be less than 0). This prevents visually undesirable effects when these mixes are selected.

6–5 BSS Background Source Select

BSS	Source Selected
0 0	Background Color
0 1	Foreground Color
1 0	Pixel data (PIX_TRANS)
1 1	Bitmap data

Pixel data may be supplied pixel by pixel (through plane mode) or nugget by nugget (across plane mode). This permits data transfer between the screen and system memory.
Bitmap data is the source data from the display buffer. The result of doing a line draw or SSV with this source selected is undefined. It is primarily used for BitBlt operations.

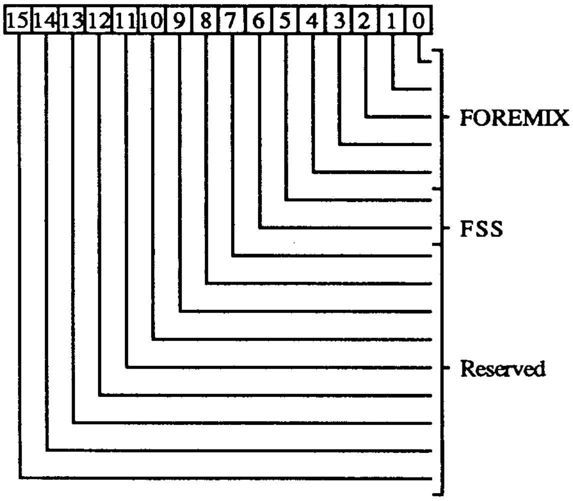
15–7 Reserved = 0

NOTE: In the table below, the destination (DST) operand is always the bitmap destination data, but the source (SRC) operand has four possible sources selected by the BSS bits. There are only 30 unique mixes (24–25, and 28–29 are duplicates).

BACKMIX	Result	BACKMIX	Result
0 0 0 0 0	not DST	1 0 0 0 0	min (SRC, DST)
0 0 0 0 1	0 (false)	1 0 0 0 1	DST–SRC with underflow
0 0 0 1 0	1 (true)	1 0 0 1 0	SRC–DST with underflow
0 0 0 1 1	DST	1 0 0 1 1	SRC+DST with overflow
0 0 1 0 0	not SRC	1 0 1 0 0	max (SRC, DST)
0 0 1 0 1	SRC xor DST	1 0 1 0 1	(DST–SRC)/2 with underflow
0 0 1 1 0	not (SRC xor DST)	1 0 1 1 0	(SRC–DST)/2 with underflow
0 0 1 1 1	SRC	1 0 1 1 1	(SRC+DST)/2 with overflow
0 1 0 0 0	not (SRC and DST)	1 1 0 0 0	DST–SRC with saturate
0 1 0 0 1	(not SRC) or DST	1 1 0 0 1	DST–SRC with saturate
0 1 0 1 0	SRC or (not DST)	1 1 0 1 0	SRC–DST with saturate
0 1 0 1 1	SRC or DST	1 1 0 1 1	SRC+DST with saturate
0 1 1 0 0	SRC and DST	1 1 1 0 0	(DST–SRC)/2 with saturate
0 1 1 0 1	SRC and (not DST)	1 1 1 0 1	(DST–SRC)/2 with saturate
0 1 1 1 0	(not SRC) and DST	1 1 1 1 0	(SRC–DST)/2 with saturate
0 1 1 1 1	not (SRC or DST)	1 1 1 1 1	(SRC+DST)/2 with saturate

FOREGROUND MIX REGISTER (FRGD_MIX)

Read at I/O Address BAE8h (Chips Extension)
Write at I/O Address BAE8h
Byte or Word Accessible



4–0 FOREMIX The foreground mix defines the mix or raster op to be used in drawing operations. It is analogous to the Background Color Mix. One of these two mixes is selected for each pixel in the current nugget based on the PIX_CNTL register. All 16 logical mixes, and five arithmetic mixes (sum, difference, min, max, and average) are supported. Options also exist for sum, average, and difference mixes to include saturation (no sum can be greater than 0FFh, and no difference can be less than 0). This prevents visually undesirable effects when these mixes are selected.

6–5 FSS Foreground Source Select

FSS	Source Selected
0 0	Background Color
0 1	Foreground Color
1 0	Pixel data (PIX_TRANS)
1 1	Bitmap data

Pixel data may be supplied pixel by pixel (through plane mode) or nugget by nugget (across plane mode). This permits data transfer between the screen and system memory.

Bitmap data is the source data from the display buffer. The result of doing a line draw or SSV with this source selected is undefined. It is primarily used for BitBlt operations.

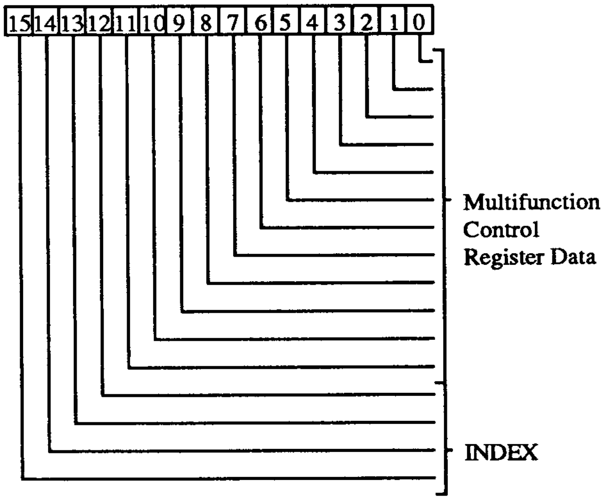
15–7 Reserved = 0

NOTE: In the table below, the destination (DST) operand is always the bitmap destination data, but the source (SRC) operand has four possible sources selected by the FSS bits. There are only 30 unique mixes (24–25, and 28–29 are duplicates).

FOREMIX	Result	FOREMIX	Result
0 0 0 0 0	not DST	1 0 0 0 0	min (SRC, DST)
0 0 0 0 1	0 (false)	1 0 0 0 1	DST–SRC with underflow
0 0 0 1 0	1 (true)	1 0 0 1 0	SRC–DST with underflow
0 0 0 1 1	DST	1 0 0 1 1	SRC+DST with overflow
0 0 1 0 0	not SRC	1 0 1 0 0	max (SRC, DST)
0 0 1 0 1	SRC xor DST	1 0 1 0 1	(DST–SRC)/2 with underflow
0 0 1 1 0	not (SRC xor DST)	1 0 1 1 0	(SRC–DST)/2 with underflow
0 0 1 1 1	SRC	1 0 1 1 1	(SRC+DST)/2 with overflow
0 1 0 0 0	not (SRC and DST)	1 1 0 0 0	DST–SRC with saturate
0 1 0 0 1	(not SRC) or DST	1 1 0 0 1	DST–SRC with saturate
0 1 0 1 0	SRC or (not DST)	1 1 0 1 0	SRC–DST with saturate
0 1 0 1 1	SRC or DST	1 1 0 1 1	SRC+DST with saturate
0 1 1 0 0	SRC and DST	1 1 1 0 0	(DST–SRC)/2 with saturate
0 1 1 0 1	SRC and (not DST)	1 1 1 0 1	(DST–SRC)/2 with saturate
0 1 1 1 0	(not SRC) and DST	1 1 1 1 0	(SRC–DST)/2 with saturate
0 1 1 1 1	not (SRC or DST)	1 1 1 1 1	(SRC+DST)/2 with saturate

MULTIFUNCTION CONTROL INDEX REGISTER (MULTIFUNC_CNTL)

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible



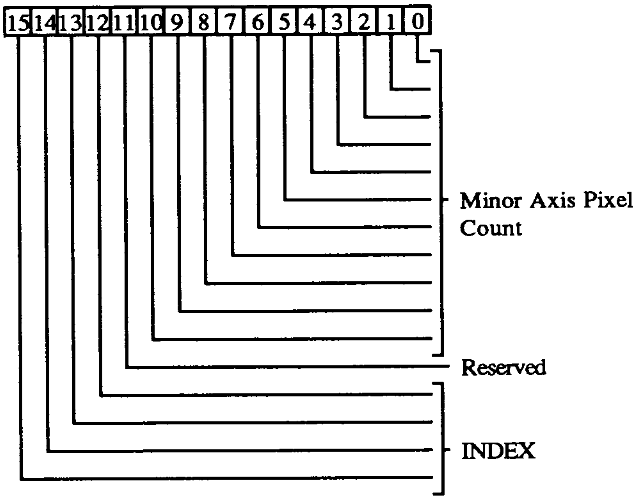
11–0 Multifunction Control Register Data. The MULTIFUNC_CNTL register is the index to a set of registers at the same address. The INDEX bits are used to determine which of the nine currently implemented registers is being written to. This data field is defined for each of the indexed registers in the pages to follow.

The 82C480 also allows these registers to be read. The index of the register to read is written to EC3[3:0] (Multifunction Control Register Read Index), then the register data is read from the least significant 12 bits at this register address (BEE8h); bits 15:12 are undefined.

15–12 INDEX indicates which Multifunction control sub-register is currently being written to.

MINOR AXIS PIXEL COUNT REGISTER (MIN_AXIS_PCNT) (MFC[0])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible



10–0 Minor Axis Pixel Count is used to define the height for BITBLT and rectangle commands. The actual height is:

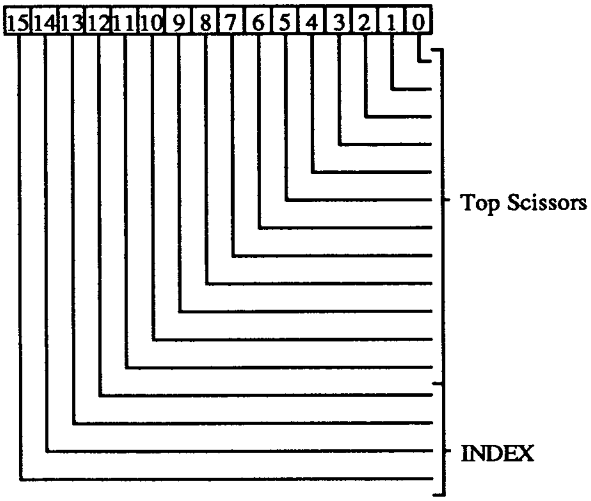
Minor Axis Pixel Count + 1

11 Reserved = 0

15–12 INDEX = 0000

TOP SCISSORS REGISTER (SCISSORS_T) (MFC[1])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible

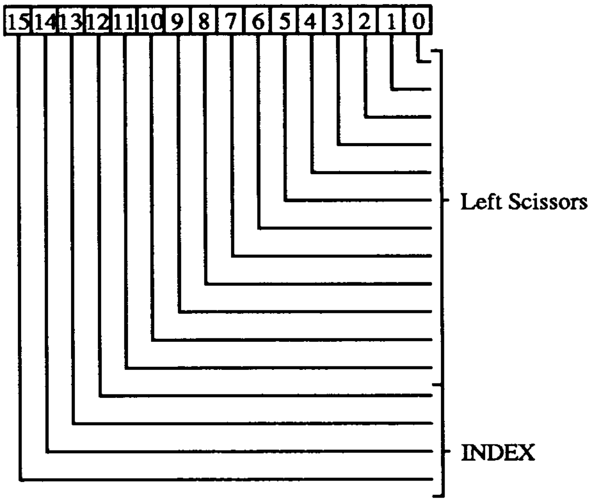


11–0 Top Scissors is the minimum Y coordinate value for the scissors rectangle.

15–12 INDEX = 0001

LEFT SCISSORS REGISTER (SCISSORS_L) (MFC[2])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible

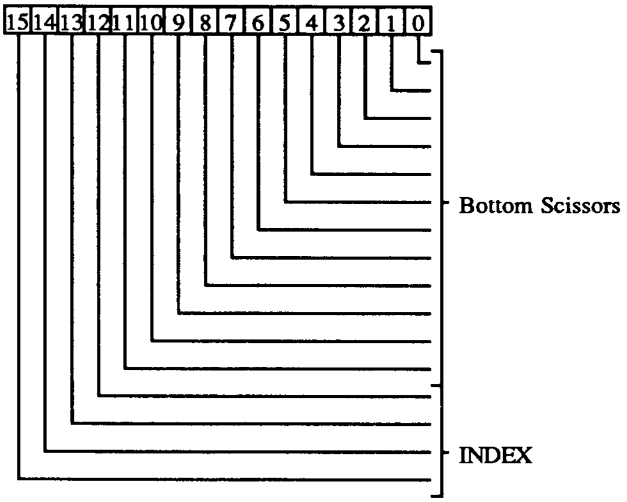


11–0 Left Scissors is the minimum X coordinate value for the scissors rectangle.

15–12 INDEX = 0010

BOTTOM SCISSORS REGISTER (SCISSORS_B) (MFC[3])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible

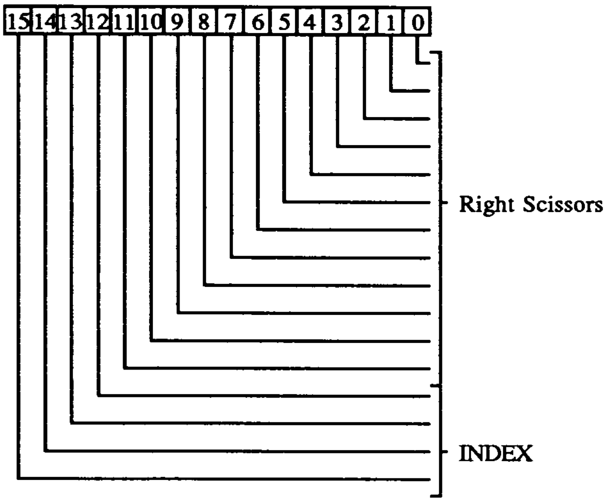


11–0 Bottom Scissors is the maximum Y coordinate value for the scissors rectangle.

15–12 INDEX = 0011

RIGHT SCISSORS REGISTER (SCISSORS_R) (MFC[4])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible

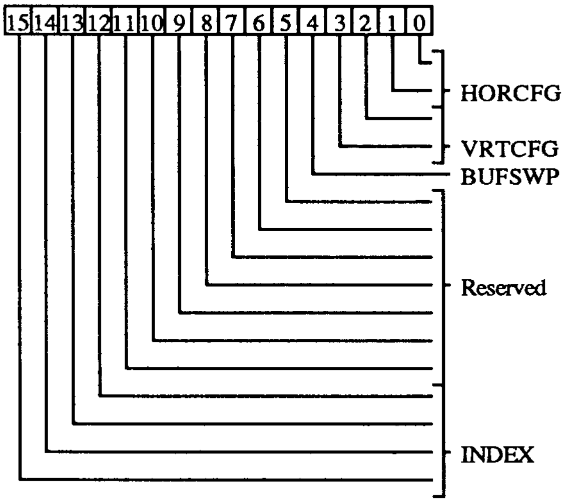


11–0 Right Scissors is the maximum X coordinate value for the scissors rectangle.

15–12 INDEX = 0100

MEMORY CONTROL REGISTER (MEM_CNTL) (MFC[5])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible



1–0 HORCFG This field contains information which the 8514/A uses to determine the X-Coordinate divisor. Bit 0 is used to indicate 5-pixel operation and should only be set if EC2[11] = 1 (the 5PN bit). Bit 1 determines if the VRAMs are interleaved in the horizontal direction. The interleave table results in the following:

HORCFG	Interleave
0 0	4
0 1	5
1 0	8
1 1	10

HORCFG = 10 always for the IBM 8514/A. For the 82C480, only the last two values, 10 and 11 are of interest.

3–2 VRTCFG specifies the amount by which to divide the Y coordinate internally when addressing different memory configurations. These bits also determine the bank interleave factor in the Y (vertical) direction. These bits are re-interpreted when 1Mbit VRAMs are used to maintain software compatibility with the 8514/A (which uses 256Kbit VRAMs). These bits are related to the number of VRAM banks in the system. The number of installed banks may be determined by reading EC2[9:8] (BANKS).

Writing VRTCFG = 00 (if this does not match the reset value) invokes Pseudo 8-Plane mode (PS8). Writing VRTCFG ≠ 00 deactivates PS8 mode.

	VRAM banks	
VRTCFG	256K	1M
0 0	2	–
0 1	4	1
1 0	–	2
1 1	–	4

VRTCFG = 01 is the 8514/A standard value (4 banks either 4-planes or 8-planes deep).

VRTCFG = 1x is used for extended resolutions beyond 1024x768.

4 BUFSWP is used to select planes when in Pseudo 8-plane mode:

0 = select buffer 0 (lower 4 planes)
1 = select buffer 1 (upper 4 planes)

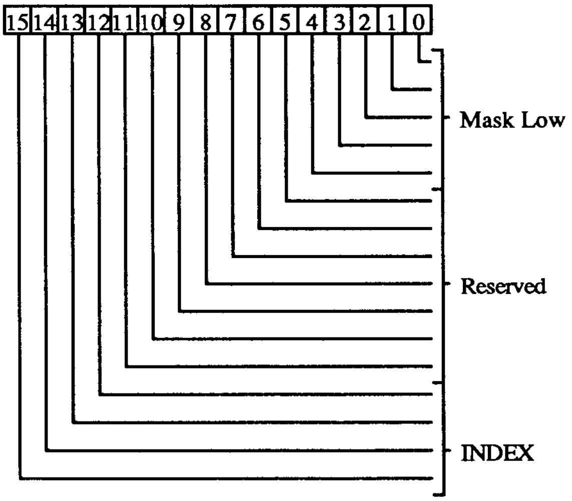
BUFSWP = 0 always when not in Pseudo 8-plane mode.

11–5 Reserved = 0

15–12 INDEX = 0101

FIXED PATTERN LOW REGISTER (PATTERN_L) (MFC[8])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible



4–0 Mask Low is used to select the mix on a pixel by pixel basis. If MIXSEL = 01, then the pattern registers are used to select foreground or background mix. Mask Low applies to even numbered nuggets (0 is leftmost on the screen); Mask High is applied to odd numbered nuggets.

0 = select BACKMIX
1 = select FOREMIX

The bits are in left to right order as they would appear on the display:

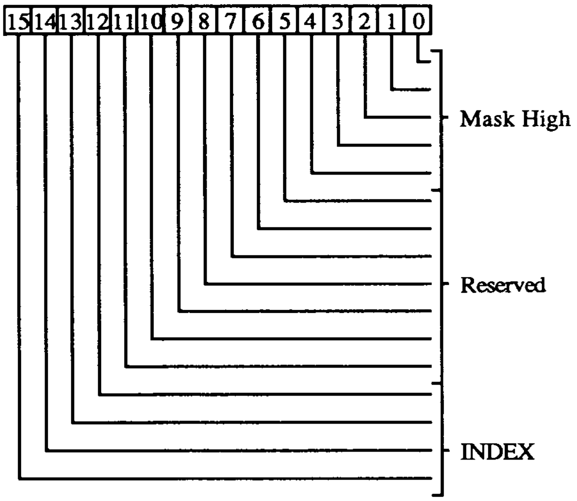
D4 = pixel 0 (within a nugget)
D3 = pixel 1
D2 = pixel 2
D1 = pixel 3
D0 = pixel 4 (if 5PN mode)

11–5 Reserved = 0

15–12 INDEX = 1000

FIXED PATTERN HIGH REGISTER (PATTERN_H) (MFC[9])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible



4–0 Mask High is used to select the mix on a pixel by pixel basis. If MIXSEL=01, then the pattern registers are used to select foreground or background mix. Mask High applies to odd numbered nuggets (0 is leftmost on the screen); Mask Low is applied to even numbered nuggets.

0 = select BACKMIX
1 = select FOREMIX

The bits are in left to right order as they would appear on the display:

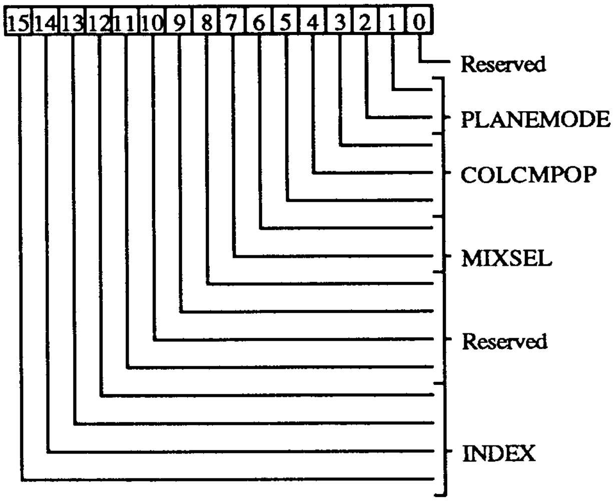
D4 = pixel 0 (within a nugget)
D3 = pixel 1
D2 = pixel 2
D1 = pixel 3
D0 = pixel 4 (if 5PN mode)

11–5 Reserved = 0

15–12 INDEX = 1001

PIXEL CONTROL REGISTER (PIX_CNTL) (MFC[A])

Read at I/O Address BEE8h (Chips Extension)
Write at I/O Address BEE8h
Byte or Word Accessible



0 Reserved = 0

2–1 PLANEMODE

0 0 = Normal Operation
0 1 = Indeterminate
1 0 = Fill area using RD_MASK as boundary mask. Does not fill second edge of boundary. Plane Mask is a mixture of RD_MASK and WRT_MASK.
1 1 = Fill area using WRT_MASK as boundary mask. Does fill second edge of boundary. Plane Mask is WRT_MASK (although RD_MASK must be non-zero for correct operation).

Note: For a complete description of area fill and the interactions of the PLANEMODE bits and masks, refer to the Area Fill discussion in Drawing Operations.

5–3 COLCMPOP (Color Comparison Operation) These bits determine the comparison operation performed on each pixel.

COLCMPOP	Operation	
0 0 0	False	(always write DST)
0 0 1	True	(never write DST)
0 1 0	DST >=	COLOR_CMP
0 1 1	DST <	COLOR_CMP
1 0 0	DST ≠	COLOR_CMP
1 0 1	DST =	COLOR_CMP
1 1 0	DST <=	COLOR_CMP
1 1 1	DST >	COLOR_CMP

7–6 MIXSEL (Mix Select)

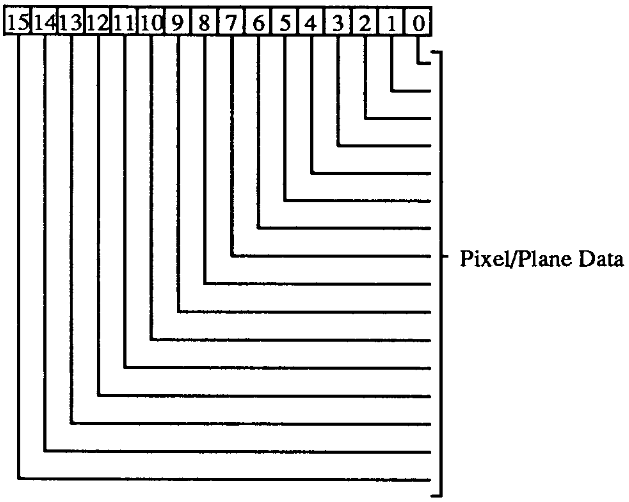
0 0 = FOREMIX is always used
0 1 = PATTERN_L,PATTERN_H select mix. (1 = FOREMIX, 0 = BACKMIX)
1 0 = Variable pixel data (from PIX_TRANS) selects the mix (1 = FOREMIX, 0 = BACKMIX).
1 1 = SRC selects the mix (used to implement transparency during a BITBLT command).

11–9 Reserved = 0

15–12 INDEX = 1010

PIXEL DATA TRANSFER REGISTER (PIX_TRANS)

Read at I/O Address E2E8h
Write at I/O Address E2E8h
Byte Accessible (16BIT=0)
Byte or Word Accessible (16BIT=1)



15–0 Pixel/Plane Data In through planes mode (PLANAR=0), bits D7:0 and D15:8 map onto bit planes 7:0 of an individual pixel. In across planes mode (PLANAR=1), bits D4:0 and D12:8 map onto pixels 0:4 within a nugget (1 bit per pixel).

The PIX_TRANS register can be used to allow data reads or writes to the display buffer. It can also be used to apply any arbitrary pattern during BITBLT or LINE drawing operations.

WARNING: The type of data transfer operation performed depends upon the state of the 16BIT bit in the CMD register. The IBM 8514/A actually only functions correctly when 16 bit I/O transfers are performed. The useful data transferred however does depend on 16BIT. That is, if 16BIT=0, then an IN AX,DX reads 8 bits of valid data on the low data bus (AL) and no valid data on the high data bus (AH). Although an IN AL,DX to E2E8h will read the correct data, it does not clear the DATARDY bit and the graphics processor will remain idle until an IN is performed to E2E9h.

The high byte of PIX_TRANS must always be read/written in order for the command to continue execution. This means that if a byte write is performed to E2E8h, it must be followed by a byte write to E2E9h before the drawing engine will start. This is independent of the state of the 16BIT bit. The effect that the 16BIT bit has is that when 16BIT=1, both high and low bytes written are used (in the order specified by BYTSEQ). If 16BIT=0 however, only the low byte is executed (regardless of the state of BYTSEQ and the data written to E2E9h is ignored). The same is true of the read operation. This means that if a byte read is performed from E2E8h, it must be followed by a byte read from E2E9h before the drawing engine will start again. This is independent of the state of the 16BIT bit. The effect that the 16BIT bit has is that when 16BIT=1, both high and low bytes read are valid data (if BYTSEQ = 1 the high and low bytes are swapped). If 16BIT=0 however, only the low byte is valid data (the data read from E2E9h should be discarded).

This is how the IBM 8514/A functions and is duplicated in the 82C480 for compatibility. The IBM Adapter Interface (AI) always appears to communicate with the 8514/A using 16–bit I/O instructions so the high byte dummy instructions are never needed. If you intend to write code which will work universally, it is probably safest to follow IBM's lead and always set 16BIT=1. If an odd number of pixels need to be written to PIX_TRANS then the high byte of the last output is discarded. Likewise, when inputing an odd length from the PIX_TRANS register, the high byte of the final data transfer may be discarded.

Note: For compatibility, this register may also be read at A2E8h and A6E8h during drawing commands.

82C480 System Interface

The 82C480 can interface as a slave to 2 different system buses: MCA (Micro Channel) and ISA (Industry Standard Architecture or PC/AT). The 82C480 auto-configures to 8-bit or 16-bit slots on either bus. MCA mode is the default, ISA mode is selected by tying ISA/ low.

In either mode, the 82C480 supports I/O reads, I/O writes, memory reads, and interrupts. It does not support memory writes (the only supported memory is ROM), burst mode transfers, matched-memory transfers, 32-bit transfers, or any Bus Master functions. All reads or writes are of the 'Asynchronous-Extended' variety where an I/O Channel Ready signal from the chip controls the length of the cycle.

82C480 registers are mapped into areas which skip around in I/O space (see the IO Map, at the beginning of the register section). However, since PC/AT system boards decode only the lower 10 address bits as 'official' I/O addresses, all internal registers appear at PC/AT I/O address 2E8h. The rest of the address bits appear on the ISA bus, and are necessary for proper register selection, but are "don't cares" for some motherboard peripherals. 8514/A I/O mapping minimizes the impact on the PC or AT.

Only the initialization ROM is memory-mapped, defaulting to the range C8000h-C9FFFh for ISA bus or C6000h-C7FFFh for MCA bus configurations. The display buffer is not memory mapped. The only way to read or write display memory is through the use of 82C480 commands. Control and data registers for such commands are mapped into I/O space.

VGA support is provided by the Micro Channel Video Bus Extension (MCA systems), or the VESA video pass-through connector (ISA systems). In VGA mode, video data and video clock are multiplexed to the 8514/A palette/DAC, and syncs are multiplexed to the 8514/A video connector. In VGA mode, the 8514/A responds to all writes to the VGA palette (3C6h-3C9h); this ensures the 8514/A palette displays the VGA colors correctly. It does not respond to reads of the VGA palette to prevent bus contention.

Operation of the upper 8 data bits on the MCA or ISA bus is controlled by Byte High Enable (BHE/). When the 82C480 is installed on an 8-bit bus, BHE/ is not driven (it is located on the 16-bit connector extension); an internal pull-up holds BHE/ inactive in this case. The 82C480 will not activate Data_Size_16 (DS16/) for word or odd byte transfers if BHE/ = 1. The host will convert word transfers to byte transfers in this case. Internal byte steering logic directs odd-byte data over D7:0 when BHE/ = 1.

The 82C480 may be used without external data transceivers in motherboard applications where 4 mA drive is sufficient, or with transceivers where 24 mA drive is required. It provides RDLO/ and RDHI/ outputs to control the direction of these transceivers.

Bus signals are connected as follows.

82C480 Pin	MCA Signal	ISA Signal
A23:20	A23:20	Ground
A19:0	A19:0	SA19:0
BHE/	-SBHE	-SBHE
AEN	M/-IO	AEN
IORD/	-S1	-IOR
IOWR/	-S0	-IOW
MEMR/	-CMD	-SMEMR
MEMW/	MADE 24	-SMEMW
IOCS16/	-CD DS16	-IOCS16
RDY	CD CHRDY	IOCHRDY
RFSH/	-RFSH	-RFSH
IRQ	-IRQ9	IRQ9
RESET	CHRESET	RESETDRV
ISA/	-CD SETUP	Ground
MEMCS16/	-CD SFDBK	-MEMCS16

Micro Channel Interface

Micro Channel mode is selected by default (ISA/ pin not grounded). The following Micro Channel signals are not used: ADL/, PREEMPT/, DS16RTN, CHRDYRTN, BURST/, TC/, ARBn, ARB/-GNT (all Bus Master/DMA functions), CHCK/, AUDIO, AUDIO GND, and all 32-bit extended signals. Addresses are latched on the falling edge of CMD/.

POS operations are described later in this document.

Industry Standard Architecture (ISA) Interface

ISA Mode is selected by tying the ISA/ pin low. The 82C480 doesn't decode LA23:17, so SMEMR/ is used instead of MEMR/. SMEMR/ is used only for the ROM; SMEMW/ isn't used. The 82C480 doesn't generate MEMCS16/, so it does not support 16-bit ROM; this is not a serious limitation, because those wanting higher performance can copy the

ROM to shadow RAM, which offers higher performance than 16-bit ROM.

The following ISA signals are not used: ALE, LA23:17, CLK, NMI/, DRQn, DACKn, T/C, MASTER/, MEMCS16/, 0WS/. Only SA addresses are used on chip; these do not need to be latched so ALE is unnecessary.

Extended Industry Standard Architecture (EISA) Interface

EISA is a superset of the ISA bus. To interface the 82C480 to an EISA bus, use the signal connections as described for the ISA bus. No EISA extension to the ISA architecture are supported.

POS Operations are not supported in ISA/EISA mode.

Queue

To improve performance, the 82C480 provides a write queue (called just the "queue") for all registers at I/O address 8000h or greater. This includes all graphics processor registers. The only registers whose write data is not queued are CRT controller, palette, interrupt, and extended configuration registers. The queue improves efficiency of writing drawing commands and parameters, as well as for Variable Data---data provided to/from the host via the PIX_TRANS register while executing a drawing command. This is the mechanism whereby data is copied between the bitmap and system memory. Thousands of pixels may be transferred during a single 82C480 command, so a queue is imperative.

Because drawing parameters are queued, changes to these registers (such as changing the drawing color) will not take effect until any drawing operations in progress are finished. This allows parameters to be set for the next command immediately.

The queue is 8 words by 20 bits (16 data, 4 address). A23:15 and A9:0 are decoded to select the queue for writing; A13:10 go into the queue along with the 16 data bits, and are decoded upon leaving the queue to select the destination register.

The GP_STAT register at 9AE8h is used to poll the status of the read and write queues:

Bit-9	GPBUSY (1=Busy)
Bit-8	DATARDY (1=Pixel Transfer Data Ready)
Bit-7:0	00000000 = 8 words available (queue empty)
	00000001 = 7 words available
	00000011 = 6 words available
	00000111 = 5 words available
	00001111 = 4 words available
	00011111 = 3 words available
	00111111 = 2 words available
	01111111 = 1 words available
	11111111 = 0 words available (queue full)

Bits 7:0 of this register is the Queue State (QS) field which shows the number of words available in the queue. Nicknamed the "thermometer" register, it shows the number of bytes available, encoded linearly. QS tells not only that there are words available, but how many; this can reduce the amount of polling necessary. QS = 0 does not imply the 82C480 graphics processor is idle; it may have consumed all the data from the queue but still be executing a drawing command. Always test the GPBUSY bit to determine if the graphics processor is idle.

The "read queue" is only one word, and is used only for Pixel Transfer Data. The Pixel Transfer Data Ready bit (DATARDY) indicates there is data waiting to be read from the PIX_TRANS register (E2E8h).

Interrupts

The interrupt request (IRQ) pin supports interrupts on the MCA or ISA bus. IRQ is normally attached to IRQ9 on either the MCA or ISA bus. In MCA mode, IRQ acts as an active-low open-collector output. In ISA mode, IRQ is 3-stated when no interrupts are enabled, it is driven low when interrupts are enabled but not active, and is driven high when interrupts are enabled and active (this is like the EGA/VGA). IRQ9 may be shared with another adapter on the ISA bus if only one has its interrupts enabled at a time. In either case, IRQ has no internal pull-up or pull-down; it relies on the motherboard to pull IRQ to an inactive state.

IRQ has IOL min = –16 mA. While this does not technically meet the ISA or MCA 24 mA spec, the board designer may still want to drive IRQ9 directly, instead of using an external 74LS125 driver.

There are four internal sources of interrupts:

IRQ3 Idle (GPIDLE): The 82C480 graphics processor is idle and the queue is empty. The GPIDLE interrupt may be used as a substitute for polling of the GPBUSY bit prior to writing a drawing command and parameters. Once the 82C480 is idle, the next command and parameters may be written to the queue at C000h-FFFFh without polling the Queue State bits.

IRQ2 Queue over/underflow (INVALIDIO): An attempt has been made to write to the queue when no words were available, or to read PIX_TRANS when no data was ready.

IRQ1 Write inside scissor rectangle (PICKFLAG): The 82C480 has written the display buffer within the Scissor Rectangle while executing a CMD_LINE, CMD_LINEAF, CMD_RECT, CMD_RECTV1, CMD_RECTV2, the destination phase of a CMD_BITBLT command or a SSV. IRQ1 is used for picking operations.

IRQ0 Vertical Blank (VBLNKFLAG): Vertical blanking is active. IRQ0 may be used to synchronize operations (such as scrolling) to the end of the frame.

Each internal interrupt goes active whenever the specified condition is true. The state of the internal interrupts may be read from the corresponding interrupt field in the Subsystem Status Register. However, the IRQ pin will only be activated when the condition becomes active, and only if that particular internal interrupt was already enabled by the corresponding interrupt enable bit in the Subsystem Control Register. To inactivate IRQ, each active internal interrupt must be cleared by writing 1 to the corresponding interrupt reset bit in the Subsystem Control Register. Interrupts may be cleared individually or in combination.

82C480 Memory Interface

Memory Architecture

The 82C480 is designed to use 256K or 1M VRAMs only (not DRAMs). Write–Per–Bit and Enhanced Page Mode features are required. The 82C480 uses a 40–bit VRAM data bus. This bus is called the Pixel Bus, and the pins are labeled PxDy where x = 0–4 and indicates the pixel within the nugget (see next section) and y = 0–7 indicates the bit (plane). Each pixel has its own write enable pin (WE4:0/), and each plane is selectively written using the VRAM Write–Per–Bit mask.

The 82C480 uses a packed pixel architecture. Each pixel is stored in a pair of VRAMs; one holds the lower nibble, one the upper nibble. 4-bpp (bit-per-pixel) systems are implemented by simply not populating the upper nibble VRAM locations. Because the VRAMs are four bits wide, the smallest number of bits per pixel for the display buffer is 4. However, by using the Plane Read Mask (RD_MASK), Plane Write Mask (WRT_MASK), and Palette Mask (DAC_MASK) registers, you can effectively divide the 4-bpp display buffer into two 2-bpp buffers or four 1-bpp buffers, each with the same resolution as the original 4-bpp buffer. This does not reduce VRAM requirements or increase resolution compared to 4-bpp, but does allow implementing a double-buffered 2-bpp or quadruple-buffered 1-bpp system using the same display buffer as for a 4-bpp system.

The 82C480 has a separate Memory Clock (MCLK) which may be adjusted to match the speed of the VRAMs used. Maximum frequencies are 40 MHz (for 100 ns VRAMs), 32 MHz (120 ns VRAMs), or 25 MHz (150 ns VRAMs). MCLK is asynchronous to Nugget Clock (NCLK), which controls video timing.

Nuggets

At this point, some terms must be defined: a nugget is the fundamental "word" of the 82C480. A nugget is either four or five pixels, depending on the 5 Pixel Nugget (5PN) bit. A double nugget is two nuggets (eight or ten pixels), from the same row and column address of a horizontally interleaved pair of RAM banks. Double nuggets are the basic unit of all 82C480 horizontal video parameters.

Within a nugget, the pixels are "backwards" order. Pixel 0 (the rightmost pixel when the nugget is viewed as a 40 bit word) appears in the leftmost position on the screen. Pixels 1, 2, 3, and 4 appear in left-to-right order on the screen. In Across-Planes mode, Pixel Transfer Data (PIX_TRANS) bits appear in the "correct" order. Only the least significant 5 bits of PIX_TRANS are used; bit-4 controls pixel–0 (the leftmost pixel of the nugget as it appears on the screen), bits 3, 2, 1, and 0 respectively control pixels 1, 2, 3, and 4.

The two halves of a double nugget are referred to as A and B, where A appears to the left of B on the screen. The IBM 8514/A always uses four banks of RAM, organized as two horizontally-interleaved A/B bank pairs. The banks are called 0A, 0B, 1A, and 1B.

The 82C480 may use one, two, or four RAM banks. The 82C480 VRAM banks are named Bank 0, 1, 2, and 3; if a single bank is used it is Bank 0; if two banks are used they are Banks 0 and 1. It can still interleave between banks using just a single pixel data bus, but it can interleave between any two banks, not just between members of an A/B pair. If a single bank is used, of course no interleaving is possible. If two banks are used, Banks 0 and 1 are interleaved horizontally ("horizontally" refers to the location of words from the same row and column address as they appear on the screen). If four banks are used, Banks 0 and 1 are interleaved horizontally on even scan lines, and Banks 2 and 3 are interleaved horizontally on odd scan lines. Banks 0 and 2 or 1 and 3 are interleaved vertically. Banks 1 and 2 or 0 and 3 may interleave cycles when moving diagonally during line drawing operations.

Variable size nuggets allow efficient support of 1280 horizontal resolution. When configured for 1024 horizontal resolution (32–bit nuggets) using eight 1 Mb VRAMs, the longest scan line which can be shifted out of the standard VRAM shift registers is:

$$\frac{(512 * 4)\text{bits/RAM} * 8 \text{ VRAMs}}{8 \text{ bits/pixel} * 2 \text{ scan lines/row address}} = 1024$$

By adding two more 1Mb VRAMs (40-bits, or 5 pixels, per nugget), the maximum horizontal resolution is increased to 1280:

$$\frac{(512 * 4)\text{bits/RAM} * 10 \text{ VRAMs}}{8 \text{ bits/pixel} * 2 \text{ scan lines/row address}} = 1280$$

Variable size nuggets have the following advantages:

1. Each VRAM row has exactly 2048 or 2560 (twice 1024 or 1280) pixels; there are no VRAM words wasted.
2. Each scan line is aligned to VRAM row boundaries and is totally contained in the VRAM row. There is no need for split buffer VRAMs nor data transfer cycles "on the fly".

1Mb VRAM Support

The 82C480 allows a 1024 x 768 system to be implemented with just four 1Mb VRAMs (for 16 colors) or eight (for 256 colors). Supporting 1M VRAMs reduces overall graphics system chip count and board area requirements. Also, the cost per bit of 1M VRAMs is predicted to fall below that of 256K VRAMs.

When using 1M VRAMs, some registers are interpreted differently to maintain register compatibility with the 8514/A.

Pseudo 8-Plane Mode

When the 8514/A board is configured for minimum memory (512KB) it normally supports four planes at 640 x 480 or 1024 x 768 resolution. Pseudo 8-plane mode (abbreviated PS8 mode) supports eight planes at 640 x 480 resolution, even though only four planes of the display buffer are populated. In this mode, data is stored as two separate 4-bpp buffers at different addresses, but is displayed together with Buffer 0 appearing as the low nibble and Buffer 1 appearing as the high nibble. During drawing operations, the high and low nibble (i.e. Buffer 1 and 0) are written separately, and the color indices and plane masks are left shifted by four bits when drawing the upper nibble. This mode is entered by setting the Y_Coordinate Divisor (VRTCFG) to 00, and the buffers are selected by setting the Swap bit; BUFSWP = 1 for Buffer 1 or BUFSWP = 0 for Buffer 0. BUFSWP = 0 always when not in Pseudo 8-Plane mode.

Because a nugget normally contains consecutive pixels from the same scan line, the 8514/A requires programming of VRTCFG = 01 (normal) or 00 (pseudo 8-plane mode). The 82C480 interprets the VRTCFG registers to enable Pseudo 8-Plane mode for software compatibility, but it doesn't use VRTCFG to determine the number of VRAM banks.

The 82C480 finds the number of VRAM banks installed by reading strapping options on the CAS3:2/ pins, which are latched into Extended Configuration register EC2[9:8].

Pseudo 8-plane mode is only allowed in medium resolution (640 x 480). The Pseudo 8-plane logic is incorporated in the 82B484 Video Support Chip.

Memory Configurations

The 82C480 supports 20 possible memory configurations, which may be categorized by the type of VRAM (256 Kb or 1 Mb), number of planes (4 or 8), number of banks (1, 2 or 4), and the nugget width (4 or 5 pixels). To simplify things, the configurations are numbered, with up to four characters as follows:

1, 2, or 4: the number of banks (1 bank
is only allowed for 1Mb VRAM)

M: if 1 Mb VRAMs are used
W: if 5 pixel nuggets (wide nuggets)
D: if 8 planes are used (deep memory)

The following figure summarizes the configurations:

VRAM Type	# of VRAM Banks	Nugget Width (pixels)	# of Planes	Memory Configuration	# of VRAM Chips	Bitmap size (KB)	Maximum Resolution
256K	2	4	4	2	8	256	512x384x4
			8	2D	16	512	512x384x8
		5	4	2W	10	320	640x480x4
			8	2WD	20	640	640x480x8
	4	4	4*	4	16	512	1024x768x4
			8*	4D	32	1024	1024x768x8
		5	4	4W	20	640	1280x1024x4
			8	4WD	40	1280	1280x1024x8
1M	1	4	4*	1M	4	512	1024x768x4
			8*	1MD	8	1024	1024x768x8
		5	4	1MW	5	640	1280x1024x4
			8	1MWD	10	1280	1280x1024x8
	2	4	4	2M	8	1024	1024x768x4
			8	2MD	16	2048	1024x768x8
		5	4	2MW	10	1280	1280x1024x4
			8	2MWD	20	2560	1280x1024x8
	4	4	4	4M	16	2048	2048x1536x4
			8	4MD	32	4096	2048x1536x8
		5	4	4MW	20	2560	2560x2048x4
			8	4MWD	40	5120	2560x2048x8

* 8514/A register-compatible configurations

Note: Maximum resolutions assume a 4:3 aspect ratio (except 1280 x 1024 which is not 4:3 but is in common use). Maximum horizontal resolution is determined as shown on the previous page, by the combined length of the VRAMs serial buffers.

Memory configurations are selected by strapping options on CAS3:2/ (BANKS), WE4/ (5PN), MA8 (1MVRAM), and P4D7 (8PLANE) pins. All options may be connected to jumpers, or overwritten by software to allow user upgrades (see Initialization section). The most common user upgrade would be from a 4-plane to an 8-plane configuration (e.g. 1M to **1MD**); another upgrade might be from a 1024 x 768 adapter to a 1280 x 1024 one (e.g. **1M** to **1MW**, or **1MD** to **1MWD**).

Memory configurations **4** and **4D** are identical to the 8514/A's base and extended memory configurations, and are compatible with any 8514/A software, whether it uses AI- or register-level commands. Memory configurations **1M** and **1MD** are similar to **4** and **4D**, respectively, except that they use 1Mb VRAMs instead of 256Kb VRAMs which the 8514/A uses. Modes **1M** and **1MD** would normally be preferable to Configurations **4** and **4D**, because 1M VRAMs offer savings in part count, area, and cost. Normally, using 1M VRAMs would require changing several register values. Extra logic has been added to the 82C480 so that Configurations **1M** and **1MD** use all the same register values as Configurations **4** and **4D**, so they are compatible with all 8514/A software, using either AI or register-level interface.

Configurations other than **4**, **4D**, **1M**, or **1MD** are referred to as Extended Configurations. Extended Configurations in general will require an extended AI or drivers, although some 8514/A drivers will be compatible with some Extended Configurations.

Whenever possible, the 82C480 interleaves VRAM banks to improve performance. VRAMs are designed so that pixels on the same scan line (i.e. same Y-coordinate) map to the same row address; bank interleaving is very effective in the X-direction because interleaved page-mode cycles can be used across a whole scan line (unless interrupted for RAM refresh). Only 1 to 4 scan lines fall in the same row address, so page mode can only be used for one to four consecutive pixels in the Y direction. This makes interleaving much less effective in the Y direction. Bank interleaving is impossible in Memory Configuration 1M and its variations, which use only one bank.

In two-bank configurations, Bank-0 and Bank-1 are interleaved in the X direction on a nugget basis---four or five pixels in Bank 0, four or five in Bank 1, four or five in Bank 0, ...

In four bank configurations, banks are interleaved in both the X and Y direction. Banks interleave 0/1 or 2/3 in the X direction; Banks interleave 0/2 or 1/3 in the Y direction. Banks interleave 0/3 or 1/2 when movement is on a diagonal, such as for Short Stroke Vectors (SSVs).

Configuration **2** and variants are Extended Configurations allowing very low-cost "entry level" 8514/A systems using 256K VRAMs with up to 640 x 480 resolution, and are user-upgradeable to full 8514/A compatibility (Configuration **4** and **4D**).

Configuration **4** and **4D** are register-compatible to the 8514/A with minimum (512KB) and maximum (1MB) memory configurations, respectively. These configurations allow resolutions up to 1024 x 768. They may be user-upgraded to Configuration **4W** or **4WD** (1280 x 1024 resolution), but an extended AI or drivers will be required.

Memory Configurations **1M** and **1MD** are register compatible to the IBM 8514/A in minimum and maximum memory configurations, respectively, but use 4 or 8 1Mb VRAMs instead of 16 or 32 256Kb VRAMs, resulting in board area and cost savings. They support resolutions up to 1024 x 768 (Note: the maximum resolution possible maintaining a 4:3 aspect ratio from a 1024 x 1024 bitmap is 1024 x 768).

Memory Configurations **1MW** and **1MWD** are extended configurations which support resolutions up to 1280 x 1024. This is also the maximum resolution available in a **2MW** or **2MWD** memory configuration. Because a 1280 x 1024 resolution implemented in 1 bank would consume all of the video memory for the display, it may be preferred to use the 2 bank implementation. This would leave some off-screen memory which could be used by the AI or extended 8514 driver.

Memory Configurations **4M** and **4MD** are extended configurations which support resolutions up to 2048 x 1536.

Memory Configurations **4MW** and **4MWD** are extended configurations which support resolutions up to 2560 x 2048. This is the highest resolution configuration.

RAM Cycle Types

The 82C480 performs the following types of VRAM cycles:

1. Read (page mode)
2. Write (page mode)
3. Read-modify-write (page mode)
4. RAS-only refresh
5. Data Transfer

Read, write, and RMW cycles may be bank interleaved. RAS-only refresh and Data Transfer cycles are performed on all banks simultaneously.

All read, write, or RMW cycles start by writing the VRAM Write-Per-Bit register, used to implement bit plane masking. The mask data comes from the Plane

Write Mask register (WRT_MASK). The VRAM Write-per-Bit registers are written by holding WE3:0/ all active at the falling edge of RAS/, while driving the WRT_MASK data on the pixel data bus. Note that because the WRT_MASK is eight bits wide and VRAMs are only four bits wide, some receive the upper nibble of the WRT_MASK, and some the lower nibble.

There are three sources of VRAM cycle requests:

1. Drawing engine (read, write, and RMW)
2. CRT Controller (Data Transfer)
3. RAM refresh logic (RAS-only refresh)

Data Transfer cycles have highest priority. All CAS/ lines go low simultaneously only during a Data Transfer cycle.

RAM refresh cycles have the next highest priority.

Drawing engine cycles have the lowest priority. These cycles always start by writing the VRAM write-per-bit register. Drawing engine operations will be done in a single page-mode cycle, unless one of the following happens:

1. RAM row address boundary is crossed by drawing engine; indicated internally by:
   a. Y Coordinate bit 2 or above changing.
   b. Y Coordinate register (CUR_Y) reloaded by host.
   c. Y Coordinate register (CUR_Y) reloaded internally.
   d. Changing from source to destination phase of Copy Rectangle (CMD_BITBLT) command. This may happen many times for a large rectangle. Once each time the internal scratch register is filled.
2. The WRT_MASK register is written by the host (requiring new write-per-bit values to be written to the VRAM).
3. A Data Transfer and/or RAM refresh cycle interrupts the page-mode operation.

82C480 Video Interface

The Video circuitry controls addressing and timing for the VRAM display and memory refresh functions. It handles high or low resolutions in interlaced or non-interlaced display modes with equal ease, and generates programmable sync polarities for controlling different monitor modes.

The hardware is based around 2 principal counters, the Horizontal Counter (HCNT) and the Vertical Counter (VCNT). The values in the Video Registers are compared with these counters to generate Blanking and Sync signals, as well as VRAM Refresh and Data Transfer cycle requests. Two other counters determine Horizontal and Vertical Sync Pulse Widths. There are no interdependencies on Blank and Sync start times.

The Display Control (DISP_CNTL) register controls whether the Video Circuitry is enabled or reset, and whether interlacing is enabled. It also controls certain memory configuration parameters for display.

Horizontal Timing

All horizontal timings and register values are in terms of double-nuggets (8 or 10 pixels). The Horizontal Counter indicates the horizontal double-nugget currently being displayed. 0 corresponds to the leftmost actively displayed double-nugget on the screen. The Horizontal Total (H_TOTAL) register indicates the total length of a single scan line, including horizontal retrace time.

Between 0 and H_TOTAL, 4 time events are detected:

1. 1/2 H_TOTAL
2. Horizontal Blank Start
3. Horizontal Sync Start
4. Full H_TOTAL

At 1/2 H_TOTAL (usually about 2/3 across the active display area), the following occur:

1. Refresh Counter Increments (causing a refresh cycle)
2. Vertical Counter Increments (if interlace)
3. Vertical Sync Width Counter Increments (if interlace and VSYNC is active)

The Horizontal Displayed (H_DISP) register specifies the double-nugget at which the internal Horizontal Blank (HBLNK) signal begins. HBLNK is ORed with Vertical Blank (discussed below) to form the BLANK output signal. HBLNK continues through the end of the scan line (Full H_TOTAL).

VRAM Data Transfer Cycles are triggered by the beginning of HBLNK. For standard VRAMs, all Data Transfers are initiated in this manner.

The Horizontal Sync Start (H_SYNC_STRT) register specifies the double-nugget at which the output signal horizontal sync (HSYNC) begins. The Horizontal Sync Width (H_SYNC_WID) register specifies the width, in double-nuggets, of the HSYNC signal. Using this scheme, an HSYNC may be programmed to start anywhere within a scan line (even at the end), and generate a full pulse width.

Horizontal Sync Polarity (HSYNCPOL), along with Vertical Sync Polarity (VSYNCPOL) bits indicate the current display resolution to the 8514 monitor:

HSYNCPOL	VSYNCPOL	Resolution
0	0	1024 x 768
0	1	720 x 350
1	0	640 x 400
1	1	640 x 480

At Full H_TOTAL, the scan line ends, and a new one is ready to begin at the left of the screen. The following events take place:

1) Refresh Counter Increments (causing refresh)
2) Vertical Counter Increments (always)
3) Vertical Sync Width Counter Increments (if VSYNC active)
4) Horizontal Blank Ends
5) Vertical Blank and certain other signals synchronize
6) Horizontal Total Counter Resets to Zero

Vertical Timing

The Vertical Counter indicates the vertical scan line currently being displayed. 0 corresponds to the top actively displayed scan line on the screen.

The Vertical Total register (V_TOTAL) indicates the total number of scan lines in a single frame, including vertical retrace time.

Between 0 and V_TOTAL, 3 time events are detected:

1) Vertical Blank Start
2) Vertical Sync Start
3) Full V_TOTAL

The Vertical Displayed (V_DISP) register specifies the scan-line at which the internal vertical blank (VBLNK) request signal begins. This signal is synchronized as VBLNK only at the ends of horizontal scan lines (see interlace discussion below). After synchronization, it is ORed with HBLNK to form the BLANK output signal. VBLNK continues through the end of the frame (V_TOTAL) when it ends synchronously on full H_TOT.

The Vertical Sync Start (V_SYNC_STRT) register specifies the scan-line at which the output signal VSYNC begins. The Vertical Sync Width register (V_SYNC_WID) specifies the width, in scan-lines (or half-scan-lines for interlace) of the VSYNC signal. Using this scheme, a VSYNC may be programmed to start anywhere within a frame (even at the end), and generate a full pulse width.

At Full V_TOTAL, the frame ends, and a new one is ready to begin at the top of the screen. The following events take place:

1) Vertical Blank Ends Synchronous to H_TOT
2) Vertical Total Counter Resets to Zero

In the V_TOTAL register, bits 11:3 indicate a number of scan lines, multiplied by a scan line modulus defined in the Display Control (DISP_CNTL) register; bits 2:0 of V_TOTAL are an adjustment added to the value in bits 11:3 to allow specifying values that are not an even multiple of the scan line modulus. Bits 11:3 are named Vertical Total Base (VTB) and bits 2:0 are named Vertical Total Adjust (VTADJ).

Similarly, bits 11:3 of V_DISP are named VDB and bits 2:0 are named VDADJ, and bits 11:3 of V_SYNC_STRT are named VSB and bits 2:0 are named VSADJ. If the adjust value is larger than the modulus, the timing is disabled. Actual timings are 1 greater than programmed in the V_TOTAL, V_DISP, and V_SYNC_STRT registers.

The scan line modulus is determined by the Memory Configuration (MEMCFG) and Double Scan (DBLSCAN) bits of the Display Control register, as shown below. MEMCFG is related to the number of banks of 256K VRAMs installed. DBLSCAN = 1 indicates that each scan line is repeated on the display. These values are the same whether 256Kb or 1Mb VRAMs are used; the 82C480 internally corrects for addressing differences with 1Mb VRAMs.

MEMCFG	DBLSCAN	Modulus	Notes
0 0	0	2	Indicates PS8 Mode
0 0	1	4	Not allowed with 1M VRAMs
0 1	0	4	8514/A (Not PS8 Mode)
0 1	1	8	
1 0	0	6	
1 0	1	12	
1 1	0	8	
1 1	1	16	

For example, in 640 x 480 resolution, V_TOTAL = 418h; which after splitting into separate fields is VTB = 83h (131d) and VTADJ = 0. The scan line modulus = 4. The resulting number of scan lines is 131*4 + 0 + 1 = 525.

For another example, in 640 x 480 resolution, V_DISP = 3BBh which can be separated into VDB = 77h (119d) and VDADJ = 3. The resulting scan line on which BLANK/ goes active is 119*4 + 3 + 1 = 480, just as expected.

Interlace mode is enabled when the Display Control register INTERLACE bit = 1. Interlace does nothing more than enable the Vertical Counter to count half scan lines rather than full scan lines; all values programmed in interlaced mode must be divided by 2 to get the actual value. For true interlacing to occur, V_TOTAL must be programmed to output an odd number of half scan lines per field.

As an example, in 1024 x 768 resolution V_TOTAL = 660h, which separates into VTB = CCh (204d) and VTADJ = 0. The resulting number of scan lines per field is (204*4 + 0 + 1)/2 = 408.5. The total number of scan lines per frame (two fields) is 817.

As a last example, in 1024 x 768 resolution V_DISP= 5FBh, which separates into VDB = BFh (191d) and VDADJ = 3. The resulting scan line on which BLANK/ goes active is (191*4 + 3 + 1)/2 = 384.

Horizontal timing occurs on a full scan line basis only. This means that scan lines have an alternating odd/even relationship with the half-scan-line count in alternate frames. That is, in even frames, the first displayed scan line starts when VCNT = 0. In odd frames, the first displayed scan line starts when VCNT = 1.

On top of the very regular VSYNC period, a slightly irregular VBLANK period occurs. Requests for VBLANK occur on a very regular N/2 + 1/2 basis, but they are synchronized only at the ends of horizontal scan lines. This means VBLANK maintains the same odd/even relationship to the half-scan-line count as does the displayed scan lines.

This has a side effect that VBLNK for even frames is one scan line longer than VBLNK for odd frames.

Refresh Counter

The Refresh Counter is an independent 9-bit counter which increments every half-scan-line (regardless of interlace mode). Every time its value changes, a VRAM memory refresh cycle is triggered, and the counter value is used as the Row Address.

For refresh to occur, this scheme requires the Video Circuitry to be enabled (using DISP_CNTL[5]). Also, the Horizontal Total register must be loaded with a value such that, with the current Video Clock input frequency, the VRAM refresh requirements are met.

Serial Bus

Nugget Clock (NCLK) is 1/8 or 1/10 pixel clock (PCLK) frequency, 30 MHz max. It is used by the 82C480 to generate CRT signals HSYNC, VSYNC, and BLANK/.

The 82C480 chipset can select the video clock from up to eight oscillators, using the CLKSEL2:0 outputs. In a pure 8514/A implementation using the 82C480, only CLKSEL0 is needed to select between the 25.175 MHz and 44.9 MHz clocks.

VRAM Serial Clocks (SCLK1:0) and Serial Output Enables (SOE3:0/) are generated by the 82B484 whenever BLANK/ is inactive; the 82B484 compensates BLANK/ for serial data pipeline delays, so the palette receives data and blanking at the same time. SCLK1:0 are stopped during BLANK/, so the next scan line will start on the correct pixel and no VRAM timings will be violated during the data transfer cycle.

There are two basic **Video Configurations, A and B**. There will of course be variations in external logic, depending on the video frequencies required, and whether the MCA Video Extension or VESA video pass-thru connector (for VGA video) is supported.

Video Configuration A enables data from only one bank at a time and supports video rates up to approximately 125 MHz. Systems using **Video Configuration A** would normally use the 82B484 Video Support Chip (for video rates up to 80MHz).

All Memory Configurations but **1M** and its variants use two shift clocks (SCLK1:0) and four output enables (SOE3:0/). SOE3:0/ drive banks 3 to 0 respectively. SCLK0 drives banks 0 and 2, SCLK1 drives banks 1 and 3; SCLK1 and SCLK0 run 180° out of phase at 1/8 the video frequency (except in Pseudo 8-Plane mode); each rising edge of SCLK0 or SCLK1 generates four pixels. SOE0/ and SOE1/ also run 180° out of phase, as do SOE2/ and SOE3/. If there are two banks, Banks 0 and 1 are alternately enabled on nugget boundaries. If there are four banks, Banks 0 and 1 are alternately enabled on even scan lines; Banks 2 and 3 are alternately enabled on odd scan lines ("odd" and "even" assume non-interlaced screens and count from scan line 0 at the top of the screen). In **Video Configuration A**, serial data from all four banks may be bused together.

Configuration **1M** and its variants are a special case, because they use only one bank of VRAM. Only four pixels are output each serial clock cycle, so SCLK0 runs at 1/4 VCLK frequency in this configuration only. NCLK runs at 1/8 VCLK as usual. SOE0/ is active the entire active display time; SOE3:1/ and SCLK1 are inactive.

Video Configuration B enables video from banks 0 and 1 (or 2 and 3) simultaneously and supports video rates up to about 300 MHz. In this Configuration, SCLK0 still drives Banks 0 and 2 and SCLK1 still drives Banks 1 and 3, but they run in phase at 1/8 video frequency, each rising edge producing 8 pixels. The shift registers must be 8 bits long and are loaded only every eighth VCLK. SOE3:0/ still drive Bank 3 through 0 respectively. SOE0/ and SOE1/ are active the entire display time on even scan lines, and SOE2/ and SOE3/ are active the entire display time on odd scan lines. In **Video Configuration B**, serial data from Banks 0 and 2 must be bused separately from Banks 1 and 3 (making the serial data bus 64/80 bits wide).

Video Configuration B may only be used with 2 or 4 banks of VRAM, but Video Configurations are otherwise not related to Memory Configurations. As a practical matter, however, Memory Configurations using 256K VRAMs (2, 4, and their variants) do not support resolutions high enough to require **Video Configuration B**. Only Memory Configurations **2M**, **4M**, and their variants are logical candidates for **Video Configuration B**; and only if you need more than the 80 MHz video rate the 82B484 supports. In these cases you must use ECL for the serial data and clock logic.

The serial data bus is 40 bits wide (80 wide for **Video Configuration B**) and numbered similarly to the pixel data bus: SxDy, where x is the pixel position in the (double) nugget (x = 0-4 for **Video Configuration A** and 0-9 for **Video Configuration B**) and y is the bit position in the pixel (y = 0-7).

Another option is to use a RAMDAC like the Brooktree Bt451/458 in **Video Configuration A**. These accept four pixels at time and multiplex them internally. Such a RAMDAC supports video rates well above 80 MHz and eliminates the need for TTL shift registers. Pseudo 8-Plane mode is not supported by the Bt451/458.

Alternate Video Registers

The 82C480 maintains two alternate video register sets which may be used to support non-standard display parameters when programmed for one of the two standard 8514/A resolutions. They are called the Alternate High Resolution (AHR) and Alternate Low Resolution (ALR) register sets. These alternate video register sets are intended for use at 1024x768 and 640x480 resolutions respectively. Each of the alternate register sets contains its own copy of the following registers and bit fields:

1.	Horizontal Total	H_TOTAL
2.	Horizontal Sync Start	H_SYNC_STRT
3.	Horizontal Sync Width	H_SYNC_WID
4.	Vertical Total	V_TOTAL
5.	Vertical Sync Start	V_SYNC_STRT
6.	Vertical Sync Width	V_SYNC_WID
7.	MEMCFG	DISP_CNTL[2:1]
8.	INTERLACE	DISP_CNTL[4]
9.	DBLSCAN	DISP_CNTL[3]
10.	CLKSEL[2:1]	EC3[10:9]

The alternate video registers/bits overlay the I/O locations of the normal registers/bits. The set of registers which is visible to the system is determined by the alternate register select bits (AHRS and ALRS) of register EC3[5:4]. The register set which is active in the 82C480 is determined by the alternate register enable bits (AHRE and ALRE) of register EC3[7:6].

When writing the alternate registers, AHRS and ALRS act as "paging registers" selecting the register sets as follows:

AHRS	ALRS	Selects
0	0	Normal Registers
0	1	Alt. Low-resolution registers
1	0	Alt. High-resolution registers
1	1	Write Protect Video registers

The intention is to load the alternate registers only once, for example under control of a config.sys file. This would then allow the 82C480 to override programs which write the IBM 8514/A values directly to the display registers with values tuned for a specific system. This is useful in systems which have monitors capable of displaying non-interlaced 1024 x 768 resolution or systems requiring a higher vertical refresh frequency than the IBM standard.

The alternate registers are activated by the 8514/A CLKSEL[0] bit in conjunction with AHRE and ALRE as follows:

CLKSEL	AHRE	ALRE	Video Register Active
0	X	0	Normal
0	X	1	Alternate Low-resolution
1	0	X	Normal
1	1	X	Alternate High-resolution

The alternate video registers do not normally need to be saved, however they all are read/write accessible when selected.

Note: Extended Configuration register EC3 should only be written in byte mode. This is because EC3 contains the alternate register selects in its low byte, and several alternate register set bits in its high byte. Writing this register as a word value may yield unpredictable results.

82C480 Initialization / Configuration

Many aspects of 82C480 operation are configurable. All of these may be selected by hardware strapping options. These strapping options are latched internally at the falling edge of RESET; other functions may be multiplexed over these pins after reset. Options may be set by jumpers; the jumpers are connected to tristate buffers that drive the option pin when enabled by RESOUT/. Most strapping options may be read and written by software, so jumpers may be omitted in favor of a configuration utility. This method combines the best of hardware and software configuration techniques.

Options are defined so that they all default to a simple 8514/A board configuration, making special 82C480 device drivers unnecessary. Extended (non-8514/A) registers are always visible to the system (there is no enable mechanism).

Reset

On reset, all 82C480 pins except RESOUT/ are tristated.

8BITDAC, ROMCS/, MA8, WE4/, CAS3:2/, and P4D7:0 latch configuration options at the falling edge of RESET; they have internal pull-ups to insure they default to 1 unless driven low to select the 0. RDHI/ and RDLO/ have 50K pullups to insure the data transceivers do not drive the system bus during reset. ROMCS/, PALRD/, and RAS/ have 50K pullups to insure the EPROM, RAMDAC, and VRAMs remain in the quiescent state during reset. All internal counters are reset when RESET = 1. Software registers are undefined after reset, unless noted otherwise.

A soft reset may be performed using the Subsystem Control register GPCTRL bits. This has the same effect as a hardware reset, except no pins are tristated, counters are not reset, and configuration options are not loaded.

POS Operation

Micro Channel POS_ID (100h-101h) are mapped to on-board ROM. When SETUP/ is active, reads of 100h or 101h activate ROMCS/ [POSEN/] and force ROM_PAGE_SEL[2:0] = 1. This allows any desired POS_ID to be burned in ROM. Boards without ROM may use tristate buffers enabled by POSEN/ to provide a POS_ID.

Register 102H, bit 0 (Card Enable) is implemented in the 82C480 itself. Card Enable is read/write accessible and responds only when SETUP/ is low.

POS registers are accessible as byte registers only and are not accessible in ISA mode.

Configuration Options

<u>8/16-bit Bus Interface</u>

The 82C480 configures itself for 8- or 16-bit bus operation. This is possible because BHE/ (SBHE/ on ISA bus) is on the 16-bit bus extension, so it is unconnected when the board is in an 8-bit slot. An internal pullup forces the 82C480 BHE/ pin inactive in this case. A0 and BHE/ are interpreted as follows:

A0	BHE/	MCA/ISA Bus Definition	82C480 Definition
0	0	16-bit transfer	16-bit transfer
0	1	Even byte on D7:0	Even byte on D7:0
1	0	Odd byte on D15:8	Odd byte on D15:8
1	1	8-bit adapter	Odd byte on D7:0

<u>Strapping Option - MCA/ISA Bus</u>

SETUP/ is only used in MCA systems; the bus always drives SETUP/ high during reset. SETUP/ may be grounded to select ISA bus interface. This option may not be read by software.

<u>Strapping Option - 1M VRAMs</u>

MA8 is only used by 1Mb VRAMs; if it is 1 on reset 1Mb VRAMs are assumed, if it is 0 256Kb VRAMs are assumed. An internal pull-up makes the pin default to 1Mb VRAMs; drive with RESOUT/ to select 256Kb. This option may be read/written at Extension Configuration register EC2[10].

<u>Strapping Option - Number of VRAM Banks</u>

CAS3/ and CAS2/ are only used in configurations of more than 2 banks, so may be used to indicate the number of VRAM banks installed on the board to the 82C480. Internal pullup resistors maintain high levels on these pins if unconnected or used to drive VRAM chip CAS/ inputs. In a 1 bank configuration, both CAS2/ and CAS3/ are tied low (or to RESOUT/); in a 2 bank configuration, CAS2/ is tied low and CAS3/ is left unconnected; in a 4 bank configuration, CAS2/ and CAS3/ are connected to bank 2 and 3 VRAM chip CAS/ inputs, respectively. The number of VRAM banks may be read/written at

Extension Configuration register EC2[9:8].

The number of VRAM banks should be reflected in the VRTCFG and HORCFG bits of the Memory Control register (MEM_CNTL). These bits are interpreted differently for 1Mb VRAMs to maintain register compatibility with the IBM 8514/A board. The least significant bit of HORCFG determines whether 4 or 5 pixel nuggets are used. This bit is related to the depth of the VRAM banks, not the number of banks.

		256Kb VRAM		1Mb VRAM		
VRT-CFG	HOR-CFG	Inter-leave VxH	Total Banks	Inter-leave VxH	Total Banks	Notes
0 0	0 X	1x1	1	n/a	n/a	
0 0	1 X	1x2	2	1x1	1	PS8 Mode
0 1	0 X	2x1	2	1x2	2	
0 1	1 X	2x2	4	1x1	1	8514/A
1 0	0 X	3x1	3	2x2	4	
1 0	1 X	3x2	6	2x1	2	
1 1	0 X	4x1	4	n/a	n/a	
1 1	1 X	4x2	8	3x1	3	

Strapping Option - 8/6 bit DACs

The IBM 8514/A uses the Inmos IMSG176 palette/DAC, which has 6-bit DACs. Other boards may use 8-bit DACs. Still others may use palette/DACs such as the Brooktree Bt478, whose DACs may function as either 8- or 6-bit, depending on the state of their 8/6 input. The 8BITDAC pin is provided for such DACs. There is no pull-up on 8BITDAC; a pull-up or pull-down must be provided to select 8 or 6 bit operation on reset; the state of 8BITDAC on the trailing edge of reset will be latched, and driven back out after reset. 8BITDAC may be read or set by software at Extended Configuration register EC2[13].

Monitor Options

The 82C480 reads analog video SENSE from LM339 comparators, just as VGA chips do, via the SENSE pin. The 82C480 also reads the Monitor ID bits from the Video Connector via pins MS2:0. SENSE may be read from the Display Status register DISP_STAT[0]; Monitor ID bits may be read from the Subsystem Status register SUBSYS_STAT[6:4]. If ROM paging is not used, P4D6 can be tied low and MS2:0 may be connected directly to the the video connector; if the ROM_PG outputs are used, MS2:0 must be driven by a tristate buffer enabled by RESOUT/. The state of the ROM PAGING bit may be read/written at EC2[6].

Other Strapping Options

All remaining options are latched from the pixel bus (P4Dy) on the falling edge of RESET. The P4Dy pixel bus allows 8 possible options, of which only 5 (P4D7–P4D3) are currently implemented. Extension bits currently unused are reserved for future definition. Internal pullups default these options to 1.

To select a 0 option, you should drive that pin with a tristate driver (e.g. 74LS244) enabled by RESOUT/. The inputs to these drivers may be grounded for options which will not change, or to jumpers for user selection. For those PxDy lines which are not to be used by the display buffer, 0 options may be selected by driving the pin directly with RESOUT/. All these options appear in the Extended Configuration Registers and may be read and/or set by software.

Strapping Option - 8 Bit Planes (8BP)

P4D7 = 1 on the falling edge of RESET (default) indicates that 8 planes of VRAM are installed; 0 indicates 4 planes. Emulates the function of a jumper on the 8514/A board. 8PLANE is latched in Subsystem Status register SUBSYS_STAT[7]. 8BP may also be read or written at Extended Configuration register EC2[7].

Strapping Option - ROM Size (ROMSIZE)

If P4D5=1 on the falling edge of RESET (default), an 8K ROM address space is decoded in system memory. If P4D5 is driven low with RESOUT/, a 32K ROM address space is decoded. The size of the ROM also effects its base address (see the ROM Base Address description below). ROMSIZE may be read or written at EC2[5]. If ROMCS/ is sampled low at reset, no ROM is mapped into address space.

Strapping Option - (ROMBASE)

P4D4:3 are latched on the falling edge of RESET to indicate the ROM base address. This configuration option allows all of the ROM address spaces in system memory to be relocated in case of address conflicts. Address conflicts are possible with any number of peripheral adapters. ROMBASE may be read or written at EC2[4:3]. Care must be taken when writing EC2 not to destroy the configuration information contained therein.

ROMBASE	8K MCA	8K ISA	32K
0 0	C8000h	C6000h	D0000h
0 1	D8000h	D8000h	D8000h
1 0	C0000h	C0000h	C0000h
1 1 (default)	C6000h	C8000h	C8000h

82C480 Pixel Operations

Mixes

The 82C480 supports 30 mixes, or raster ops. These are selected by the FOREMIX bits of FRGD_MIX or the BACKMIX bits of the BKGD_MIX register, as shown in the table below. In addition to all 16 possible 2-operand logical mixes, there are 14 arithmetic mixes including min, max, average, sum, and difference (either SRC – DST or DST – SRC). Sum, difference, and average may be saturating (i.e. no sum can be greater than FFh and no difference can be less than 0) or allow overflow (sum and average) and underflow (difference). Saturation prevents unexpected visual results. A fixed or variable (supplied by the host) pattern may be applied to any drawing operation.

MIX	RESULT	
0h = 00000b =	~DST	
1h = 00001b =	0	
2h = 00010b =	1	
3h = 00011b =	DST	
4h = 00100b =	~SRC	
5h = 00101b =	SRC ^ DST	/* XOR */
6h = 00110b =	~ (SRC ^ DST)	
7h = 00111b =	SRC	
8h = 01000b =	~ (SRC * DST)	/* NAND */
9h = 01001b =	~ (SRC * ~DST)	
Ah = 01010b =	~ (~SRC * DST)	
Bh = 01011b =	~ (~SRC * ~DST)	/* OR */
Ch = 01100b =	SRC * DST	/* AND */
Dh = 01101b =	SRC * ~DST	
Eh = 01110b =	~SRC * DST	
Fh = 01111b =	~SRC * ~DST	/* NOR */
10h = 10000b =	min(SRC,DST)	
11h = 10001b =	DST – SRC	with underflow
12h = 10010b =	SRC – DST	with underflow
13h = 10011b =	SRC + DST	with overflow
14h = 10100b =	max(SRC,DST)	
15h = 10101b =	(DST – SRC)/ 2	with underflow
16h = 10110b =	(SRC – DST)/ 2	with underflow
17h = 10111b =	(SRC + DST)/ 2	with overflow
18h = 11000b =	DST – SRC	with saturate
19h = 11001b =	DST – SRC	with saturate
1Ah = 11010b =	SRC – DST	with saturate
1Bh = 11011b =	SRC + DST	with saturate
1Ch = 11100b =	(DST – SRC)/ 2	with saturate
1Dh = 11101b =	(DST – SRC)/ 2	with saturate
1Eh = 11110b =	(SRC – DST)/ 2	with saturate
1Fh = 11111b =	(SRC + DST)/ 2	with saturate

Source Operand Selection

The destination operand (DST) always comes from the bitmap. However, the source operand (SRC) can come from one of four places based on the Foreground or Background Source Select (FSS or BSS) fields. FSS and BSS are decoded identically, only FSS is shown below:

7	6	5	4	3	2	1	0
---	FSS		FOREMIX				

FSS	Foreground Source Select
0 0	Background Color (BKGD_COLOR)
0 1	Foreground Color (FRGD_COLOR)
1 0	Variable Pixel Data (PIX_TRANS)
1 1	Bitmap Data

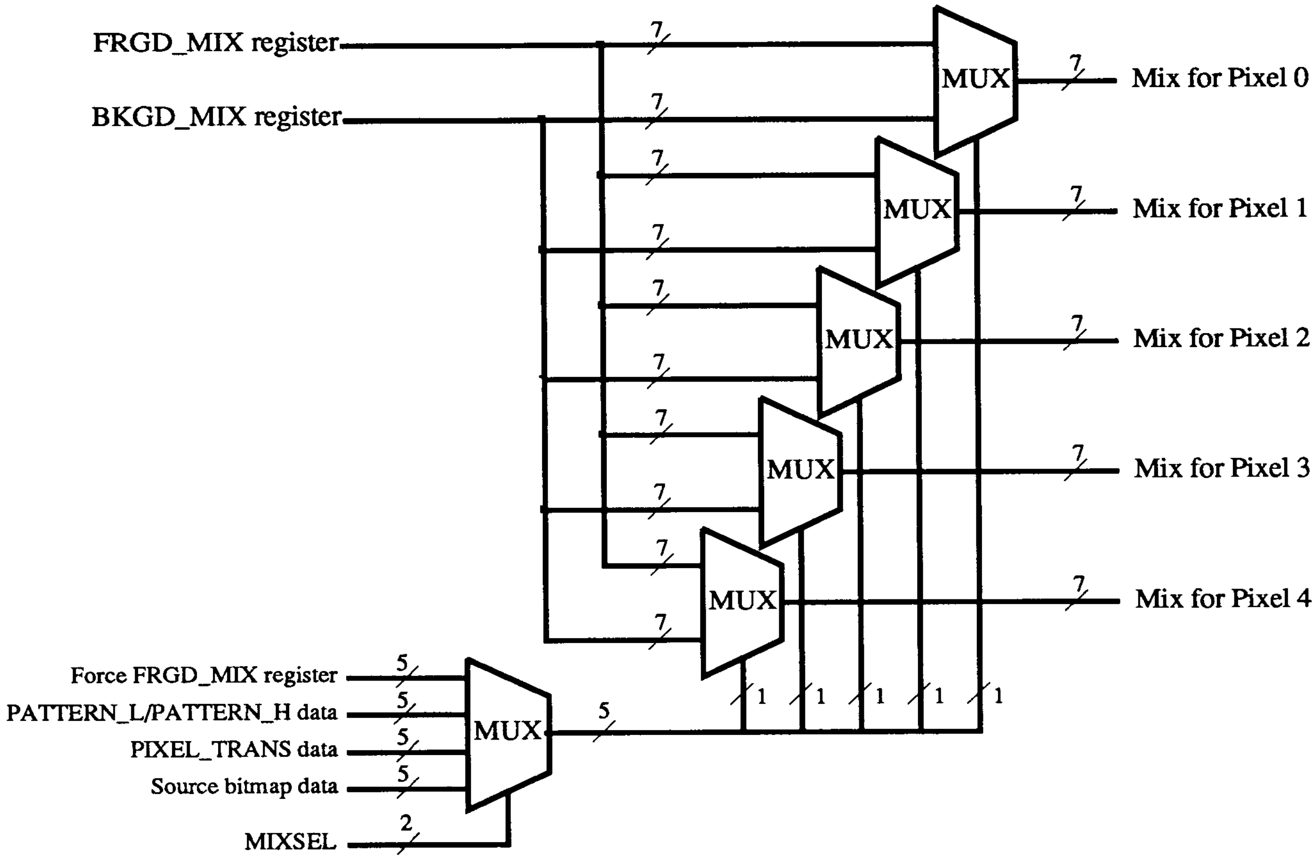
The Background Color (BKGD_COLOR) and Foreground Color (FRGD_COLOR) are register values, used to draw in a solid color or pattern of two solid colors.

Pixel Transfer (PIX_TRANS) is provided by/to the host, one byte per pixel or nugget (see Across Planes vs Through Planes, below). This allows applying any arbitrary pattern to a drawing operation, or copying data to/from system memory.

Bitmap data is from the source rectangle of a CopyRect command. This Source Select should only be used for the CopyRect command; results are undefined for other commands. Bitmap data is commonly referred to as "source" data, but should not be confused with "SRC" data, which is an ALU operand selected by either BSS or FSS. If FSS and BSS = 11, the terms "source" and "SRC" are equivalent.

Foreground and Background

The 82C480 supports two Source Select/Mix pairs: Foreground and Background. The Foreground Source Select (FSS) is always used with the Foreground Mix (FRGD_MIX), and the Background Source Select (BSS) is always used with the Background Mix (BKGD_MIX). The combination of Foreground Source Select and Foreground Mix will be referred to simply as the Foreground; the combination of Background Source Select and Background Mix will be referred to as the Background. There are also registers for the Foreground Color (FRGD_COLOR) and



Mix Select for each Pixel Drawing Engine

Background Color (BKGD_COLOR); these are not associated with the Foreground and Background mix. The Foreground Mix may select either Foreground Color or Background Color, and similarly for the Background Mix. It is observed that the AI associates the Foreground Color with the Foreground and the Background Color with the Background, but as far as the operation of the hardware is concerned, "Foreground Color" and "Background Color" are just words used to distinguish between the two color registers.

Whether the Foreground or Background is used is determined on a pixel-by-pixel basis; even though the Memory Controller operates on four or five pixels at a time, each pixel Arithmetic Logic Unit (ALU) can independently use the Foreground or Background.

Foreground Select Mode

In one further level of indirection, there are four possible methods of selecting Foreground vs. Background, selected by the mix select (MIXSEL) bits (PIX_CNTL[7:6]). MIXSEL applies to all ALUs.

Mix Select Mode (MIXSEL):

0 0	=	Foreground is always used (most common value)
0 1	=	PAT_L/PAT_H select Foreground (1 selects FRGD_MIX)
1 0	=	VAR (Variable data) selects Foreground (1 selects FRGD_MIX)
1 1	=	Transparency selects Foreground

If MIXSEL = 00 (most commonly used option), the Foreground is always selected.

If MIXSEL = 01, PATTERN_L (PAT_L) and PATTERN_H (PAT_H) select Foreground. PAT_L and PAT_H are both nugget-wide registers; PAT_L for even-numbered nuggets (starting with nugget 0 at the left of the screen), and PAT_H for odd-numbered nuggets. Bits (0),1, 2, 3, and 4 of PAT_L and PAT_H control pixels (4),3, 2, 1, and 0 of the nugget, respectively (5PN=1). These allow applying a fixed destination-aligned pattern. It may be a pattern of the Foreground and Background Color, or a pattern of Foreground and Background Mixes. To apply a 2D pattern, PAT_L and PAT_H must be reloaded on each scan line; there is no interrupt mechanism to prompt for reloading.

If MIXSEL =10, PIX_TRANS selects Foreground. PIX_TRANS is the variable data from the host. Bits 0,1, 2, 3, and 4 control pixels 4, 3, 2, 1, and 0 of each destination nugget respectively. This operation is described as 'across the planes'.

If MIXSEL = 11, transparency selects Foreground. Transparency is 1 (true) if source data = 1 for every bit where RD_MASK = 1. RD_MASK is rotated right one bit for purposes of transparency comparisons. This is to allow the RD_MASK to specify different planes for transparency and area filling operations. If transparency is true, the Foreground is selected; otherwise the Background is selected. Source data is stored in the scratch register during the bitblt (see next section); the result of the transparency test is stored in place of bit 7 of the source.

MIXSEL = 11 is used to allow "source transparency". Typically, one bit plane is designated as the transparency plane. If that bit is 1, the destination is left unmodified. Visually, the source data appears "transparent" for that pixel (i.e. after copying the source, you still see the old destination data at that pixel).

Should you select MIXSEL = 11 and FSS and/or BSS = 11 (SRC is bitmap data), you would expect to see the source data written to the destination at all the appropriate places. This is not quite what happens, because the foreground select bit overwrites bit 7 of the source, so instead of seeing the source written to the destination, you see the source with bit 7 modified.

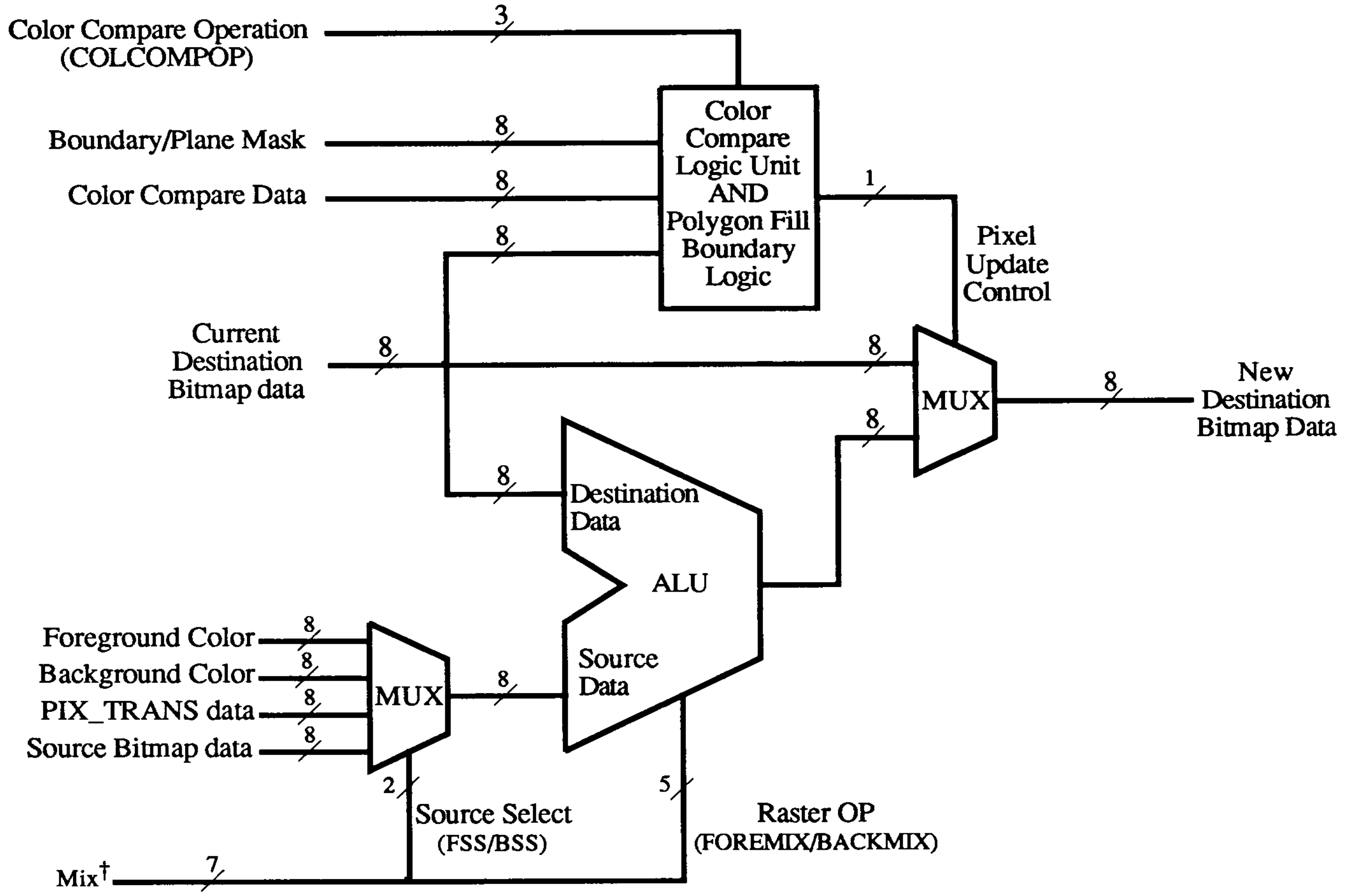
MIXSEL = 11 is also used to do a stretch blt--i.e., a bitblt where the source data is stored in only one plane, and during the course of the bitblt it is expanded to eight planes. This is done by selecting (for example) MIXSEL = 11, FSS = 01, FOREMIX = 7, BSS = 00, BACKMIX = 7, and RD_MASK = 01h. In this case, the source is a monochrome image stored in plane 7 (selected by RD_MASK>>1 = 80h); every pixel where bit 7 = 1, is drawn in Foreground Color, and every pixel where bit 7 = 0 is drawn in Background Color.

Stretch blts are used for drawing bitmapped characters. Character fonts may be cached with a different character in each plane to conserve memory, and then drawn in the current foreground color. By selecting the mixes, the character cell may be opaque (i.e. Foreground or Background Color) or transparent.

Color Comparison

The 82C480 can do a magnitude comparison of the destination data to a fixed reference color, stored in the Color Comparison register (COLOR_CMP). If the comparison is true, the destination is left unchanged; if false, the drawing operation operates normally. The effect is that the certain color(s) in the destination (i.e. "background") appear in front of the source (i.e. "foreground"). IBM refers to this operation as underpainting, meaning that the source pixels appear to be painted under the destination data.

Note that Color Comparison compares the destination data to a reference color; this is different than transparency (above), which tests the source data; the 82C480 does both.



Pixel Drawing Engine

† This Mix comes from one of the 5 individual pixel mixes shown in the diagram on the previous page.

The type of magnitude comparison between COLOR_CMP and DST is determined by the COLCMPOP bits of the Pixel Control register (PIX_CNTL[5:3]), as defined below:

COLCMPOP 5 4 3	Comparison	Notes
0 0 0	0	Always false
0 0 1	1	Always true
0 1 0	DST >= C_CMP	
0 1 1	DST < C_CMP	
1 0 0	DST ≠ C_CMP	Not equal
1 0 1	DST = C_CMP	
1 1 0	DST <= C_CMP	
1 1 1	DST > C_CMP	

By selecting the proper mixes, the Color Comparison logic can implement a Color Exchange operation: all pixels of a specified color are replaced by a different color. This Color Exchange will be done on a rectangle defined by the Clipping Rectangle, or on a Polygon, as defined in Polygon Fill.

Plane Masks

The 82C480 has two host-writeable 8-bit plane mask registers: Plane Read Mask (RD_MASK) and Plane Write Mask (WRT_MASK). The 82C480 generates a third, the Plane Mix Enable (PME) under control of the PLANEMODE bits (PIX_CNTL[2:1]). If PLANEMODE = 10, PME = WRT_MASK * ~RD_MASK; else PME = WRT_MASK.

A 0 in WRT_MASK always prevents writing the corresponding plane in memory, using the VRAM write-per-bit capability. A 0 in PME always prevents the corresponding planes from taking part in any mix (i.e. raster op); results are not specified for arithmetic mixes where a non-contiguous set of planes is enabled by PME.

The function of RD_MASK is more complicated:

During variable data read cycles, RD_MASK can determine which planes are read by the host depending on the value of the PLANEMODE bits. If PLANEMODE = 00, then the RD_MASK register has no effect on the data transfer. This is useful for reading the entire bitmap in Through Plane mode (PLANAR=0). If PLANEMODE = 10 or 11, the pixel data returned should be read back in Across Plane mode (PLANAR=1). The RD_MASK is rotated right one bit and a logical test is performed between the rotated RD_MASK and bitmap data. Those pixels which have 1s in all planes where rotated RD_MASK = 1 return a 1. Because the data is returned in Across Planes mode, the drawing engine is moving one nugget at a time and the data read back will be nugget aligned. This is useful for reading just one plane or color pattern.

In the read phase of a Copy Rectangle (CMD_BITBLT) command, when MIXSEL = 11 (Bitmap data selects Foreground), RD_MASK selects which planes to use to determine if the source is transparent. Those pixels which have 1s in all planes where RD_MASK = 1 are transparent. Plane 7 in the Scratch Register (see below) is used to store the transparency result, so only 128 colors are available when doing this operation.

In both cases above, RD_MASK is rotated right one bit before being used.

During Polygon Fill, the PLANEMODE bits determine the Boundary Mask. The Boundary Mask selects which plane(s) will describe the boundary of the polygon. Pixels which have 1s in all planes where RD_MASK = 1 are boundary pixels. In this case, RD_MASK is not rotated. RD_MASK bit Functions:

	Enabled Plane	
RD MASK	Variable Data Read	Polygon Fill
0000 0001b	7	0
0000 0010b	0	1
0000 0100b	1	2
0000 1000b	2	3
0001 0000b	3	4
0010 0000b	4	5
0100 0000b	5	6
1000 0000b	6	7

The following table shows the mask values for the two area fill operations:

PIX_CNTL [2:1]	Boundary Mask	Plane Mask
1 0	RD_MASK	WRT_MASK * (~RD_MASK)
1 1	WRT_MASK	WRT_MASK

Across Planes vs. Through Planes

In Pixel Data write mode (PCDATA = WRTDATA = 1), the 82C480 will stop before drawing each pixel (or nugget) for the host to write data in the Pixel Data register (PIX_TRANS). In Pixel Data read mode (PCDATA = 1, WRTDATA = 0), the 82C480 will stop after reading each pixel (or nugget) for the host to read data from PIX_TRANS. PCDATA = 1 applies to all drawing operations, including lines and short stroke vectors. This allows drawing lines using any arbitrary user-defined line pattern.

When PCDATA = 1, Pixel Data may be read/written across planes (AP) or through planes (TP).

In TP mode (PLANAR = 0), PIX_TRANS is interpreted as an 8-bit value representing all eight planes of a single pixel. When doing bitblts in this mode, the 82C480 handles data just one pixel at a time, not one nugget at a time.

In TP pixel data read mode (PLANAR = 0, PCDATA = 1, WRTDATA = 0), the drawing engine waits to write the source pixel to PIX_TRANS before moving on to the next pixel. If necessary, it will wait for the host to read the last pixel from PIX_TRANS.

In TP pixel data write mode (PLANAR = 0, PCDATA = WRTDATA = 1), the drawing engine waits to receive a new PIX_TRANS byte from the queue before drawing each pixel. What effect PIX_TRANS has on the pixel operation is determined by the mix select (MIXSEL) field (PIX_CNTL[7:6]), and the mix registers (FRGD_MIX and BKGD_MIX). It is entirely possible that PIX_TRANS has no effect on the pixel operation at all, but the drawing engine still must receive pixel data from the host before it will proceed.

In AP mode (PLANAR = 1), PIX_TRANS is interpreted as a four/(five) bit quantity, where each bit controls one pixel. PIX_TRANS bits (0),1, 2, 3, and 4 control pixels (4),3, 2, 1, and 0, respectively. WRT_MASK can be used to control which bit planes are modified. AP mode should not be used with line draw and SSV commands, because these commands inherently operate on pixels, while AP mode works on nuggets, and results are difficult to predict.

In AP pixel data read mode (PLANAR = PCDATA = 1, WRTDATA = 0), PIX_TRANS bits (0)/1 through 3 contain the transparency result of pixels (4)/3 through 0 of the last nugget read. The Plane Read Mask (RD_MASK) register can be used to select which planes participate in the transparency test.

In AP pixel data write mode (PLANAR = PCDATA = WRTDATA = 1), each of PIX_TRANS bits (0)/1 through 3 selects foreground/background for pixels (4)/3 through 0 of the current nugget if MIXSEL =10. If MIXSEL ≠ 2, PIX_TRANS has no effect on pixel operations, but the drawing machine still waits for a new PIX_TRANS byte before proceeding.

To improve performance in either AP or TP modes, PIX_TRANS data may be read/written two nuggets or pixels at a time by selecting 16-bit mode (_16BIT = 1). In this mode, PIX_TRANS is treated as a 16-bit register, and needs to be read/written only half as often.

Also to improve performance, writes to PIX_TRANS are queued; this allows writing up to 8 pixels (16 if _16BIT = 1) in advance. Reads from PIX_TRANS are not queued. See separate Queue section for details of Queue operation.

Scratch Register

During the source phase of a Copy Rectangle (CMD_BITBLT) command, up to eight nuggets are read into the Scratch Register. Nuggets are read using page mode (bank-interleaved page mode if there are two or four banks). Pixel alignment is done as the data is written into the scratch register. If the source is nugget-aligned, all reads are done in nugget mode; if the source is not nugget-aligned, the edges are read in pixel mode, and the interior is read in nugget mode.

During the destination phase of the Copy Rectangle command, data is read from the scratch register, combined with the destination data as specified by the FOREMIX or BACKMIX, and written back to the destination address. If the Mix is a function of the destination data, page-mode RMW cycles are used, otherwise simple page-mode write cycles are used. If the destination is nugget-aligned, all writes are done in nugget mode; otherwise the edges are written in pixel mode and the interior of the rectangle is written in nugget mode.

82C480 Drawing Operations

The 82C480 supports 6 different drawing algorithms, including 2 for lines and 4 for rectangles. These algorithms define the motion from the current XY position, and are executed in the "drawing engine" circuitry. Other circuitry, the ALU, selects whether certain pixels or image planes are written, which color is written, etc. Marking of pixels takes place only inside the scissor rectangle; the drawing operation proceeds even outside the rectangle, but WE4:0/ are inhibited so no pixels are marked. (The scissor rectangle is defined by the XY coordinates programmed in the four SCISSORS registers).

The choice between Through Planes and Across Planes modes is important for the execution of the drawing algorithms. In general, TP mode is needed only to copy data between system memory and the bitmap. TP mode is relatively slow, requiring that each pixel be accessed in its own VRAM cycle. Most algorithms work faster in AP mode, since a whole nugget (32 or 40 bits) may be accessed in a single VRAM cycle.

The drawing engine can take advantage of AP mode by invoking "nugget" mode. In this mode, the current X coordinate (CUR_X) can be moved a whole nugget in one machine cycle (a "machine cycle" is the basic cycle of the geometry state machine). "Pixel" mode requires 4 machine cycles for a nugget (even though the entire nugget could be written in one VRAM cycle). Some algorithms (Fast Fill and the Source pass of the Copy Rectangle) assume nugget mode, whereas others invoke it only when appropriate.

The selection of Pixel Data, using the PCDATA bit, controls the geometry engine only. It causes VRAM accesses to be limited by the transfer rate of Pixel Data across the system bus. That is, one byte of Pixel Data enables 1 VRAM access (either pixel or nugget!). PCDATA does not control whether Pixel Data is used as an ALU control or data source! The Mix (FRGD_MIX, BKGD_MIX) and Pixel Control (PIX_CNTL) registers control these aspects.

Note that AP Pixel Data mode for Bresenham lines is tricky if not impossible to implement. This is because each new byte of Pixel Data triggers one VRAM memory cycle. Since every Bresenham "bump" to a new scan line requires a new memory cycle, and since it is difficult to determine how many pixels will appear on each scan line, it is difficult to synchronize AP Pixel Data bytes with Bresenham drawing. Instead use TP mode.

Command Port bits INC_X and INC_Y effect the starting corner of the rectangle/line. If neither is set, the starting corner is the lower right. If INC_Y is set and INC_X is not, drawing starts in the upper right corner. All algorithm discussions below assume both bits are set, and drawing starts in the upper left corner.

Line

The 82C480 Line command uses the Bresenham algorithm. Bresenham, like most Digital Differential Analysis (DDA) algorithms, first determines the major or independent axis (if the line were projected onto the X and Y-axis, the independent axis is the one with the longer projection). The major axis coordinate (X or Y) is the independent coordinate. The slope of the line is calculated for the other (dependent) coordinate as a function of the independent coordinate. For each pixel drawn the algorithm increments (or decrements) the independent coordinate, and adds the slope to the dependent coordinate, and if the fractional remainder (the error term) is greater than 0.5, the dependent coordinate is incremented; otherwise it stays the same.

The method above requires a division to compute the slope. This division can be avoided, however, since the algorithm is unaffected by multiplying by a constant.

The algorithm may be expressed by the C pseudo-code on the following page. The major axis is indicated by the 82C480 Y Major Axis (YMAJAXIS) bit; if YMAJAXIS = 1, Y is the major axis. Before drawing the line, the image space is decomposed into eight octants, and the octant in which the line is drawn is encoded by the INC_Y, YMAJAXIS, and INC_X bits (CMD[7:5]).

SSV Drawing

Short Stroke Vector (SSV) is another line drawing function. SSV involves much less overhead than Line Draw (CMD_LINE); only one byte is required per vector, compared to four (or more) 16-bit words per vector if drawn by the Line command. Two SSVs may be generated with a single 16-bit write to the SSV Register. This lends itself well for character/text drawing.

```
/* 82C480 Implementation of Bresenham algorithm.

   The following registers must be initialized by software before issuing a CMD_LINE command:

     CUR_X              =    Initial x-axis coordinate
     CUR_Y              =    Initial y-axis coordinate
     MAJ_AXIS_PCNT      =    Length of line projected onto major axis (dmajor)
     DESTX_DIASTP       =    2*dminor – 2*dmajor
     DESTY_AXSTP        =    2*dminor
     ERROR_TERM         =    2*dminor – dmajor

        Where:
                dminor  =    length of minor axis projection
                dmajor  =    length of major axis projection
   ____________________________________________________________________

     CMD_LINE()
     {
     int   xstep,ystep;
     if    (INC_Y)  {ystep = 1}  else  {ystep = –1}
     if    (INC_X)  {xstep = 1}  else  {xstep = –1}
     for (i=0; i < MAJ_AXIS_PCNT; i++)
         {
         plot(CUR_X,CUR_Y);        /* Plot the pixel at the current x,y position */
         if (ERROR_TERM > 0)
             {
             CUR_X  += xstep;      /* Make a diagonal step */
             CUR_Y  += ystep;
             ERROR_TERM  +=  DESTX_DIASTP;
             }
         else
             {                     /* Make a linear step */
             if (YMAJAXIS)  {CUR_Y  +=  ystep}  else  {CUR_X  +=  xstep};
             ERROR_TERM  +=  DESTY_AXSTP;
             }
         }
     }
```

When using SSVs, the Command Register must first be set for no drawing command (CMD_NOP) and Short Stroke Vectors Enabled (LINETYPE = 1). LASTPIX and BYTSEQ Command Register bits are used by the SSV command. After the Command Register is set, a series of SSV drawing orders may be written to the SHORT_STROKE Register. This series may be padded with Null SSV bytes (all 0's).

Each SSV specifies a length (0 to 15 pixels), a direction (multiple of 45°), and a bit to indicate whether the SSV is drawn (visible) or not (invisible).

In all SSVs, the Current Position (CUR_X,CUR_Y) is moved from the starting position to a position which is N pixels (Length) away from the starting position, in the direction specified. A length of "0" will cause no movement. If drawing is not specified (SSVDRAW=0), no writes to memory are performed.

If drawing is specified, writes to memory take place as the current position is moved. The first pixel is always drawn at the Current Position. If LASTPIX is set, N pixels are written. If LASTPIX is not set, N + 1 pixels are written. If LASTPIX = 1 and LENGTH = 0, one pixel is drawn at the Current Position.

See the SSV register description for more details.

Rectangle

The 82C480 has one form of bitblt called Copy Rectangle (CMD_BITBLT), and three forms of rectangle fill called Fast Rectangle (CMD_RECTV2), Rectangle X (CMD_RECT), and Rectangle Y (CMD_RECTV1). Each of these is selected by the Draw Command bits in the Command Register CMD[15:13].

Copy Rectangle

The Copy Rectangle (CMD_BITBLT) command is a 2-pass algorithm used for copying one region of image bitmap memory into another in foreground or background.

In TP mode, it may also be used to conditionally blit system memory data into the Destination rectangle, contingent upon Source rectangle data. This setup would require PLANAR = 0 and MIXSEL = 11. This is not a standard usage of Through Plane mode according to IBM. This usage implies switching between Foreground and Background Mixes (based on Source data), whereas IBM claims TP mode uses only the Foreground Mix!

The first pass of the algorithm reads data from the Source image bitmap rectangle into the scratch register. The motion algorithm used for reading is identical to the Rectangle X algorithm in Across Plane mode (below) taking advantage of Nugget mode when possible. During this pass, pixel data is realigned through a barrel shifter into its Destination Intra-Nugget Pixel Position (INPP) alignment. The ALU is not used in this pass. If Command register DRAW bit (CMD[4]) is clear, Pixel Data may be read from the Source rectangle during this pass, but Across Planes is the only viable format. Data read this way is in the original INPP alignment, not realigned.

The second pass of the algorithm writes data from the scratch register into the Destination rectangle. The ALU is used to mix the Destination data with Pixel Data (if TP = WRTDATA = 1), Fixed Data, or Source Data, as chosen by the Mix. The algorithm used is identical to Rectangle X, using either Across or Through Plane mode. Across Plane mode is the fastest mode, using 1 bit-per pixel Variable Data (if AP = WRTDATA = 1), Pattern Register Data, or Source Data as the Mix select. The Color Comparators may always be used to control Underpaint.

Fast Rectangle

CMD_RECTV2 uses a left-to-right alternating up-down sweep algorithm, and always draws a nugget at a time where possible. This is not the fastest rectangle command, but it is the faster of the two Y–direction rectangle commands.

If the left edge of the rectangle is nugget-aligned, drawing starts in nugget mode. Otherwise, drawing starts as follows: drawing starts in pixel mode at the upper-left corner of the rectangle. One pixel at a time is drawn, going to the right until a nugget boundary is reached; a maximum of three pixels (four for five-pixel nuggets) are drawn this way. Drawing proceeds downwards, drawing one partial nugget at a time until the bottom of the rectangle is reached. The write enables are used to write only the pixels of the nugget that fall inside the rectangle. The Current Position moves right to the next nugget, and drawing proceeds upwards in nugget mode. Drawing continues in nugget mode, alternating up and down directions, until the right edge of the rectangle to be drawn is less than a nugget wide.

If the right edge of the rectangle is nugget-aligned, drawing ends in nugget mode. Otherwise, the right edge is drawn exactly as described above for the left edge; one pixel at a time, until the right edge of the rectangle is reached.

The pixel-mode operations at the left and right edges seem to be required because the 8514/A lacks the logic to detect if the left and right edges of a rectangle both fall within the same nugget.

Fast Rectangle assumes Across Plane operation. The selection of Across Plane or Through Plane modes by the PLANAR Command register bit is ignored.

X Rectangle

The Rectangle X (CMD_RECT) drawing command is used for drawing or reading a rectangle in a horizontal orientation. This may be used in either TP or AP modes. In TP mode, it may be used most effectively for copying Variable Data between system memory and image memory. In AP mode, it is the fastest drawing command for large area fills, since it makes optimum use of page mode. It is also the drawing command used for area fills and floods (see below).

Motion starts in the upper left-hand corner of the rectangle (assuming INC_X and INC_Y are set in the command register), and proceeds to the right. Once the right end of the rectangle is reached, the Y position is incremented, and X position is reset to the left edge of the rectangle. Motion then repeats as in the first row, continuing until the whole rectangle is drawn.

When a full nugget (4/5 adjacent pixels aligned on a nugget boundary) can be written, this drawing algorithm will take full advantage of "nugget mode". All 4/5 pixels will be drawn by the drawing engine in one machine cycle. Otherwise, pixels in less than full nuggets (e.g. at the left and right edges or rectangles) will be drawn in "pixel mode", accumulating Write Enables one pixel at a time.

Nugget mode is never used in TP mode, because TP requires one memory cycle per pixel.

Y Rectangle

Rectangle Y (CMD_RECTV1) is used for drawing or reading a rectangle one pixel at a time in a vertical orientation. This type of drawing motion is useful for TP data manipulation, particularly copying Pixel Data between the system memory and bitmap when the data is vertically oriented. While it may be used in AP mode, it offers no speed advantage. Motion starts in the upper left-hand corner of the rectangle (assuming INC_X and INC_Y are set in the Command register) and proceeds downward in a 1-pixel-wide column. Once at the bottom of the rectangle, the X position is moved right by one pixel, and the Y position is reset to the top of the rectangle. This motion repeats for all columns, and the process continues until the whole rectangle is drawn.

Polygon Fill

The 82C480 contains hardware to perform Polygon Fills using the CMD_RECT command with the proper PLANEMODE bit values. Polygon Fill is defined as filling an arbitrary area using any available mix, where the area is defined by a continuous boundary of the specified boundary color(s). The area may be concave or contain holes, and is filled using an "odd/even" algorithm. The area is filled by setting up a destination rectangle which encloses the polygon, setting PLANEMODE = 1Xb, and performing a CMD_RECT command. The 82C480 fills the polygon in AP mode, four pixels at a time, at full speed.

Inside detection is done by defining the left edge of the destination rectangle as "outside". For each scan line, each time a pixel of the boundary color is crossed, the inside/outside state is toggled. The 82C480 can handle multiple inside/outside transitions per nugget.

There are two methods of defining the boundary color(s), selected by PLANEMODE bit 0:

If PLANEMODE[0] = 0, the boundary is defined to be those pixels having 1s in all bit positions where there are 1s in the Plane Read Mask (RD_MASK). The Plane Mask applied during the fill operation is:

Plane Mask = WRT_MASK * ~(RD_MASK)

Usually, a single plane would be used to mark the boundary, and if this plane is masked off from the palette, the boundary would be invisible; this limits the system to 128 colors. When filling the polygon, the left boundary is marked but not the right; this allows producing a filled polygon that is only one pixel wide.

If PLANEMODE[0] = 1, the boundary is defined to be those pixels having 1s in all bit positions where there are 1s in the Plane Write Mask (WRT_MASK) register. For example, if WRT_MASK = FFh, the boundary would be marked in color FFh. The Plane Mask applied during the fill operation is:

Plane Mask = WRT_MASK

When filling the polygon, both the right and left boundaries are overwritten with the selected color. The advantage of this method is it does not require a separate plane to mark the boundary, so it is the only one that works with a 256 color system.

To facilitate drawing outlines of areas, the 82C480 provides an Outline command. This is similar to the Line command, but after drawing the first pixel of the line, it draws a pixel only when the Y coordinate increments. This produces a boundary of the enclosed area which is compatible with the Area Fill algorithm. Area filling may produce the wrong results if the boundary is self-intersecting. Using XOR mix with the Outline command will eliminate some errors, but not all. Careful orientation of the outline commands can ensure a correct result.

There is a variation of the area fill function which may be called the Fill/BitBlt. This allows a BitBlt to be done to the interior of a polygon. The Fill/BitBlt uses the RD_MASK and WRT_MASK exactly the same as the standard Fill operation.

Text Modes

There is no hardware text mode as in the VGA. The AI supports two types of fonts: bitmapped fonts and stroke fonts. Bitmap fonts are stored in system memory as bitmaps, and stroke fonts are stored as a sequence of SSVs. The AI maintains a most-recently-used cache of four fonts of either type in a non-displayed area of the bitmap. If the character is in the cache, it can be drawn by just blitting; if it is not cached, the AI takes care of generating an image of the character and adding it to the cache.

This page was intentionally left blank

Application Schematic Examples

This section includes schematic examples showing how to connect the 82C480 chip. The schematics are broken down into three main groups for discussion:

1) System Bus Interface

82C480 PC/AT (ISA) Bus Circuit Example

Interfacing the 82C480 to the PC Bus requires buffering of the data bus (74LS245) and optional connection of an EPROM/ROM. All other connections between the PC Bus and the 82C480 are direct and do not require any additional components. If a 32K ROM is used and paging into an 8K address space is desired, then the Monitor ID bits must be driven onto pins MS2:0 by a 74LS244 or 74LS125.

82C480 Micro Channel Bus Circuit Example

Interfacing the 82C480 to the Micro Channel requires buffering of the data bus (74LS245) and optional connection of an EPROM/ROM. If a ROM is used, the ROM addresses need to be latched using 74LS373 octal latches or equivalent. All other connections between the PC Bus and the 82C480 are direct and do not require any additional components. If a 32K ROM is used and paging into an 8K address space is desired, then the Monitor ID bits must be driven onto pins MS2:0 by a 74LS244 or 74LS125.

2) Display Memory Interface

82C480 Memory Configuration A

This is the memory configuration which is most commonly implemented for the lower resolutions supported by the 82C480 (1280x1024 or below). In this configuration, the memory banks are interleaved horizontally and share the same 32/40 bit serial data path.

82C480 Memory Configuration B

This memory configuration is used in high resolution implementations. It allows the highest possible serial data output (64/80 bit per serial clock). This memory configuration is not compatible with the 82C484 support chip.

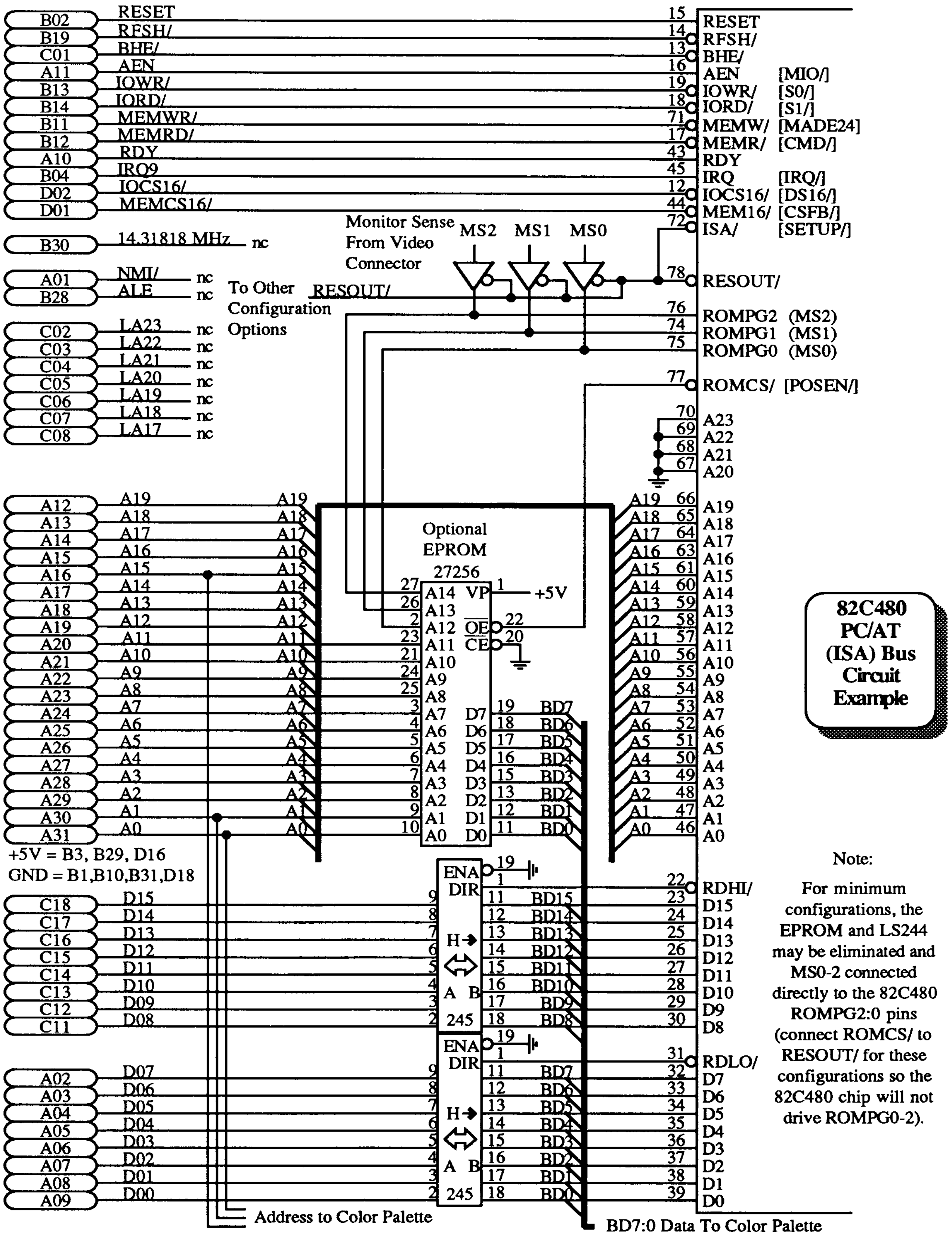
3) Video Interface

82C480 Video Interface Using an InMOS IMSG176/178 Color RAMDAC

The IMSG176 is used on the IBM 8514/A adapter card. It has a triple 6-bit DAC output and requires a current reference. An IMSG178 may be used to obtain the extended color definition offered by a triple 8-bit DAC output. An LM339 is required for Monitor Sense. Applications implementing 5-pixel nuggets must use a 74F374 to multiplex the 8514 and VGA video data paths.

82C480 Video Interface Using a Brooktree Bt471/478 Color RAMDAC

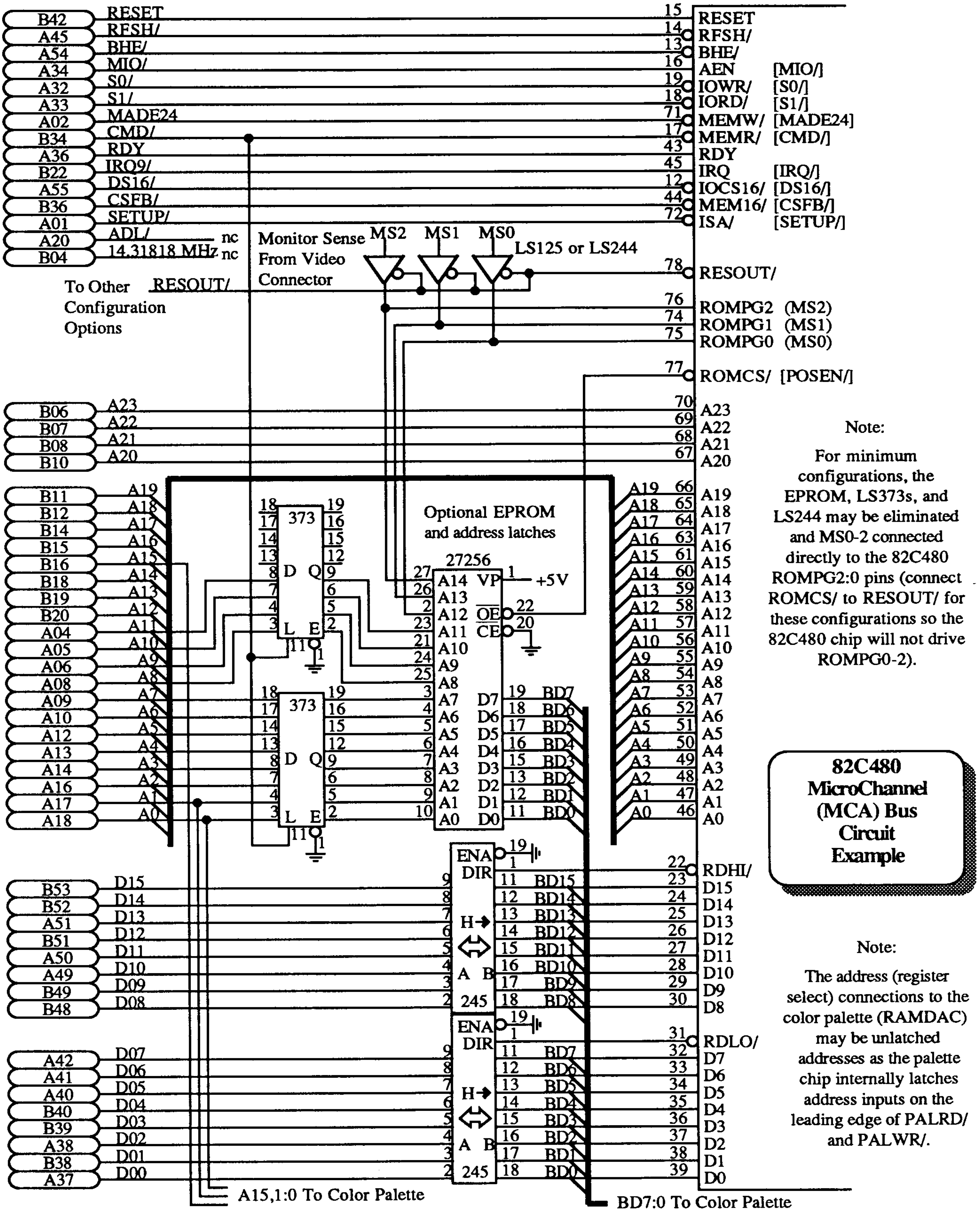
The Bt471/478 are similar in function to the IMSG176/178. The Bt471/478 are shown using a voltage reference (44-pin PLCC). An LM339 is required for Monitor Sense unless Bt475/477 RAMDACs are used. These RAMDACs implement the voltage comparators internally eliminating the need for the LM339. Applications implementing 5-pixel nuggets must use a 74F374 to multiplex the 8514 and VGA video data paths.


B02 RESET 15 RESET
B19 RFSH/ 14 RFSH/
C01 BHE/ 13 BHE/
A11 AEN 16 AEN [MIO/]
B13 IOWR/ 19 IOWR/ [S0/]
B14 IORD/ 18 IORD/ [S1/]
B11 MEMWR/ 71 MEMW/ [MADE24]
B12 MEMRD/ 17 MEMR/ [CMD/]
A10 RDY 43 RDY
B04 IRQ9 45 IRQ [IRQ/]
D02 IOCS16/ 12 IOCS16/ [DS16/]
D01 MEMCS16/ 44 MEM16/ [CSFB/]
72 ISA/ [SETUP/]
B30 14.31818 MHz nc
A01 NMI/ nc
B28 ALE nc
C02 LA23 nc
C03 LA22 nc
C04 LA21 nc
C05 LA20 nc
C06 LA19 nc
C07 LA18 nc
C08 LA17 nc
Monitor Sense From Video Connector
MS2 MS1 MS0
To Other Configuration Options
RESOUT/
78 RESOUT/
76 ROMPG2 (MS2)
74 ROMPG1 (MS1)
75 ROMPG0 (MS0)
77 ROMCS/ [POSEN/]
70 A23
69 A22
68 A21
67 A20
A12 A19, A13 A18, A14 A17, A15 A16, A16 A15, A17 A14, A18 A13, A19 A12, A20 A11, A21 A10, A22 A9, A23 A8, A24 A7, A25 A6, A26 A5, A27 A4, A28 A3, A29 A2, A30 A1, A31 A0
Optional EPROM 27256
27 A14, 26 A13, 2 A12, 23 A11, 21 A10, 24 A9, 25 A8, 3 A7, 4 A6, 5 A5, 6 A4, 7 A3, 8 A2, 9 A1, 10 A0
VP 1 +5V
OE 22
CE 20
D7 19 BD7, D6 18 BD6, D5 17 BD5, D4 16 BD4, D3 15 BD3, D2 13 BD2, D1 12 BD1, D0 11 BD0
66 A19, 65 A18, 64 A17, 63 A16, 61 A15, 60 A14, 59 A13, 58 A12, 57 A11, 56 A10, 55 A9, 54 A8, 53 A7, 52 A6, 51 A5, 50 A4, 49 A3, 48 A2, 47 A1, 46 A0
+5V = B3, B29, D16
GND = B1,B10,B31,D18
C18 D15, C17 D14, C16 D13, C15 D12, C14 D11, C13 D10, C12 D09, C11 D08
ENA 19, DIR 1, H→, ⇔, A B, 245
9 8 7 6 5 4 3 2
11 BD15, 12 BD14, 13 BD13, 14 BD12, 15 BD11, 16 BD10, 17 BD9, 18 BD8
22 RDHI/
23 D15, 24 D14, 25 D13, 26 D12, 27 D11, 28 D10, 29 D9, 30 D8
A02 D07, A03 D06, A04 D05, A05 D04, A06 D03, A07 D02, A08 D01, A09 D00
11 BD7, 12 BD6, 13 BD5, 14 BD4, 15 BD3, 16 BD2, 17 BD1, 18 BD0
31 RDLO/
32 D7, 33 D6, 34 D5, 35 D4, 36 D3, 37 D2, 38 D1, 39 D0
Address to Color Palette
BD7:0 Data To Color Palette

82C480 PC/AT (ISA) Bus Circuit Example

Note:

For minimum configurations, the EPROM and LS244 may be eliminated and MS0-2 connected directly to the 82C480 ROMPG2:0 pins (connect ROMCS/ to RESOUT/ for these configurations so the 82C480 chip will not drive ROMPG0-2).



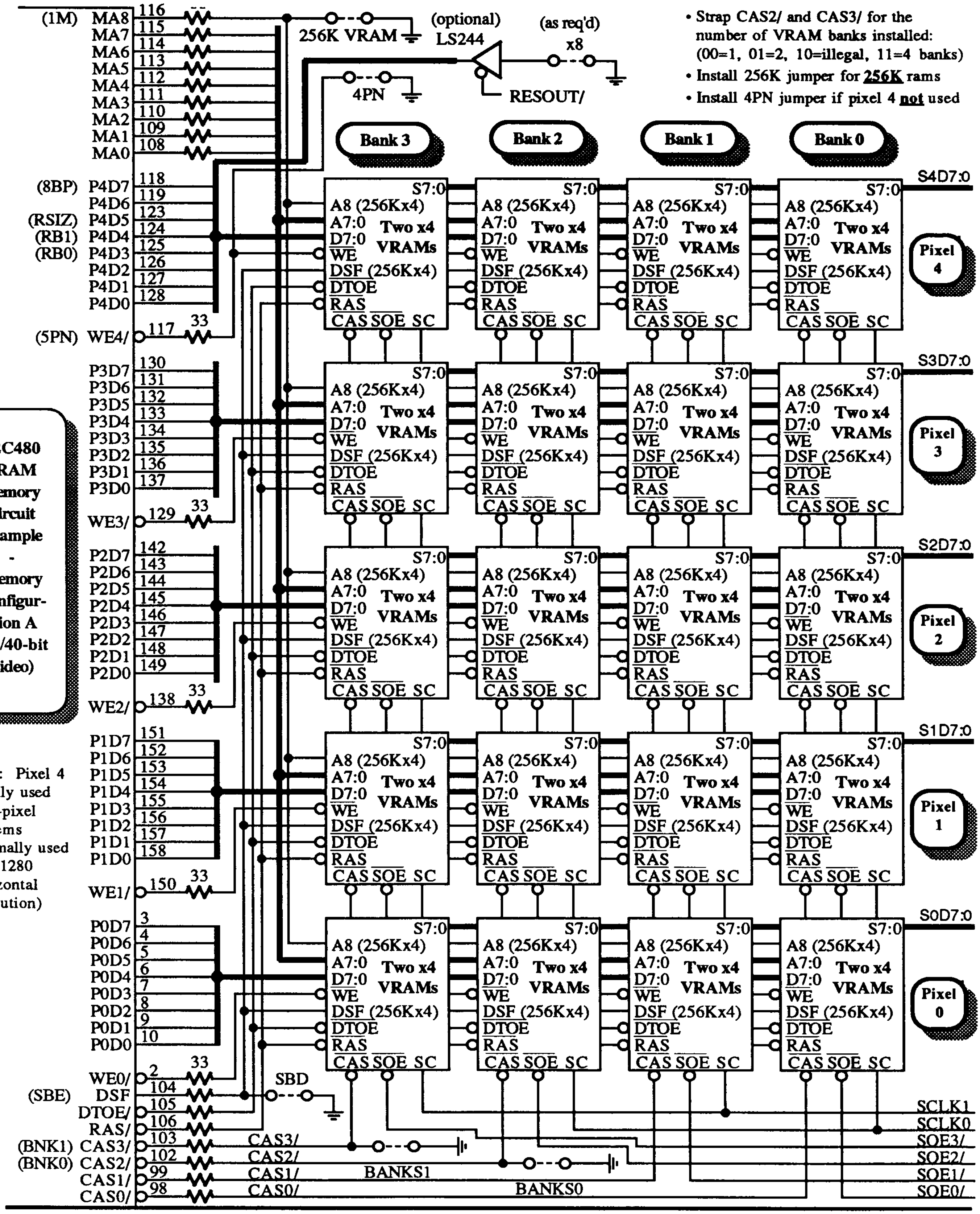
Note:

For minimum configurations, the EPROM, LS373s, and LS244 may be eliminated and MS0-2 connected directly to the 82C480 ROMPG2:0 pins (connect ROMCS/ to RESOUT/ for these configurations so the 82C480 chip will not drive ROMPG0-2).

82C480 MicroChannel (MCA) Bus Circuit Example

Note:

The address (register select) connections to the color palette (RAMDAC) may be unlatched addresses as the palette chip internally latches address inputs on the leading edge of PALRD/ and PALWR/.



82C480 VRAM Memory Circuit Example - Memory Configuration A (32/40-bit Video)

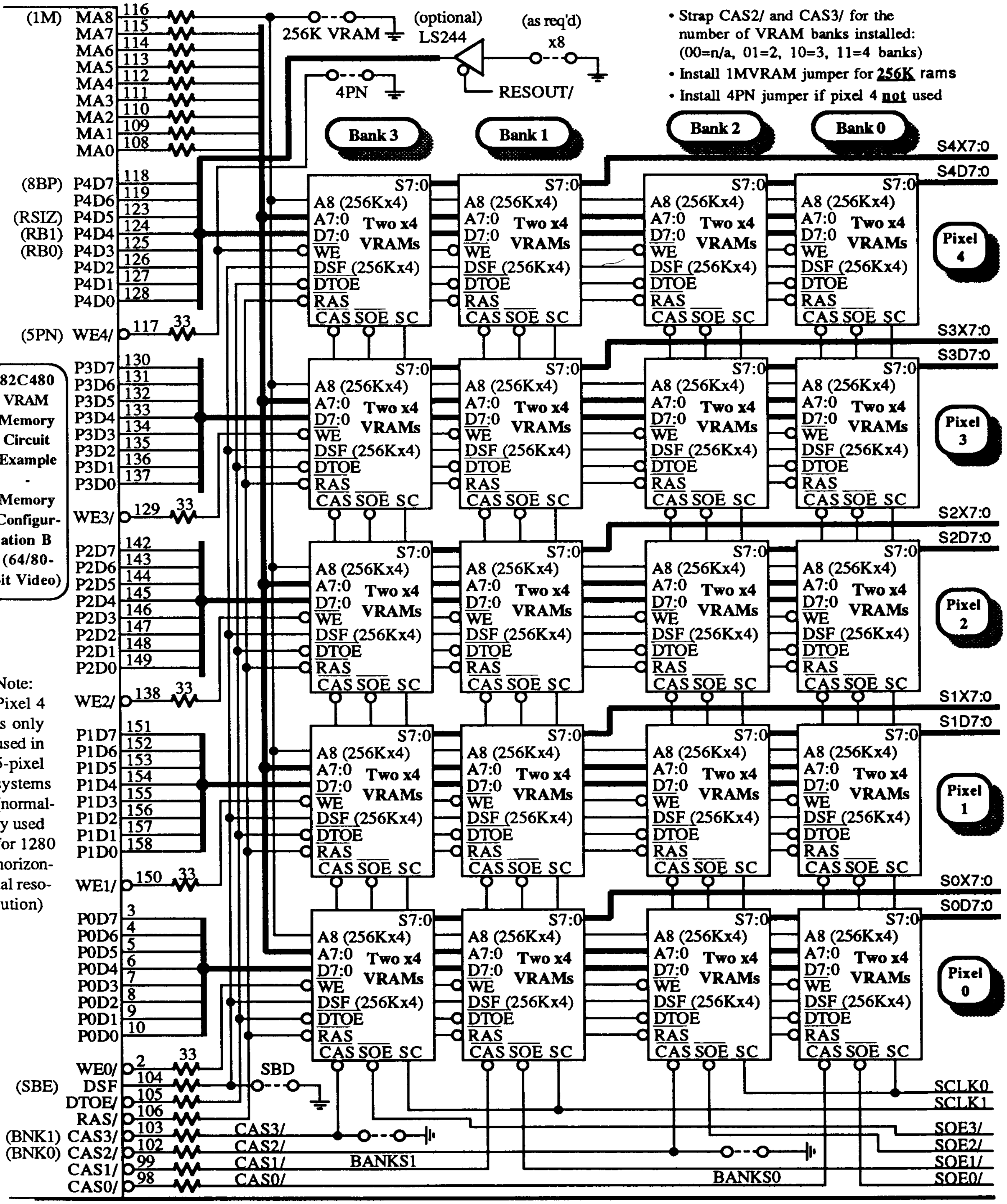
- Strap CAS2/ and CAS3/ for the number of VRAM banks installed: (00=1, 01=2, 10=illegal, 11=4 banks)
- Install 256K jumper for **256K** rams
- Install 4PN jumper if pixel 4 **not** used

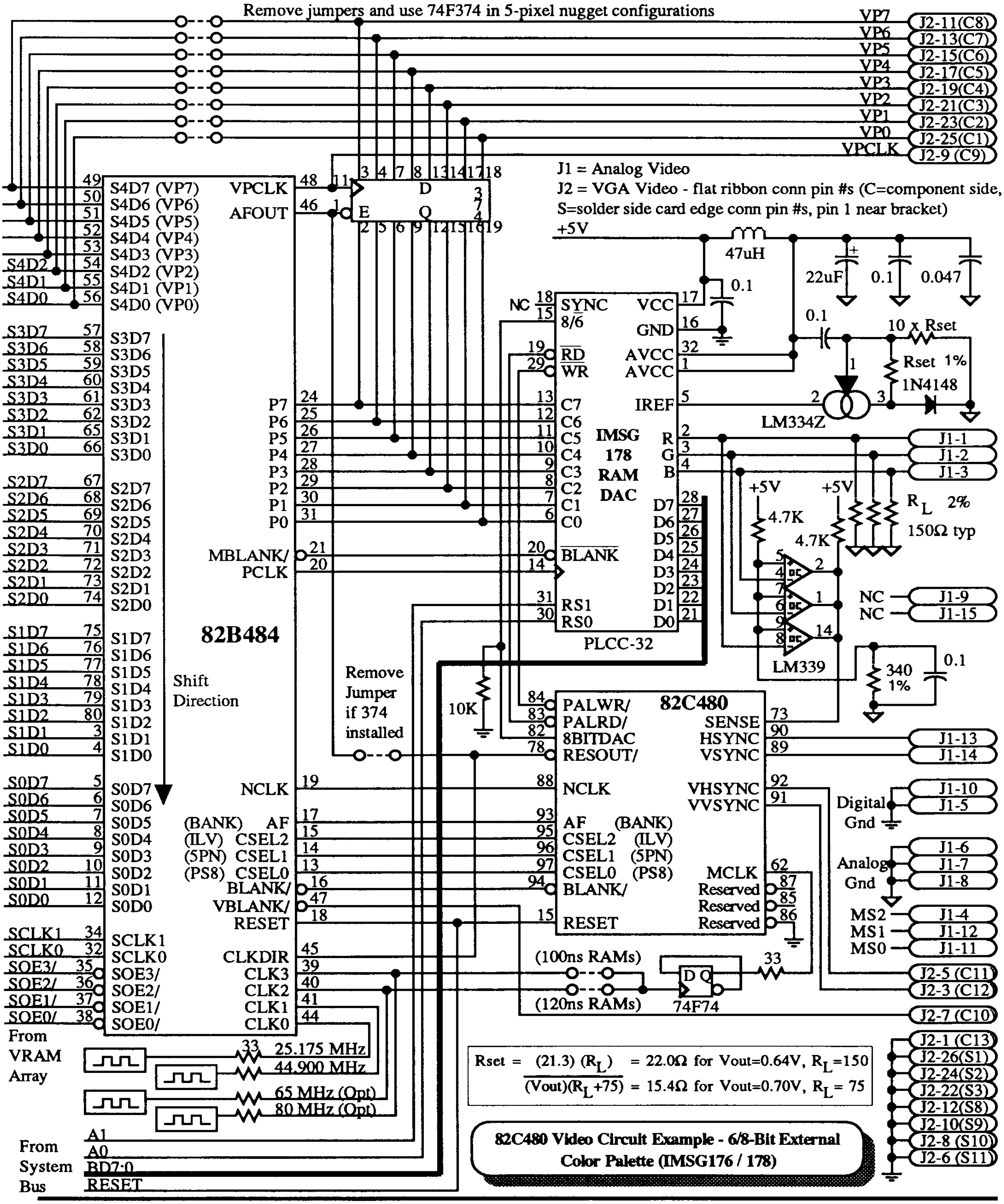
Note: Pixel 4 is only used in 5-pixel systems (normally used for 1280 horizontal resolution)

82C480 VRAM Memory Circuit Example - Memory Configuration B (64/80-bit Video)

- Strap CAS2/ and CAS3/ for the number of VRAM banks installed: (00=n/a, 01=2, 10=3, 11=4 banks)
- Install 1MVRAM jumper for **256K** rams
- Install 4PN jumper if pixel 4 **not** used

Note: Pixel 4 is only used in 5-pixel systems (normally used for 1280 horizontal resolution)



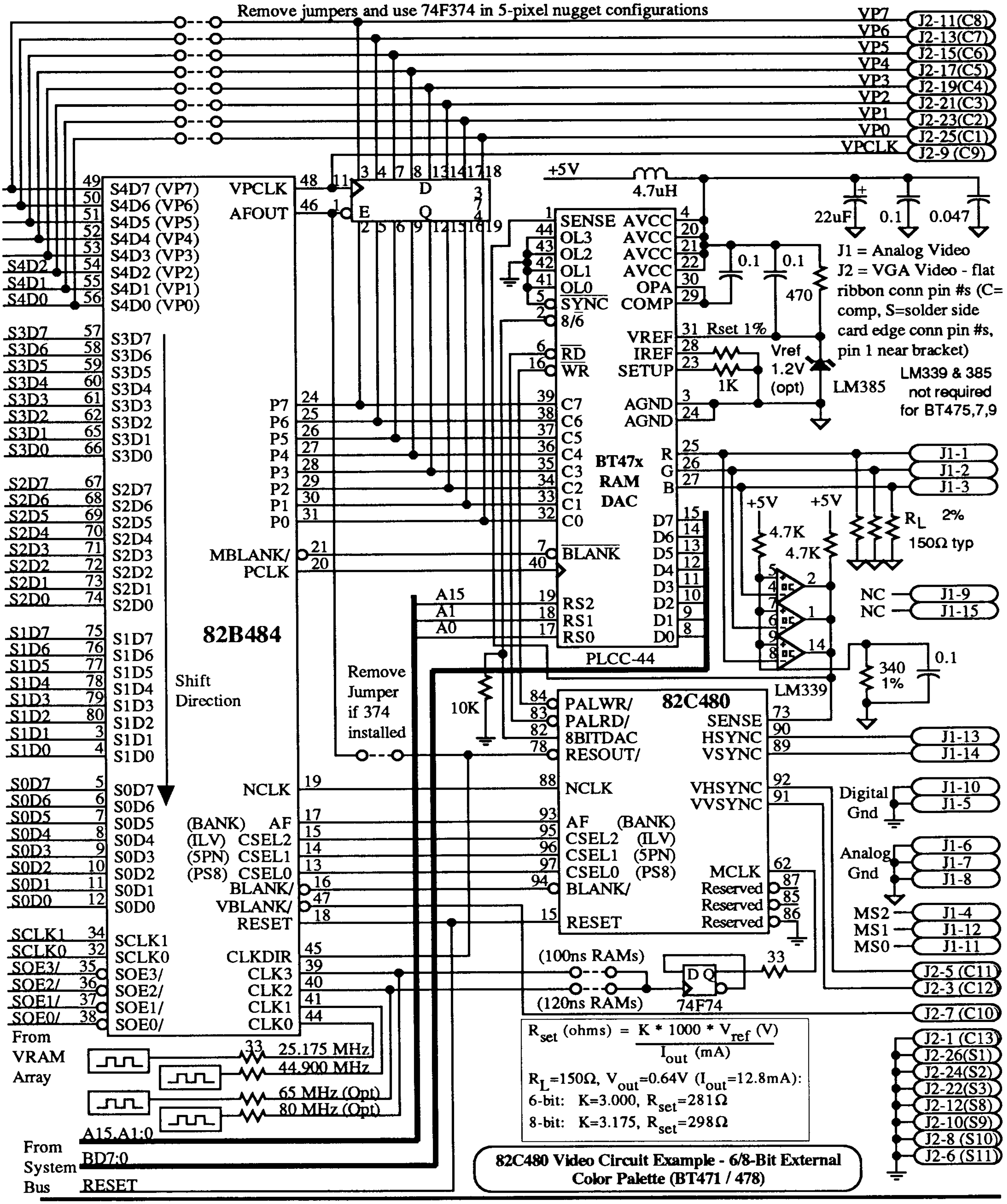


Remove jumpers and use 74F374 in 5-pixel nugget configurations

J1 = Analog Video
J2 = VGA Video - flat ribbon conn pin #s (C=component side, S=solder side card edge conn pin #s, pin 1 near bracket)

$$\text{Rset} = \frac{(21.3)\,(R_L)}{(\text{Vout})(R_L+75)} \quad \begin{array}{l} = 22.0\Omega \text{ for Vout=0.64V, } R_L=150 \\ = 15.4\Omega \text{ for Vout=0.70V, } R_L=75 \end{array}$$

82C480 Video Circuit Example - 6/8-Bit External Color Palette (IMSG176 / 178)


Remove jumpers and use 74F374 in 5-pixel nugget configurations
VP7 J2-11(C8)
VP6 J2-13(C7)
VP5 J2-15(C6)
VP4 J2-17(C5)
VP3 J2-19(C4)
VP2 J2-21(C3)
VP1 J2-23(C2)
VP0 J2-25(C1)
VPCLK J2-9 (C9)
82B484
S4D7 (VP7) S4D6 (VP6) S4D5 (VP5) S4D4 (VP4) S4D3 (VP3) S4D2 (VP2) S4D1 (VP1) S4D0 (VP0)
S3D7 S3D6 S3D5 S3D4 S3D3 S3D2 S3D1 S3D0
S2D7 S2D6 S2D5 S2D4 S2D3 S2D2 S2D1 S2D0
S1D7 S1D6 S1D5 S1D4 S1D3 S1D2 S1D1 S1D0
S0D7 S0D6 S0D5 S0D4 S0D3 S0D2 S0D1 S0D0
SCLK1 SCLK0 SOE3/ SOE2/ SOE1/ SOE0/
Shift Direction
VPCLK AFOUT P7 P6 P5 P4 P3 P2 P1 P0 MBLANK/ PCLK NCLK
(BANK) AF (ILV) CSEL2 (5PN) CSEL1 (PS8) CSEL0 BLANK/ VBLANK/ RESET CLKDIR CLK3 CLK2 CLK1 CLK0
From VRAM Array
D E Q
Remove Jumper if 374 installed
10K
A15 A1 A0
+5V 4.7uH
22uF 0.1 0.047
SENSE OL3 OL2 OL1 OL0 SYNC 8/6 RD WR C7 C6 C5 C4 C3 C2 C1 C0 BLANK RS2 RS1 RS0
AVCC AVCC AVCC AVCC OPA COMP VREF IREF SETUP AGND AGND R G B D7 D6 D5 D4 D3 D2 D1 D0
BT47x RAM DAC
PLCC-44
0.1 0.1 470
Rset 1% 1K Vref 1.2V (opt) LM385
J1 = Analog Video
J2 = VGA Video - flat ribbon conn pin #s (C= comp, S=solder side card edge conn pin #s, pin 1 near bracket)
LM339 & 385 not required for BT475,7,9
J1-1 J1-2 J1-3
+5V +5V 4.7K 4.7K
RL 2% 150Ω typ
LM339
340 1% 0.1
NC J1-9
NC J1-15
82C480
PALWR/ PALRD/ 8BITDAC RESOUT/ NCLK AF (BANK) CSEL2 (ILV) CSEL1 (5PN) CSEL0 (PS8) BLANK/ RESET
SENSE HSYNC VSYNC VHSYNC VVSYNC MCLK Reserved Reserved Reserved
J1-13 J1-14
Digital Gnd J1-10 J1-5
Analog Gnd J1-6 J1-7 J1-8
MS2 J1-4
MS1 J1-12
MS0 J1-11
(100ns RAMs) (120ns RAMs) 74F74 D Q 33
J2-5 (C11) J2-3 (C12) J2-7 (C10)
J2-1 (C13) J2-26(S1) J2-24(S2) J2-22(S3) J2-12(S8) J2-10(S9) J2-8 (S10) J2-6 (S11)
33 25.175 MHz
44.900 MHz
65 MHz (Opt)
80 MHz (Opt)
From System Bus A15,A1:0 BD7:0 RESET

$$R_{set}\text{ (ohms)} = \frac{K * 1000 * V_{ref}\text{ (V)}}{I_{out}\text{ (mA)}}$$

R_L=150Ω, V_{out}=0.64V (I_{out}=12.8mA):

6-bit: K=3.000, R_{set}=281Ω

8-bit: K=3.175, R_{set}=298Ω

82C480 Video Circuit Example - 6/8-Bit External Color Palette (BT471 / 478)

This page was intentionally left blank

82C480 Electrical Specifications

82C480 ABSOLUTE MAXIMUM CONDITIONS

Symbol	Parameter	Min	Max	Units
P_D	Power Dissipation	–	0.5	W
V_{CC}	Supply Voltage	–0.5	6.5	V
V_I	Input Voltage	–0.5	V_{CC}+0.5	V
V_O	Output Voltage	–0.5	V_{CC}+0.5	V
T_{OP}	Operating Temperature (Ambient)	–25	85	°C
T_{STG}	Storage Temperature	–40	125	°C

Note: Permanent device damage may occur if Absolute Maximum Ratings are exceeded. Functional operation should be restricted to the conditions described under Normal Operating Conditions.

82C480 NORMAL OPERATING CONDITIONS

Symbol	Parameter	Min	Typ	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
T_A	Ambient Temperature	0	–	70	°C

82C480 DC CHARACTERISTICS

(Under Normal Operating Conditions Unless Noted Otherwise)

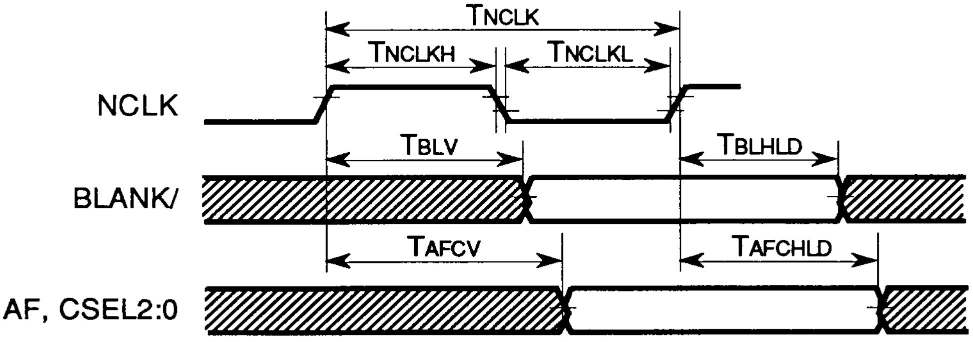
Symbol	Parameter	Notes	Min	Max	Units
I_{CC1}	Power Supply Current	@40 MHz CLK, 0°C	–	–	mA
I_{IL}	Input Leakage Current	Input, Tristate and Bidirectional Pins	–	10	uA
		@V_{IL} = –0.5V for pins with internal pullups	–	300	uA
I_{OZ}	Output Leakage Current	High Impedance	–100	+100	uA
V_{IL}	Input Low Voltage		–0.5	0.8	V
		8BITDAC pin only	0	0.3	V
V_{IH}	Input High Voltage		2.0	V_{CC}+0.5	V
		8BITDAC pin only	0.7	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL} = –16 mA (IRQ)	–	0.4	
		I_{OL} = –12 mA (HSYNC, VSYNC)	–	0.4	V
		I_{OL} = –4 mA (all others)	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} = 8 mA (IRQ)	2.4	–	V
		I_{OH} = 0.5 mA (all others)	2.4	–	V

Electrical specifications contained herein are preliminary and subject to change without notice.

82C480 AC TIMING CHARACTERISTICS - NUGGET CLOCK TIMING

Symbol	Parameter	Notes	Min	Max	Units
T_{NCLK}	Nugget Clock Cycle Time	30 MHz., max	33	Note 1	nS
T_{NCLKH}	Nugget Clock High Time		15	–	nS
T_{NCLKL}	Nugget Clock Low Time		15	–	nS
T_{BLV}	Nugget Clock to Blank Valid		–	25	nS
T_{AFCV}	Nugget Clock to AF, CSEL valid		–	25	nS
T_{BLHLD}	Blank hold from NCLK high		2	–	nS
T_{AFCHLD}	AF, CSEL Hold from NCLK High		2	–	nS

Note 1: The Nugget Clock frequency must be high enough to satisfy the refresh requirements of the VRAMs. The limiting display mode will be the one for which the horizontal total value is greatest. Refresh is performed for two VRAM rows each horizontal scan line. Therefore, for 256K VRAMs, the Refresh Period = TNCLK*(horizontal total)*256; and twice this value for 1M VRAMs.



82C480 Nugget Clock Timing

82C480 AC TIMING CHARACTERISTICS - RESET TIMING

Symbol	Parameter	Notes	Min	Max	Units
–	RESET Pulse Width		64 T_{MCLK}	–	nS
T_{MCLK}	Memory Clock period	Dependant on VRAM access	25	–	nS
		time.			
T_{MCLKH}	Memory Clock High		11	–	nS
T_{MCLKL}	Memory Clock Low		11	–	nS

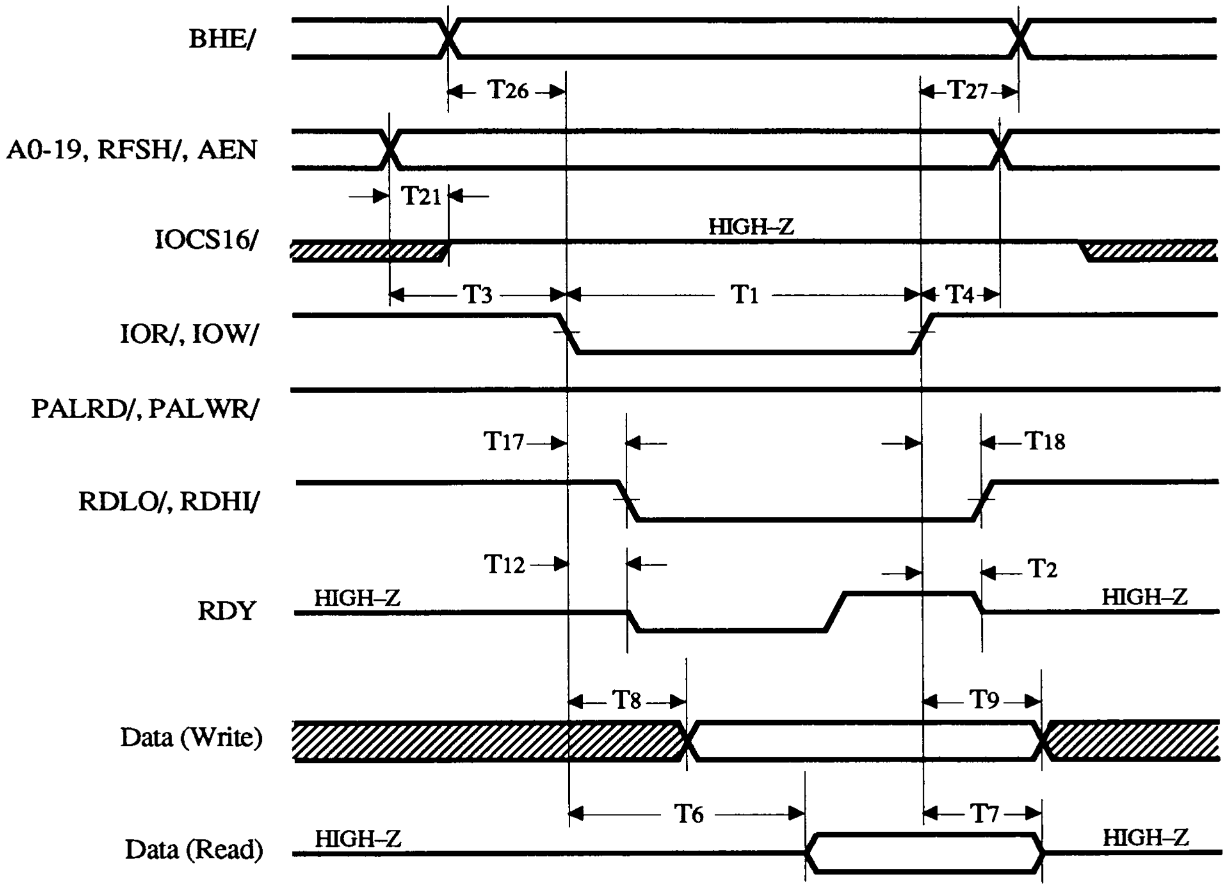
Output load C_L= 50pF unless otherwise noted.

82C480 AC TIMING CHARACTERISTICS - ISA BUS TIMING

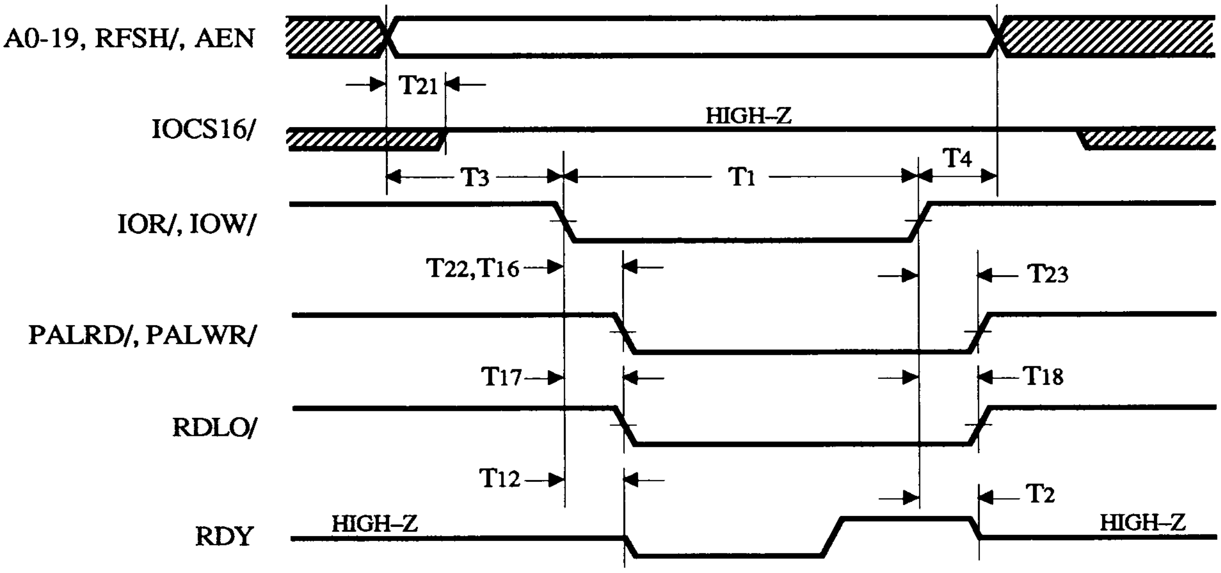
Symbol	Parameter	Notes	Min	Max	Units
T1	IOR/, IOW/ Pulse Width	Video registers	$3*T_{NCLK}$	–	nS
		All others	$4*T_{MCLK}$	–	nS
T2	RDY tristate (high-z) from IOR/,IOW/ inactive	Note 2	–	25	nS
T3	A19:0, RFSH/, AEN setup to command		40	–	nS
T4	A19:0, RFSH/, AEN hold from command		20	–	nS
T6	I/O Read Data valid from IOR/ active		–	50	nS
T7	I/O Read Data hold from IOR/ inactive		0	40	nS
T8	IOW/ active to Data in Valid		–	15	nS
T9	I/O Write Data hold from IOW/ inactive		0	–	nS
T12	IOR/, IOW/ to RDY inactive (Low)		–	25	nS
T16	ROMCS/, PALRD/ active from MEMR/, IOR/	Note 3	0	45	nS
T17	RDLO/, RDHI/ active from MEMR/, IOR/		–	45	nS
T18	RDLO/, RDHI/ inactive from MEMR/, IOR/		–	45	nS
T19	ROMCS/, PALRD/ inactive from MEMR/, IOR/	Note 3	0	25	nS
T20	IOCS16/ active from Address Valid		–	25	nS
T21	IOCS16/ inactive from Address Valid		–	25	nS
T22	PALWR/ active from IOW/	Note 3	–	25	nS
T23	PALWR/ inactive from IOW/	Note 3	–	25	nS
T25	IOR/, IOW/ recovery time		80	–	nS
T26	BHE/ valid to IOR/, IOW/ active		25	–	nS
T27	BHE/ hold from IOR/, IOW/ inactive		30	–	nS
T28	ROMPG[2:0] valid from MEMR/ active		–	25	nS
T29	ROMPG[2:0] hold from MEMR/ inactive		0	–	nS

Note 2: The 82C480 will asynchronously bring RDY active (high-z) if the current command cycle ends while ready is inactive (low).
Note 3: The 82C480 extends palette I/O read/write accesses (asynchronously) with RDY.

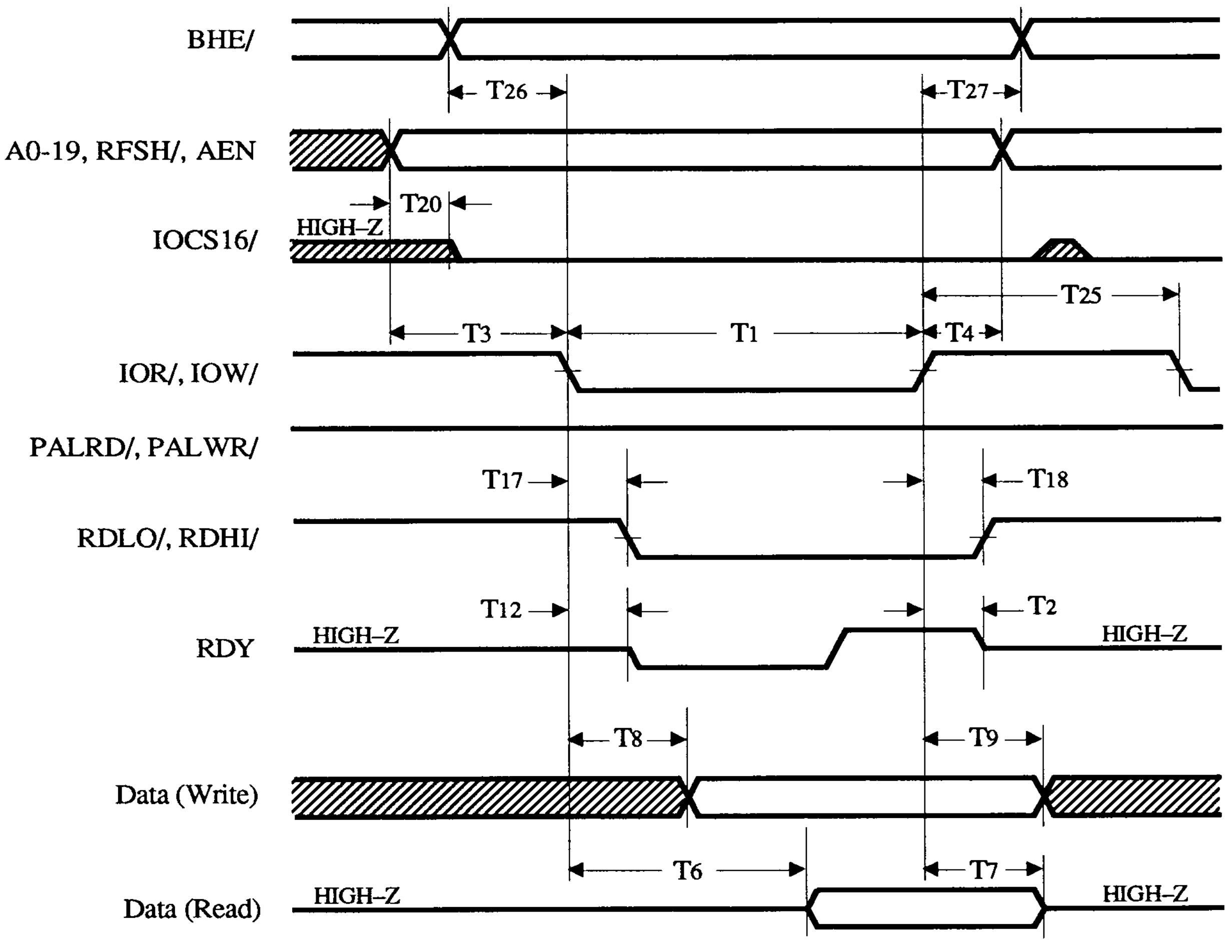
Output load C_L= 50pF unless otherwise noted.



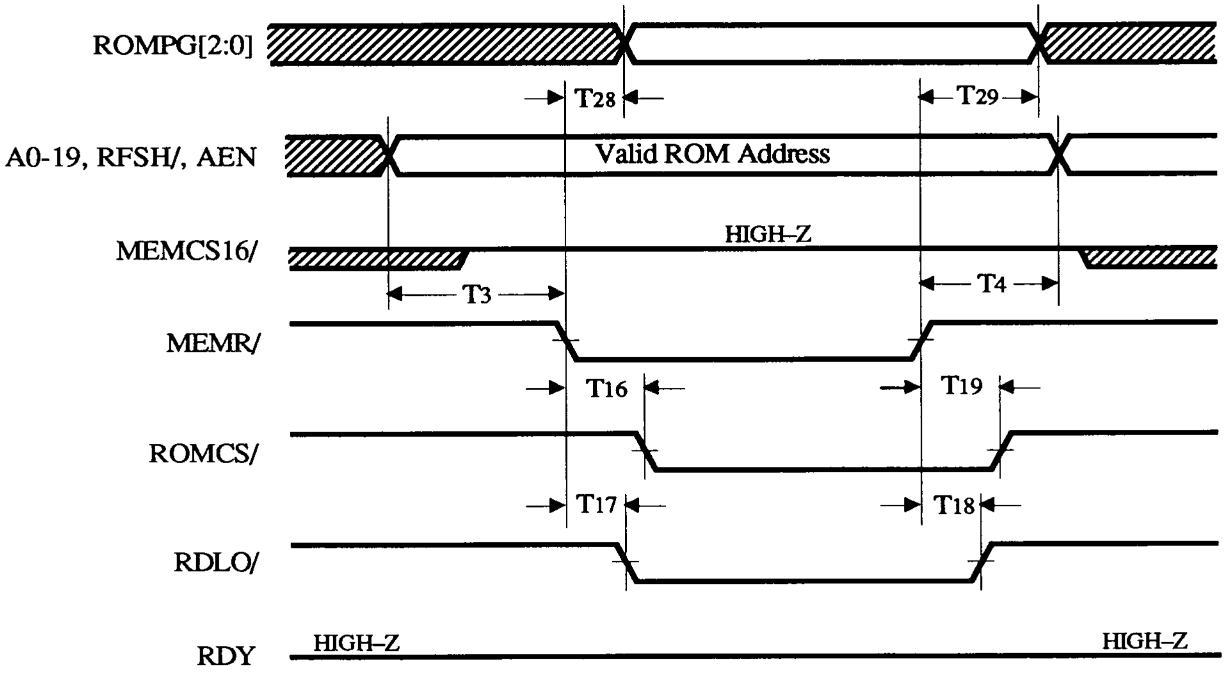
ISA 8-Bit Bus I/O Cycle Timing



ISA Bus Palette I/O Cycle Timing



ISA Bus 16-Bit I/O Cycle Timing

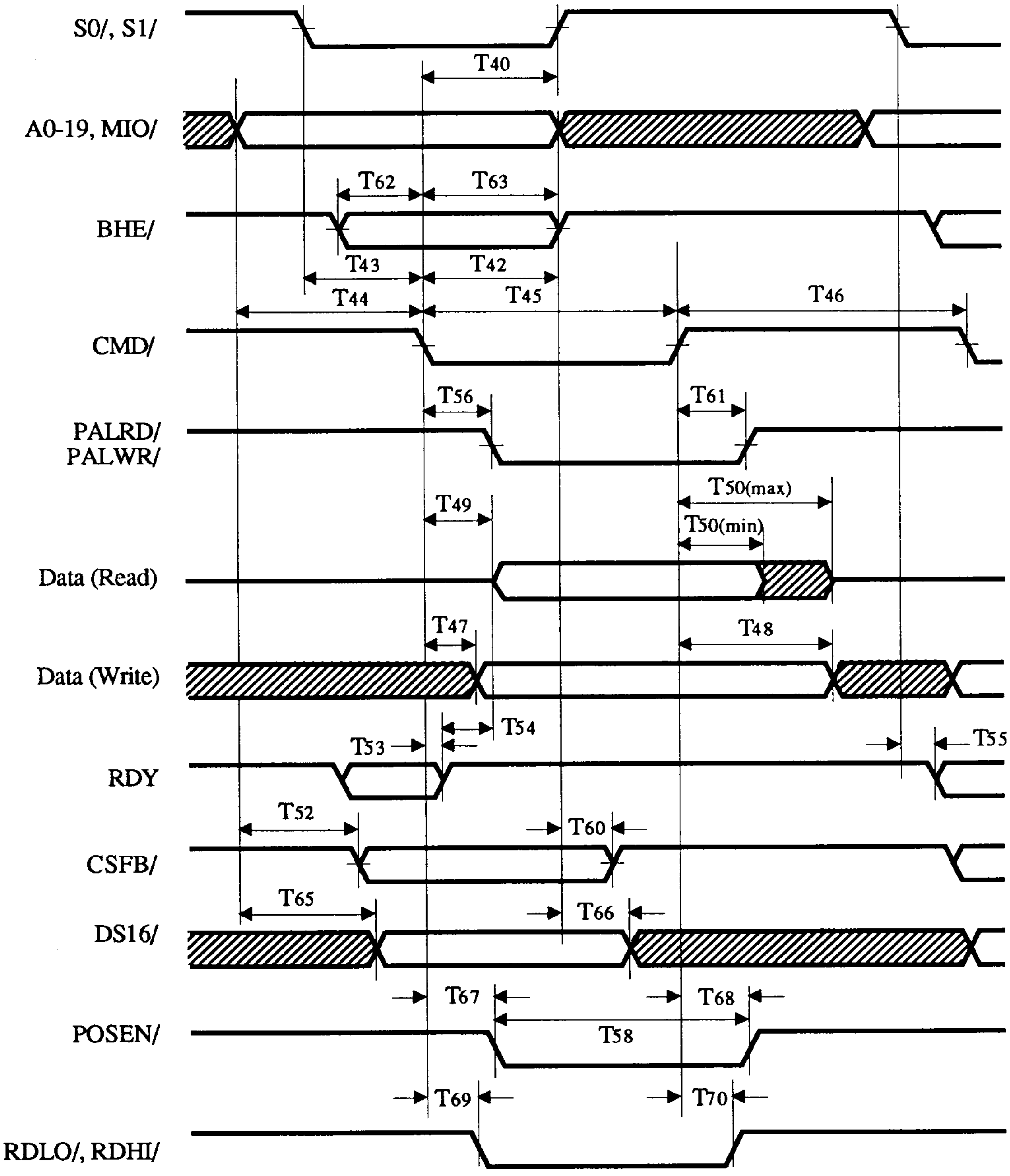


ISA Bus ROM Memory Cycle Timing

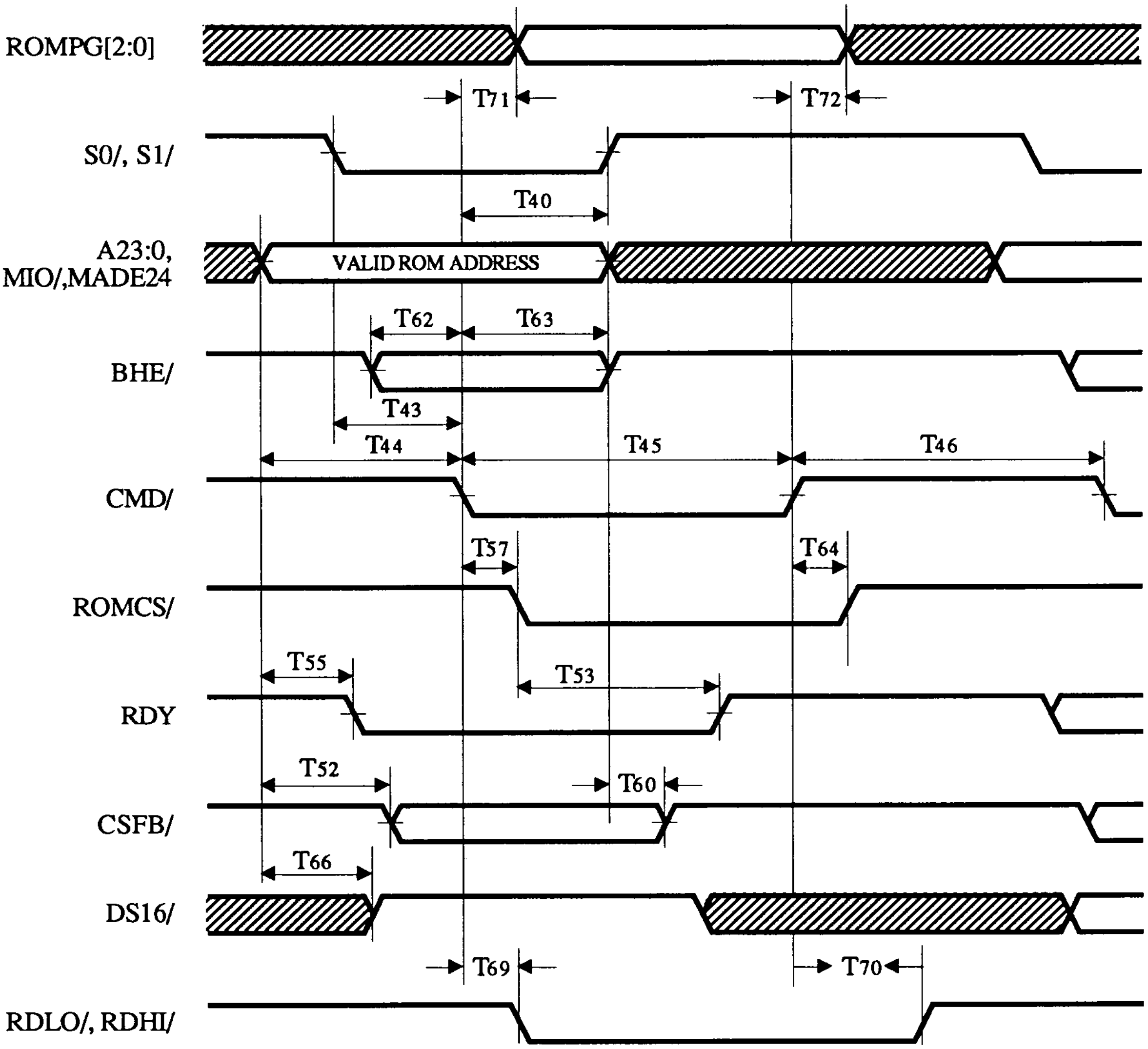
82C480 AC TIMING CHARACTERISTICS - MCA BUS TIMING

Symbol	Parameter	Notes	Min	Max	Units
T40	Status hold from CMD/ inactive		20	–	nS
T42	Address hold from CMD/ inactive		30	–	nS
T43	Status setup to CMD/ active		45	–	nS
T44	Address Valid setup to CMD/ active		55	–	nS
T45	CMD/ Pulse Width	Video registers	$3*T_{NCLK}$	–	nS
		All others	$4*T_{MCLK}$	–	nS
T46	CMD/ inactive to next CMD/ active		80	–	nS
T47	CMD/ active to Write Data Valid		–	15	nS
T48	Write data hold from CMD/ inactive		0	–	nS
T49	Read data valid from CMD/ active		–	50	nS
T50	Read data hold from CMD/ inactive		0	40	nS
T52	CSFB/ active from Address valid		–	25	nS
T53	RDY active from CMD/ active		–	$64*T_{MCLK}$	nS
T54	Read data valid from RDY active (high)		–	50	nS
T55	RDY inactive (low) from Address valid		–	25	nS
T56	PALWR/ delay from CMD/ active		–	25	nS
T57	ROMCS/, PALRD/ active from CMD/ active		–	45	nS
T59	ROMPG[2:0] Valid from CMD/ active		–	40	nS
T60	CSFB/ inactive from Address invalid		–	25	nS
T61	PALWR/ inactive from CMD/ inactive		–	25	nS
T62	BHE/ valid to CMD/ active		25	–	nS
T63	BHE/ hold from CMD/ active		30	–	nS
T64	ROMCS/, PALRD/ inactive from CMD/ inactive		–	25	nS
T65	DS16/ active from Address valid		–	25	nS
T66	DS16/ inactive from Address invalid		–	25	nS
T67	POSEN/ active from CMD/ active		–	45	nS
T68	POSEN/ inactive from CMD/ inactive		–	25	nS
T69	RDLO/, RDHI/ active from CMD/ active		–	45	nS
T70	RDLO/, RDHI/ inactive from CMD/ inactive		–	45	nS
T71	ROMPG[2:0] valid from CMD/ active		–	25	nS
T72	ROMPG[2:0] hold from CMD/ inactive		0	–	nS

Output load C_L= 50pF unless otherwise noted.



MCA I/O BUS Cycle Timing



MCA Bus ROM Memory Cycle Timing

82C480 AC CHARACTERISTICS

VRAM TIMING - SWITCHING CHARACTERISTICS

Symbol	Parameter	Min	Typ	Max	Units
T_{RAC}	Data Access Time from RAS/			100	nS
T_{CAC}	Data Access Time from CAS/			30	nS
T_{AA}	Access time from Column Address			60	nS
T_{ACP}	Access time from rising edge of CAS/			60	nS
T_{OEA}	Access time from OE/			30	nS
T_{OFF}	Output disable time from CAS/ high			30	nS
T_{OEZ}	Output disable time from OE/ high			30	nS

Note:All VRAM timing characteristics are preliminary values referenced to a 40 MHz MCLK.

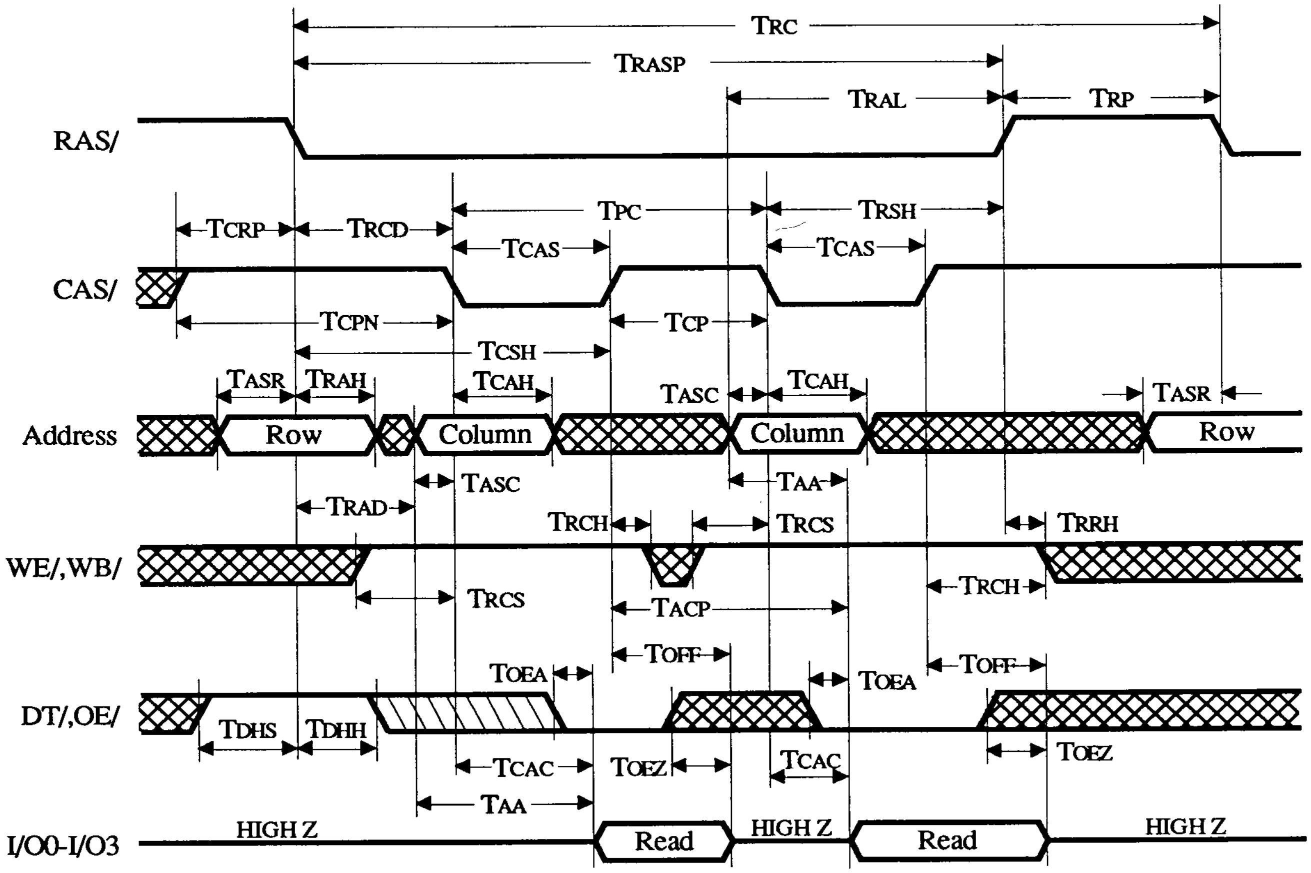
VRAM TIMING - TIMING REQUIREMENTS

Symbol	Parameter	Min	Typ	Max	Units
T_{RC}	Random read or write cycle time	200			nS
T_{RWC}	Read-modify-write cycle time	300			nS
T_{PC}	Fast-page cycle time	75			nS
T_{PRWC}	Fast-page read-modify-write cycle time	150			nS
T_T	Transition time (rise and fall)	3			nS
T_{RP}	RAS/ precharge	80			nS
T_{RAS}	RAS/ pulse width	100	T_{NCLK}*(H_TOTAL+1)/2		nS
T_{RASP}	Fast-page RAS/ pulse width	100	T_{NCLK}*(H_TOTAL+1)/2		nS
T_{RSH}	RAS/ hold from CAS/	35			nS
T_{CPN}	CAS/ precharge	15			nS
T_{CP}	Fast-page CAS/ precharge time	15			nS
T_{CAS}	CAS/ pulse width	50			nS
T_{CSH}	CAS/ hold from RAS/	110			nS
T_{RCD}	RAS/ to CAS/ delay	50			nS
T_{CRP}	CAS/ high to RAS/ low precharge	25			nS
T_{ASR}	Row Address setup time	5			nS
T_{RAH}	Row Address hold time	15			nS
T_{ASC}	Column Address setup to CAS/	10		25	nS
T_{CAH}	Column Address hold time	20			nS

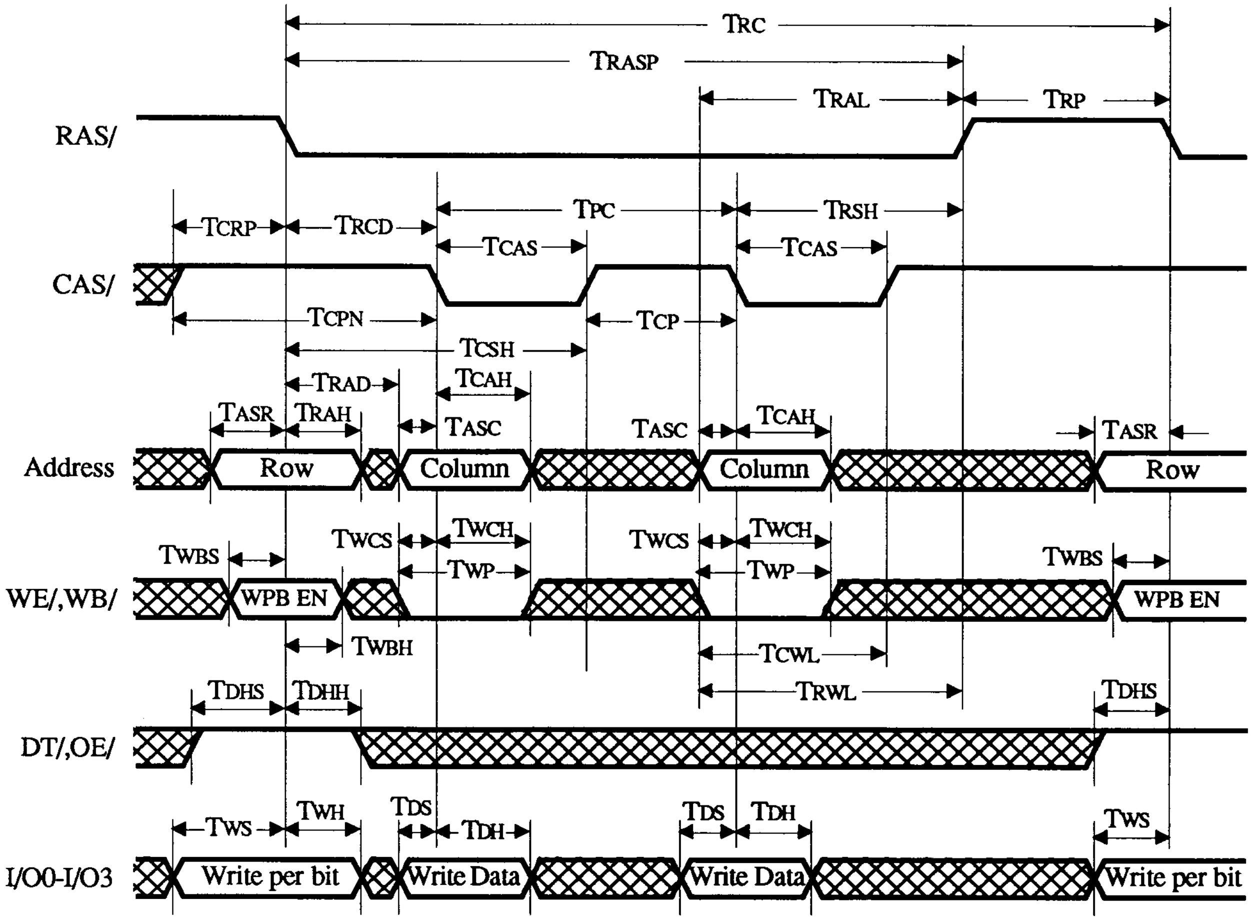
82C480 AC CHARACTERISTICS

VRAM TIMING - TIMING REQUIREMENTS (continued)

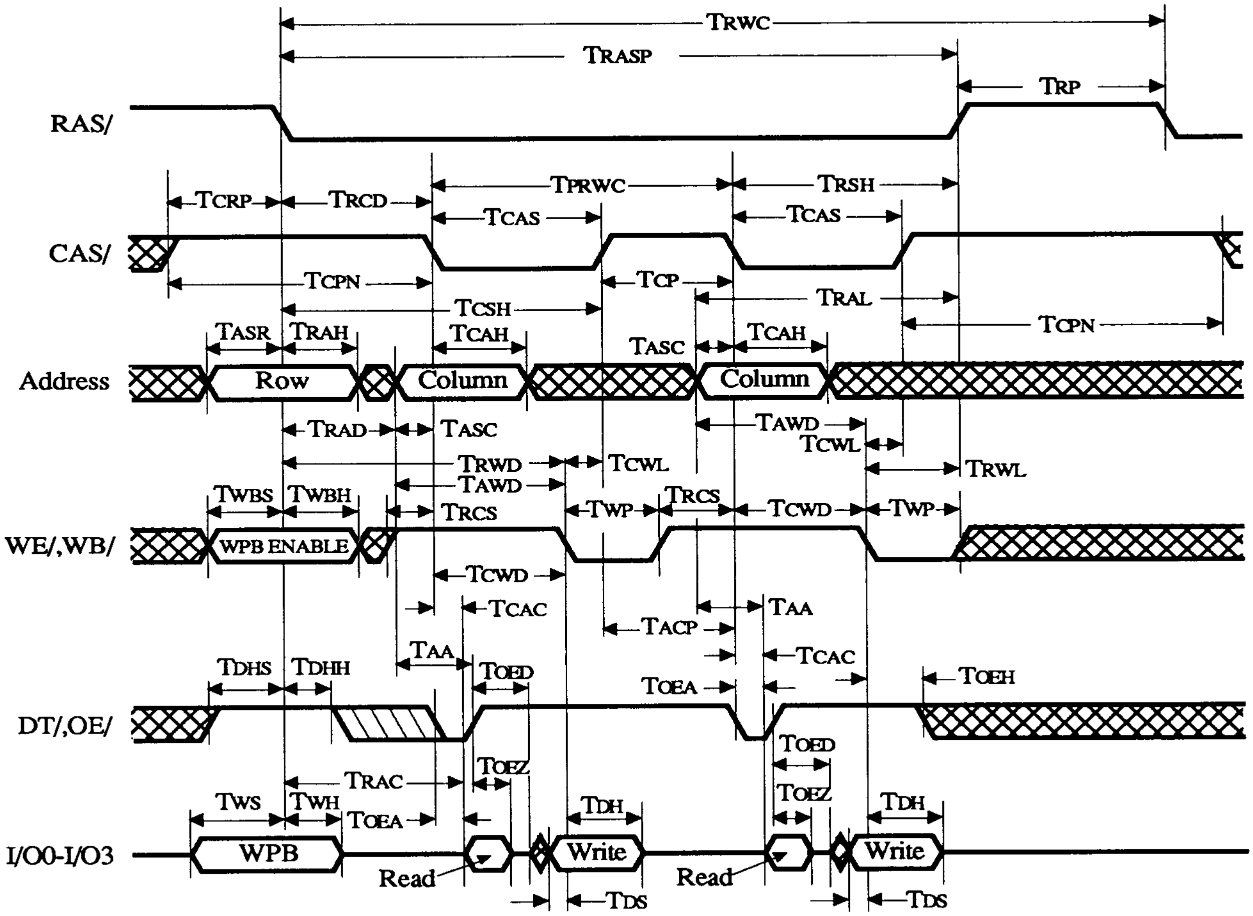
Symbol	Parameter	Min	Typ	Max	Units
T_{RAD}	RAS/ to Column Address delay time	25		45	nS
T_{RAL}	Column Address to RAS/ lead time	25			nS
T_{RCS}	Read command setup time	10			nS
T_{RRH}	Read command hold time after RAS/ high	15			nS
T_{RCH}	Read command hold time after CAS/ high	10			nS
T_{WCS}	Write command setup time	N/A			nS
T_{WCH}	Write command hold time	N/A			nS
T_{WP}	Write command pulse width	50			nS
T_{RWL}	Write command to RAS/ lead time	50			nS
T_{CWL}	Write command to CAS/ lead time	50			nS
T_{DS}	Data-in setup time	10			nS
T_{DH}	Data-in hold time	25			nS
T_{AWD}	Column Address to WE/ delay	100			nS
T_{CWD}	CAS/ to WE/ delay	100			nS
T_{RWD}	RAS/ to WE/ delay	150			nS
T_{OED}	OE/ high to data-in setup delay	15			nS
T_{OEH}	OE/ high hold time after WE/ low	35			nS
T_{RPC}	RAS/ high to CAS/ low precharge time	25			nS
T_{REF}	Refresh time interval	T_{NCLK}*(H_TOTAL+1)/2			mS
T_{DLS}	DT/ low setup time	5			nS
T_{RDH}	DT/ low hold time after RAS/ low	100			nS
T_{CDH}	DT/ low hold time after CAS/ low	40			nS
T_{DHS}	DT/ high setup time	25			nS
T_{DHH}	DT/ high hold time	25			nS
T_{DTR}	DT/ high to RAS/ high delay	10			nS
T_{DTC}	DT/ high to CAS/ high delay	10			nS
T_{WBS}	Write-per-bit setup time	5			nS
T_{WBH}	Write-per-bit hold time	25			nS
T_{WS}	Write bit selection setup time	5			nS
T_{WH}	Write bit selection hold time	25			nS
T_{DTH}	DT/ high hold time after RAS/ high	25			nS



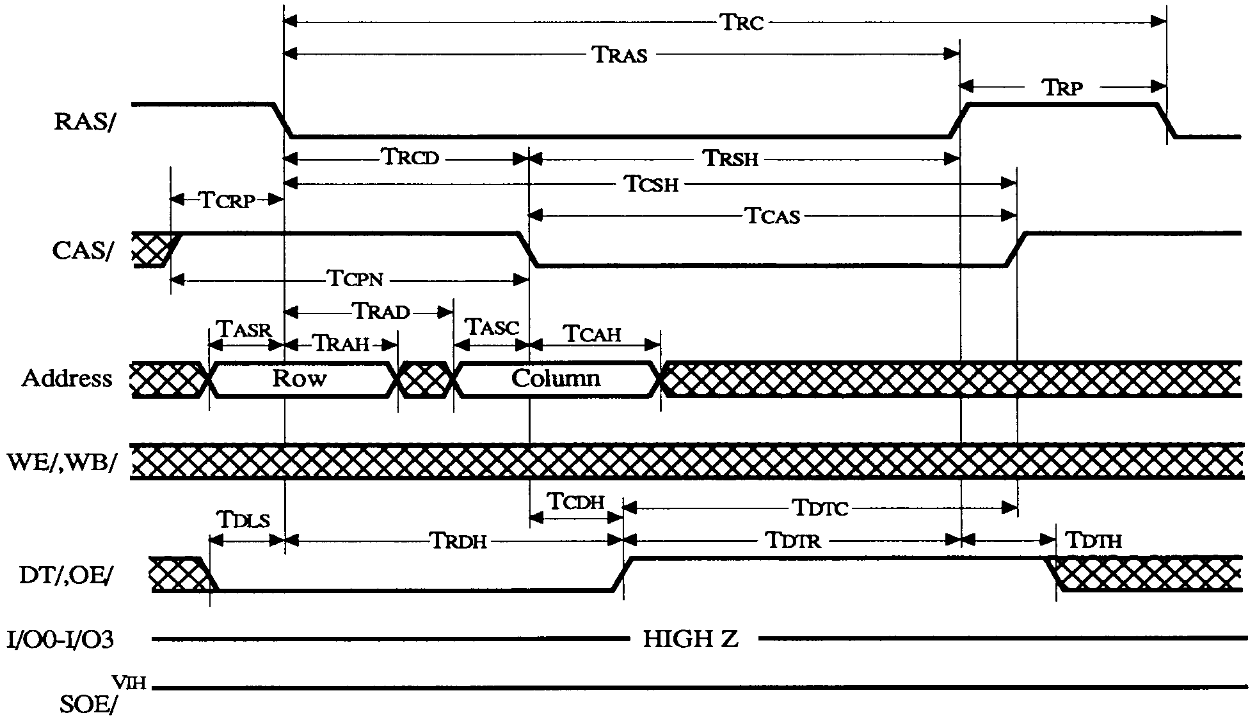
VRAM Fast-page Read Cycle Timing



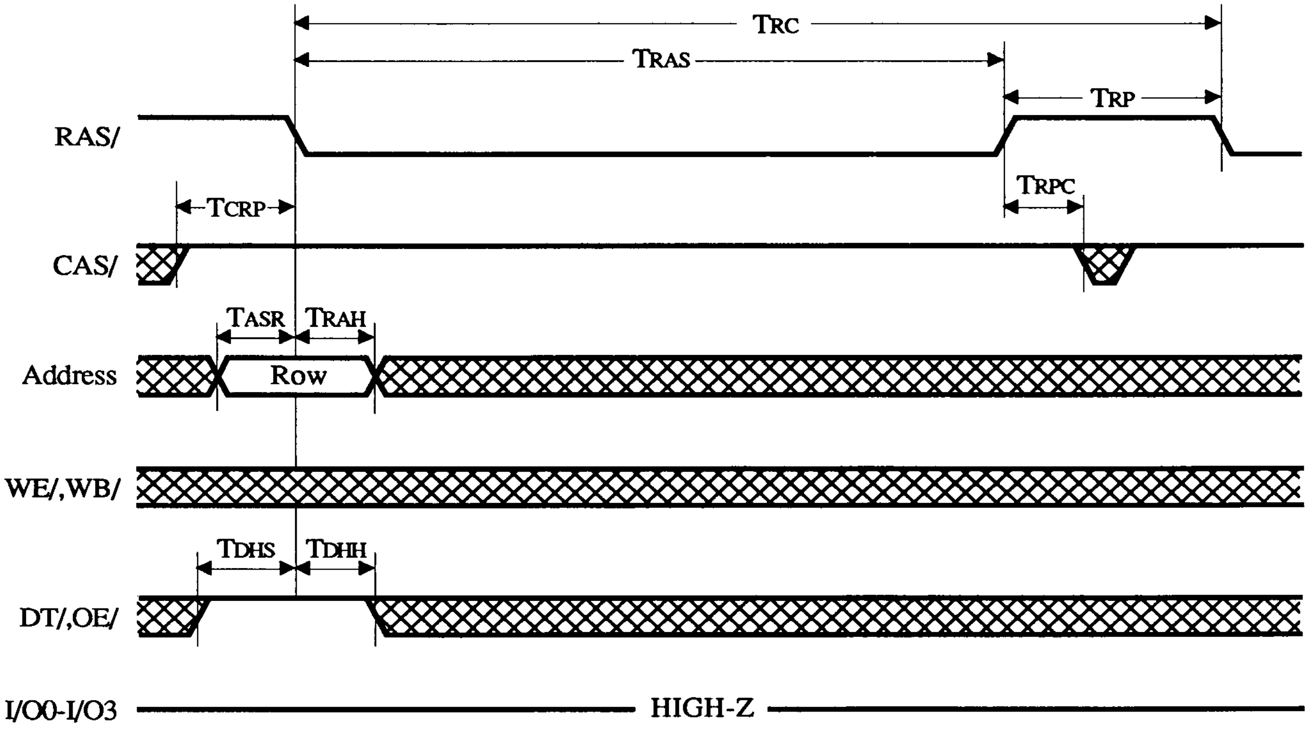
VRAM Fast-page Write Cycle Timing



VRAM Fast-page Read-Modify-Write Cycle Timing



VRAM Data Transfer Cycle Timing

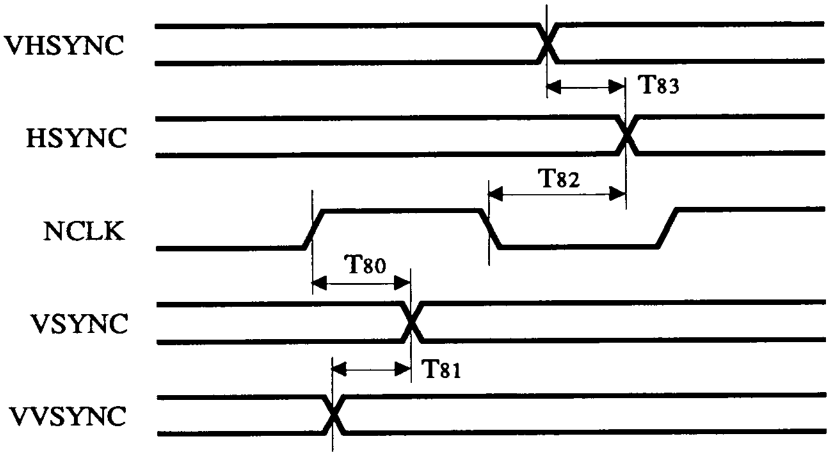


VRAM Refresh Cycle Timing

82C480 AC TIMING CHARACTERISTICS - VIDEO TIMING

Symbol	Parameter	Notes	Min	Max	Units
T80	VSYNC delay from PCLK		–	25	nS
T81	VSYNC delay from VVSYNC		–	25	nS
T82	HSYNC delay from NCLK		–	25	nS
T83	HSYNC delay from VHSYNC		–	25	nS

Output load C_L= 50pF unless otherwise noted.



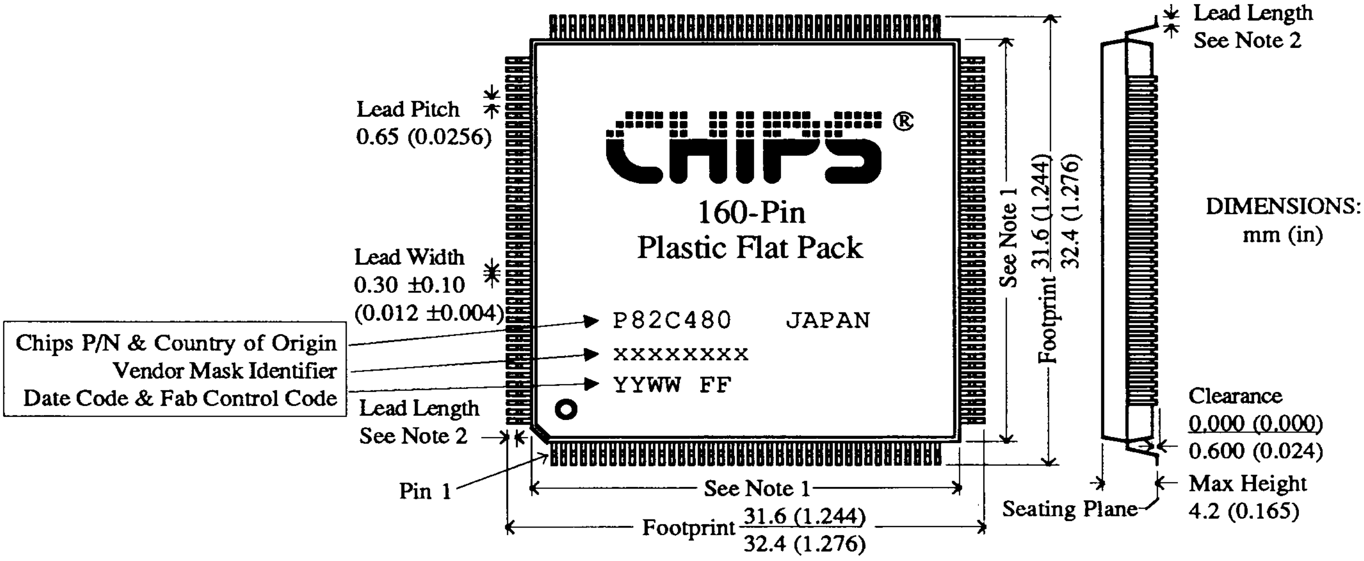
82C480 Video Timing

Video Register Parameters

Register Name	I/O Address	Video Mode				
		640x480 (Non-Interlaced)	640x480 Pseudo-8	1024x768 (Interlaced)	1024x768 (Non-Interlaced)	1280x1024 (Interlaced)
ADVFUNC_CNTL	4AE8	0003	0003	0007	0001	0001
DISP_CNTL	22E8	0023	0021	0033	0023	0033
MEM_CNTL	BEE8[5]	5006	5002	5006	5006	5006
H_TOTAL	02E8	0063	0063	009D	009D	00C3
H_DISP	06E8	004F	004F	007F	007F	009F
H_SYNC_STRT	0AE8	0052	0052	0081	0081	00A1
H_SYNC_WID	0EE8	002C	002C	0016	0016	001B
V_TOTAL	12E8	0418	0830	0660	0660	0918
V_DISP	16E8	03BB	0779	05FB	05FB	07FB
V_SYNC_STRT	1AE8	03D2	07A8	0600	0600	0800
V_SYNC_WID	1EE8	0022	0022	0008	0008	000C
SCISSORS_T	BEE8[1]	1000	1000	1000	1000	1000
SCISSORS_L	BEE8[2]	2000	2000	2000	2000	2000
SCISSORS_B	BEE8[3]	31DF	31DF	32FF	32FF	33FF

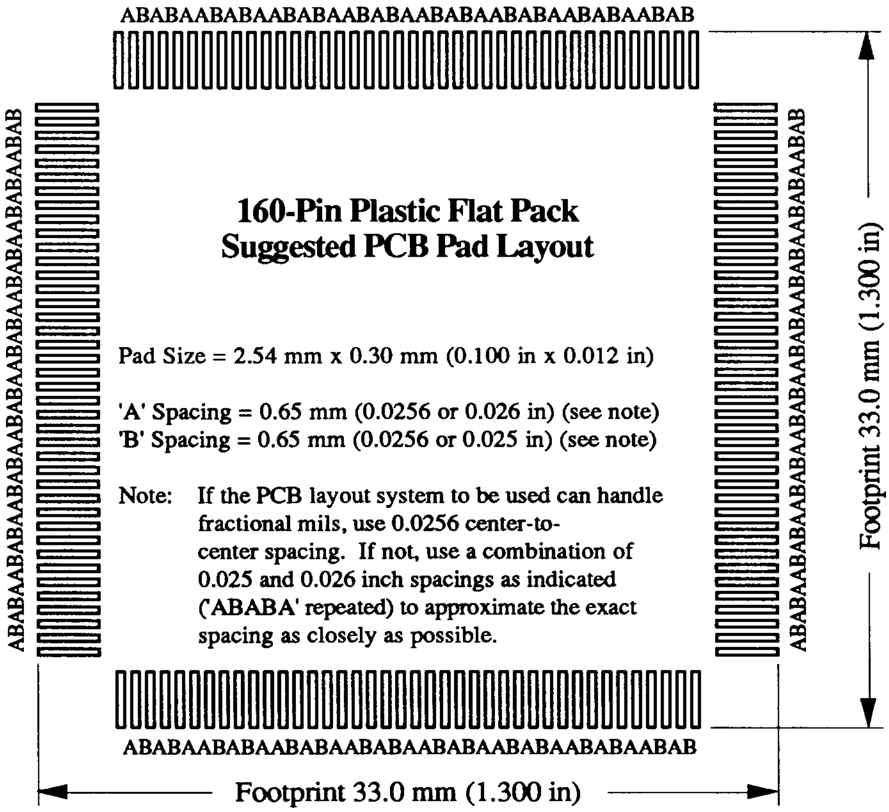
This page was intentionally left blank

82C480 Mechanical Specifications



Note 1: Package Body Size = 28 ±0.2 (1.102 ±0.008)
Note 2: Lead Length = 0.8 ±0.2 (0.031 ±0.008)

82C480 Suggested PCB Pad Layout


ABABAABABAABABAABABAABABAABABAABABAABAB
ABABAABABAABABAABABAABABAABABAABABAABAB
ABABAABABAABABAABABAABABAABABAABABAABAB
ABABAABABAABABAABABAABABAABABAABABAABAB
Footprint 33.0 mm (1.300 in)
Footprint 33.0 mm (1.300 in)

160-Pin Plastic Flat Pack
Suggested PCB Pad Layout

Pad Size = 2.54 mm x 0.30 mm (0.100 in x 0.012 in)

'A' Spacing = 0.65 mm (0.0256 or 0.026 in) (see note)
'B' Spacing = 0.65 mm (0.0256 or 0.025 in) (see note)

Note: If the PCB layout system to be used can handle fractional mils, use 0.0256 center-to-center spacing. If not, use a combination of 0.025 and 0.026 inch spacings as indicated ('ABABA' repeated) to approximate the exact spacing as closely as possible.