Chapter 1. Technical Overview

Single-Chip 2-D Graphics Accelerator

- ◇ Ultra-high-speed local-bus display controller uses workstation display technology to give maximum acceleration with graphical user interfaces such as Microsoft Windows
- ◇ Sophisticated architecture, full-custom chip design, and advanced 0.8 micron process technology deliver maximum power at an affordable cost
- ◇ On-chip SVGA unit for DOS compatibility
- ◇ 208-pin plastic QFP package

High Bandwidth

- ◇ Interleaved VRAM controller uses a PLL timing generator to extract 100% of VRAM bandwidth
- ◇ Sophisticated local-bus controller extracts full bandwidth from PCI and VL host buses
- ◇ Supports 32- and 64-bit RAMDACs at speeds up to 200 MHz for maximum display bandwidth
- ◇ Pipelined internal architecture is 32 bits wide throughout to ensure maximum throughput

Full Software Support

- ◇ Drivers for Windows 3.1, Windows NT, Microstation, OS/2 Presentation Manager, and AutoCAD R12
- ◇ VESA-compatible BIOS
- ◇ DOS application drivers

Powerful Graphics Features

- ◇ Supports high- and true-color at large screen sizes:
  - ○ 24-bit true-color to 1280x1024 pixels
  - ○ 16-bit high-color to 1600x1200 pixels
- ◇ Fully accelerates 8, 15, 16, 24, and 32 bits/pixel
- ◇ Draws lines and polygons (with optional pattern fill) at full VRAM bandwidth
- ◇ Performs bit block-transfer (BitBlt) from screen to screen at VRAM bandwidth, and from host to screen (with color expansion) at host bus bandwidth
- ◇ Provides automatic clipping against window and screen edges
- ◇ Supports patterning, plane masking, and boolean operations of pixels during drawing; supports polylines and mesh polygons; supports pick mode
- ◇ Directly mapped linear frame buffer allows efficient CPU access to frame-buffer memory without banking
- ◇ Supports multimedia via frame-buffer coprocessor interface
- ◇ Supports VESA DPMS power management

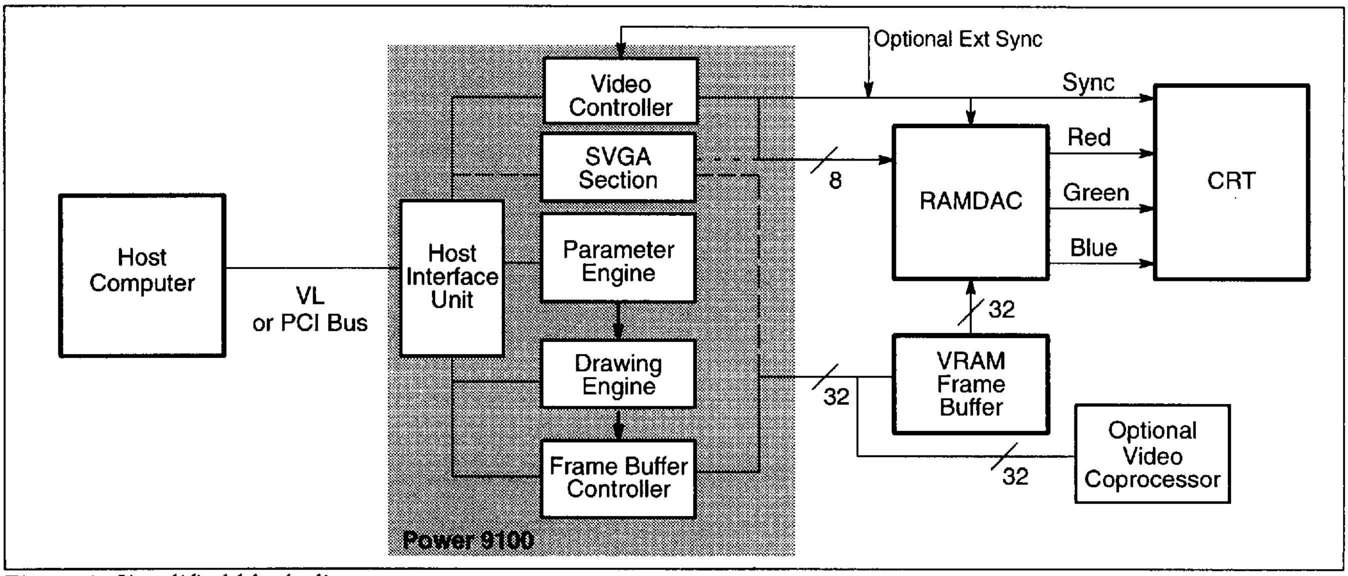


Figure 1. Simplified block diagram

1.1. Architecture

The WEITEK Power 91000 is a hard-wired display processor. Designed specifically for high-performance personal computers, its architecture takes full advantage of the speed of video RAM and high-speed local-bus interfaces.

Based on WEITEK's workstation display controllers, the Power 9100 is the latest in a series of controllers designed from an architectural strategy that can be summarized in two sentences:

1. Get the highest possible bandwidth.
2. Use all of it.

We have found that these simple statements capture the essence of high-performance controller design.

The following sections describe how these principles were applied in the main sections of the Power 9100:

1. Frame buffer controller
2. Parameter engine
3. Drawing engine
4. Host bus interface
5. Video controller

The Power 9100 contains an additional unit:

6. SVGA unit

for backward compatibility.

1.1.1. FRAME BUFFER CONTROLLER

Capturing the highest possible bandwidth starts at the frame buffer. The Power 9100 uses three techniques to achieve (and use) the highest possible frame buffer bandwidth:

1. VRAM frame buffer
2. Interleaving
3. PLL-controlled memory timing

VRAM FRAME BUFFER

The Power 9100 uses a VRAM frame buffer for maximum performance. While VRAM is more expensive than DRAM, it is dual-ported; the stream of pixel data needed to refresh the display does not detract from the drawing rate. DRAM systems, in contrast, divide their bandwidth between these two functions, causing drawing rates to degrade as screen sizes increase.

The Power 9100 supports screen sizes of up to four megapixels, and full drawing bandwidth is retained even at these large screen resolutions.

INTERLEAVING

The Power 9100's two 32-bit banks of VRAM are interleaved for maximum performance.

Interleaving is a technique that takes advantage of the fact that RAM access times are much longer than their write-enable times. Two banks of RAM are placed in parallel, with common address and data lines, but separate write enable lines. One bank contains the even-numbered data words; the other contains the odd-numbered words. With interleaved RAM, writing two words of data only takes one cycle longer than writing one, giving the performance of 64-bit memory, while saving pins and giving a better architectural match to a 32-bit controller.

PLL-CONTROLLED MEMORY TIMING

To get maximum speed out of memory devices, you have to precisely match a large number of timing specifications. Typical memory controllers approximate this with timing patterns based on the system clock. Since memory speeds and system clock speeds are similar, this approach is too granular to match the memory parameters closely, and it results in significant performance loss.

The Power 9100 can match memory parameters exactly, using a PLL (phase-locked loop) timing generator and a programmable memory controller. This allows the frame buffer to be used at its full theoretical capacity.

1.1.2. PARAMETER ENGINE

The graphics core of the Power 9100 is crucial to using all of the frame buffer's bandwidth. In addition to providing a low-latency path between the host and the frame buffer, the core must provide large amounts of acceleration if the full frame buffer bandwidth is to be used. The importance of acceleration stems from three factors:

1. Local-bus interfaces, while fast, are not as fast as the Power 9100's frame buffer.
2. General-purpose CPUs are inefficient at simple drawing operations, and cannot perform them at full host-bus bandwidth (let alone full frame-buffer bandwidth).
3. The user is better served if the CPU is running applications, not serving as an inefficient graphics controller.

1.1. Architecture, continued

The Power 9100's graphics accelerator is divided into two loosely coupled graphics engines: the *parameter engine* and the *drawing engine.* The *parameter engine* prepares drawing operations for execution by the drawing engine (which is described in the next section). The parameter engine's basic function is to take input coordinates from the host and convert them to a form usable by the drawing engine. The input parameters include the *x,y* vertices of polygons and the corners of bit block-transfer *(BitBlt)* regions. The parameter engine tests the vertices against window and screen boundaries, tests for exceptions, and performs trivial rejection. Finally, it transfers commands that pass these tests to the drawing engine to be rendered into the frame buffer.

The parameter engine works independently of the drawing engine; the parameters for a new operation can be loaded while the drawing engine is busy.

The parameter engine prepares four kinds of "polygons" for drawing: quadrilaterals, triangles, lines, and points." It also handles screen-to-screen BitBlt and host-to-screen BitBlt.

The parameter engine handles all exception testing, trivial rejection, status reporting, and access to parameter engine registers. Once the parameter engine verifies that a drawing command should be performed (that is, it has valid parameters and has not been trivially rejected), it passes the operation to the drawing engine. The parameter engine's exception testing is very fast, completing in a few cycles.

1.1.3. DRAWING ENGINE

The drawing engine performs three basic functions:

1. It draws quadrilaterals, triangles, and lines (the *quad* operation).
2. It performs screen-to-screen BitBlt (the *blit* operation).
3. It performs host-to-screen BitBlt (the *pixel1* and *pixel8* operations).

The quad operation draws quadrilaterals, triangle, lines, and points (the last three being treated by the hardware as special cases: quadrilaterals with one or more identical vertices). Triangles, lines, and points can always be rendered correctly, but the drawing engine cannot draw horizontally convex quadrilaterals. That is, it cannot cross from the inside to the outside of the same object more than once per scan line. This means that "bow ties" cannot be drawn (such quads are decomposed into two triangles by the driver software), though "hourglasses" can. See figure 2.

The blit operation copies a rectangular area of the display from one screen location to another.

The pixel1 operation takes monochrome, one-bit-per-pixel data from the host, expands the pixels internally to the current pixel depth (typically by assigning the foreground and background colors to one and zero bits), and writes them to the frame buffer. Up to 32 pixels can be transferred to the Power 9100 in a single word.

The pixel8 operation takes color pixels of 8, 16, or 32 bits, packed as four, two, or one pixel per word, respectively, and writes them to the frame buffer.

1.1.4. HOST BUS INTERFACE

The Power 9100 connects directly to PCI and VESA local buses. With small amounts of glue logic it can be connected directly to processor buses.

The Power 9100 itself runs asynchronously to its bus clock. It supports both big-endian and little-endian address formats.

The Power 9100 is memory-mapped; each command has a unique memory address. In many cases, this means that command and data are given in a single write operation: the address specifies what operation is to be performed on the data being transferred.

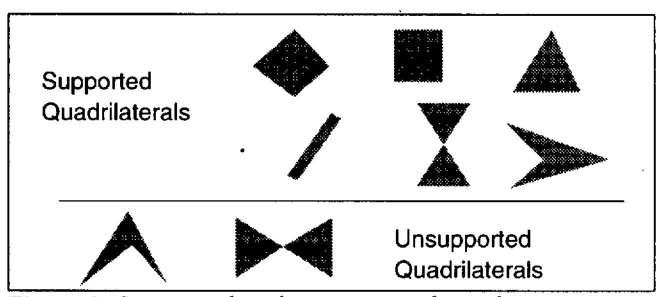


Figure 2. Supported and unsupported quads

1.1. Architecture, continued

1.1.5. VIDEO CONTROLLER

A typical board design feeds the VRAM shift registers into a RAMDAC, which converts digital pixel data into an analog RGB video signal. The host initializes the control registers and look-up tables in the RAMDAC through the Power 9100's RAMDAC access instructions.

The Power 9100 also generates horizontal and vertical synchronization and blanking signals, and controls the clocking of the video data. The timing of these signals is programmed through the video controller registers. The Power 9100's divided dot clock is completely asynchronous to the Power 9100's main system clock.

The Power 9100 supports 32- and 64-bit RAMDACs, such as the IBM RGB525 and the Brooktree Bt 485. By supporting RAMDACs with wide input buses at speeds of up to 200 MHz, the Power 9100 has no difficulty supporting large screens at high color depths and ergonomic refresh rates.

1.1.6. SVGA UNIT

For compatibility with older applications, the Power 9100 contains an on-chip SVGA. This SVGA unit is independent of the main graphics engine of the Power 9100, although the two share bus, frame buffer, and video interfaces. Control can be switched between the two graphics units under software control.

1.2. Major Differences Between the Power 9100 and the Power 9000

The Power 9100 builds upon the strengths of its predecessor, the Power 9000. This section summarizes the most important differences between the two.

1.2.1. ADDITIONAL FEATURES

1. On-chip VESA Local (VL) Bus interface
2. On-chip PCI Bus interface
3. On-chip SVGA unit
4. Native and SVGA modes
5. Full acceleration of 16- through 32-bit graphics
6. Higher clock speeds
7. Four-color pattern RAM (but reduced from 16x16 to 8x8) can be used for four-color dither at full speed
8. Aperture memory mapping (frame buffer can be mapped into a 64 KB aperture in addition to its usual 4 MB linear mapping)
9. Video coprocessor support
10. ROM BIOS control logic
11. Configuration EEPROM control logic
12. Clock generator control logic
13. Serial clock and serial enable generation for VRAMs
14. Text transparency
15. 256 raster-ops
16. Support for 4 MB frame buffer
17. 50 MHz operation

1.2.2. ADDRESSING

Power 9100 addressing and general address formats are different from those on the Power 9000.

1.3. Details of Graphics Operation

1.3.1. THE GRAPHICS PIPELINE

Graphics operations flow through the *graphics pipeline,* first through the parameter engine, and then through the stages of the drawing engine.

THE PARAMETER ENGINE

Functions. The parameter engine determines what will happen as the result of each drawing operation. It calculates status information, including whether the operation is clipped by the viewing window or the screen edge, and whether the operation can be drawn at all before passing the operation to the drawing engine. If the drawing engine is busy, or if the request contains illegal parameters, the parameter engine does not pass on the request. (The parameter engine performs these tests only on coordinate register loads and drawing commands. Other operations, such as setting color registers or video timing registers, bypass the parameter engine.)

The parameter engine performs clipping calculations on each x,y vertex. It compares these points against all four edges of the screen and viewing window, and against each other. It also tests for trivial rejection and trivial acceptance.

The parameter engine detects, but cannot correct an illegal request, such as a request for a horizontally convex quad. Such quadrilaterals must be rendered in software. The status register flags such problems. In addition, the interrupt signal can be used to interrupt the host when such exceptions occur.

Access. All accesses to parameter engine functions and registers complete in a few cycles; the parameter engine is always accessible.

THE DRAWING ENGINE

Functions. The drawing engine accepts one drawing operation at a time from the parameter engine. When processing a drawing operation, it first determines which pixels are "touched" by the drawing process *(scan conversion)*, then determines the color value of each touched pixel *(raster ops)*.

Scan conversion. Two line-drawing engines follow the left and right edges of the quadrilateral on each scan line, and a pixel-processing engine fills the region between these edges. (Pixel1, pixel8, and blit operations are limited to rectangular areas, while the quad operation can have edges at any angle).

Raster ops. The color of each touched pixel is determined by the raster-op function, which is further conditioned by the contents of other registers (see section 1.3.5).

Access. The drawing engine can remain busy for long periods when drawing large quads or performing a blit operation on a large portion of the screen. The quad and blit operations are started by a read operation. The read requests that the operation take place, and returns the contents of the status register, which indicates whether or not the request has been granted. The Power 9100 does not accept a request if the drawing engine is busy, or if an exception has occurred, nor does it queue requests. Therefore, the software must check the status register, which contains both exception and drawing engine status bits, and resubmit any quad or blit request that is not accepted. While no new drawing operations can be started until the drawing engine is idle, the host can load the parameter engine's coordinate registers with the vertices of a new operation while the drawing engine is busy, thereby overlapping the processing of two objects.

1.3.2. DRAWING QUADRILATERALS

The drawing engine assigns the two edges at either the top-most or bottom-most vertex to its two Bresenham *line-drawing engines.* These engines do not actually draw anything in the frame buffer, but they traverse the boundaries of the quadrilateral on each scan line. The *pixel-processing engine* then fills in the pixels between the boundaries found by the line-drawing engines. (This filling operation is also clipped against the clipping window and screen boundaries.) See figure 3. When a line-drawing engine reaches another vertex, it starts down the new edge.

In *oversized* mode, the Power 9100 draws perimeter pixels according to the Bresenham algorithm. Oversized mode must be selected to draw points and lines, as X11's drawing rules do not "touch" pixels in zero-width objects (meaning nothing is drawn).

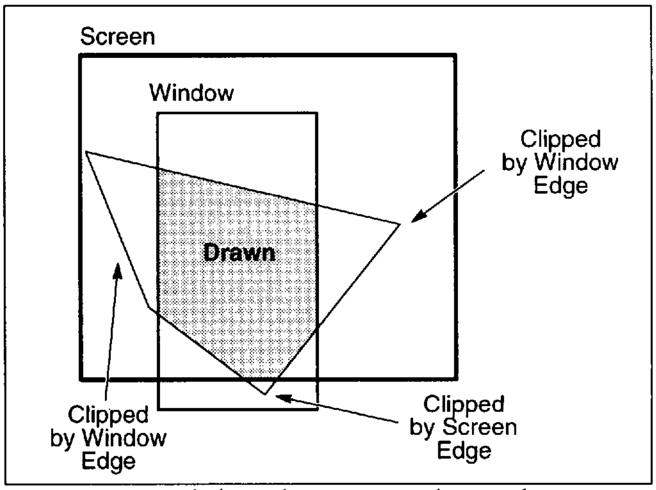


Figure 3. A quad clipped against window and screen

1.3. Details of Graphics Operation, continued

In *X11 mode,* the Power 9100 draws only those pixels whose centers lie within the perimeter of the quad. Pixels whose centers lie precisely on the perimeter are drawn in accordance with X11's tie-breaker rules. (See figure 4.) Coordinates can either be specified relative to the clipping window or relative to the previous vertex.

Polylines, meshed triangles, and meshed quadrilaterals are all supported; they are drawn by transferring the new vertices and issuing another draw command.

1.3.3. DRAWING BIT MAPS

To transfer a bit map to the screen, the host first sets up the *x,y* coordinates of the destination, and then transfers data to the Power 9100 with the pixel1 and pixel8 operations. The Power 9100 draws the pixels on the screen, auto-incrementing the current *x,y* location after each pixel. Bit-map drawing proceeds from left to right; when it reaches the right-hand edge of the target bit map, the Power 9100 automatically wraps to the next line.

The pixel8 operation draws colored pixels; up to four eight-bit pixels (or two 16-bit pixels, or one 32-bit pixel) are transferred and drawn through a single bus transfer from the host. This mode is used to transfer color images.

The pixel1 operation, which draws monochrome bit maps, is ideal for text transfer. Its basic operation is similar to pixel8, but instead of transferring, say, four eight-bit pixels per operation, it transfers up to 32 one-bit pixels. The Power 9100 expands the one and zero bits of the data word into pixels at the current color depth.

Pixel data must be padded out to multiples of 32 bits per scan line for pixel8 transfers; leftover pixels in the current word do not wrap to the next line. Pixel1 allows the left-over pixels to wrap, however, making it especially suitable for sending narrow blocks of monochrome data, such as character bit maps.

1.3.4. BIT BLOCK TRANSFER

The blit operation moves a rectangular block of pixels from one part of the screen to another. It handles overlapping source and destination blocks properly; the source block arrives unchanged at the destination, as if it were moved off-screen, then copied back at its new destination.

Like quad, blit is a "fire-and-forget" operation; once initiated, it runs to completion without additional attention from the host. As a large blit can involve over a million pixels, the host driver code should test the busy bit in the Power 9100's status register before attempting another drawing operation when a blit could be in progress.

1.3.5. COLOR SELECTION

Once a pixel has been touched, there still remains the question of what color it will be. Screen color is selected at five levels:

1. the initial (source) color
2. the pattern color
3. the raster-op color
4. the plane mask
5. the RAMDAC look-up table color (8-bit modes only; higher pixel depths use direct color)

The Power 9100 applies each of these in turn. The *pattern RAM* allows imposition of an 8x8-bit repeating pattern on the data. See figure 5. The *raster-op* function is a three-input boolean function controlled by an eight-bit minterm array. The three inputs are:

1. The source color
2. The destination color (the color value currently at the *x,y* location being written)
3. The color registers color[3..0].

The Power 9100 applies the same function to each of the bits in the pixel.

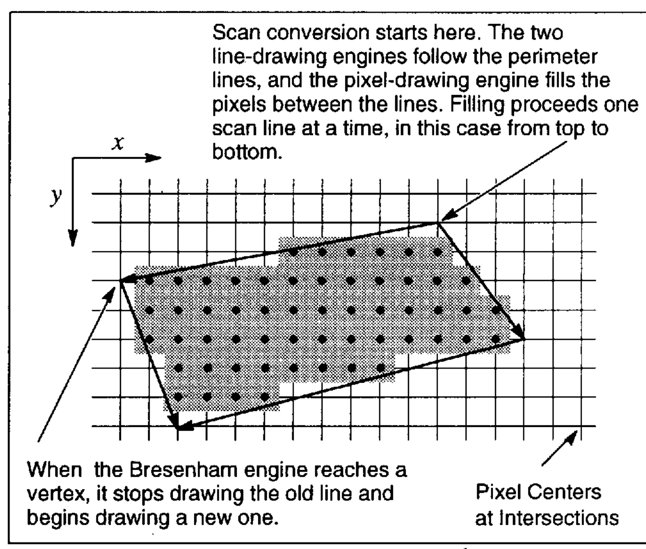


Figure 4. Scan conversion using X11 rules

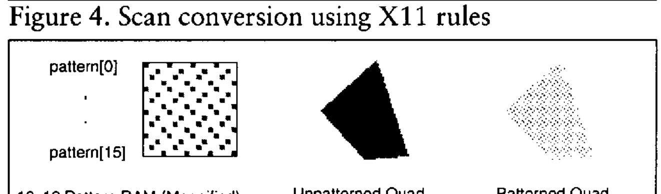


Figure 5. Patterning

1.4. Host Bus Interface

The Power 9100 supports the VL and PCI buses directly, with no glue logic. See figures 6 through 9. Other buses can be accommodated with a small amount of external glue logic.

Some pins on the Power 9100 change function depending on which bus is selected. Rather than introduce a confusing third signal nomenclature to that of PCI and VL, we have provided two pin configuration diagrams: one for each bus, each using that bus' signal naming conventions. See section 13.4 for the pin configuration.

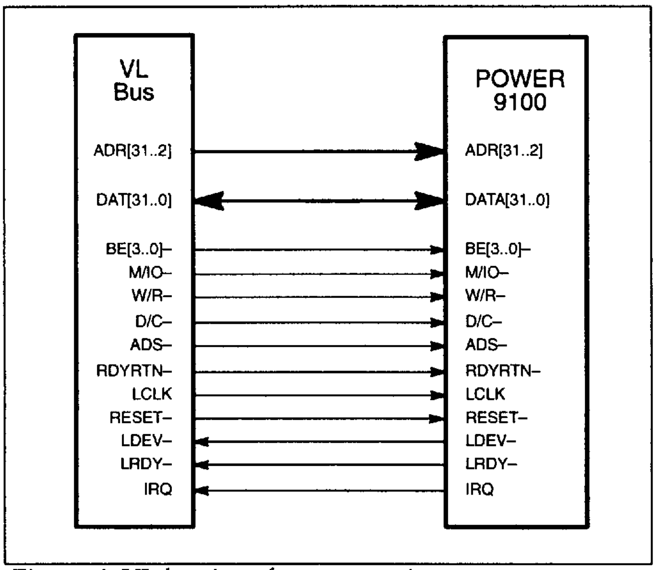


Figure 6. VL bus interface connection

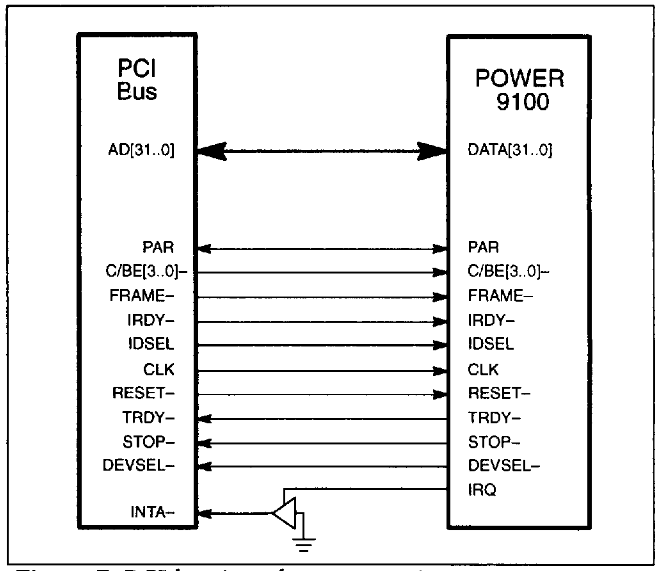


Figure 7. PCI bus interface connection

VL Bus		
Signal	**I/O**	**Description**
DATA[31..0]	I/O	Data bus
ADR[31..2]	Input	Address bus
BE[3..0]–	Input	Byte enable
W/R–	Input	Write or read status
M/IO–	Input	Memory or I/O status
ADS–	Input	Address data strobe
LDEV–	Output	Local device
LRDY–	Output	Local ready
RDYRTN–	Input	Ready return
IRQ	Output	Interrupt request
LCLK	Input	VL clock
RESET–	Input	Reset
D/C–	Input	Data or code status

Figure 8. VL bus interface signals

PCI Bus		
Signal	**I/O**	**Description**
DATA[31..0]	I/O	Address and data bus
C/BE[3..0]–	Input	Bus command/byte enable
PAR	I/O	Parity
IDSEL	Input	Initialization device select
STOP–	Output	Stop
FRAME–	Input	Cycle frame
DEVSEL–	Output	Device select
TRDY–	Output	Target ready
IRDY–	Input	Initiator ready
IRQ	Output	Interrupt request
CLK	Input	PCI clock
RESET–	Input	Reset

Figure 9. PCI bus interface signals

1.5. System Block Diagram and Signal Descriptions

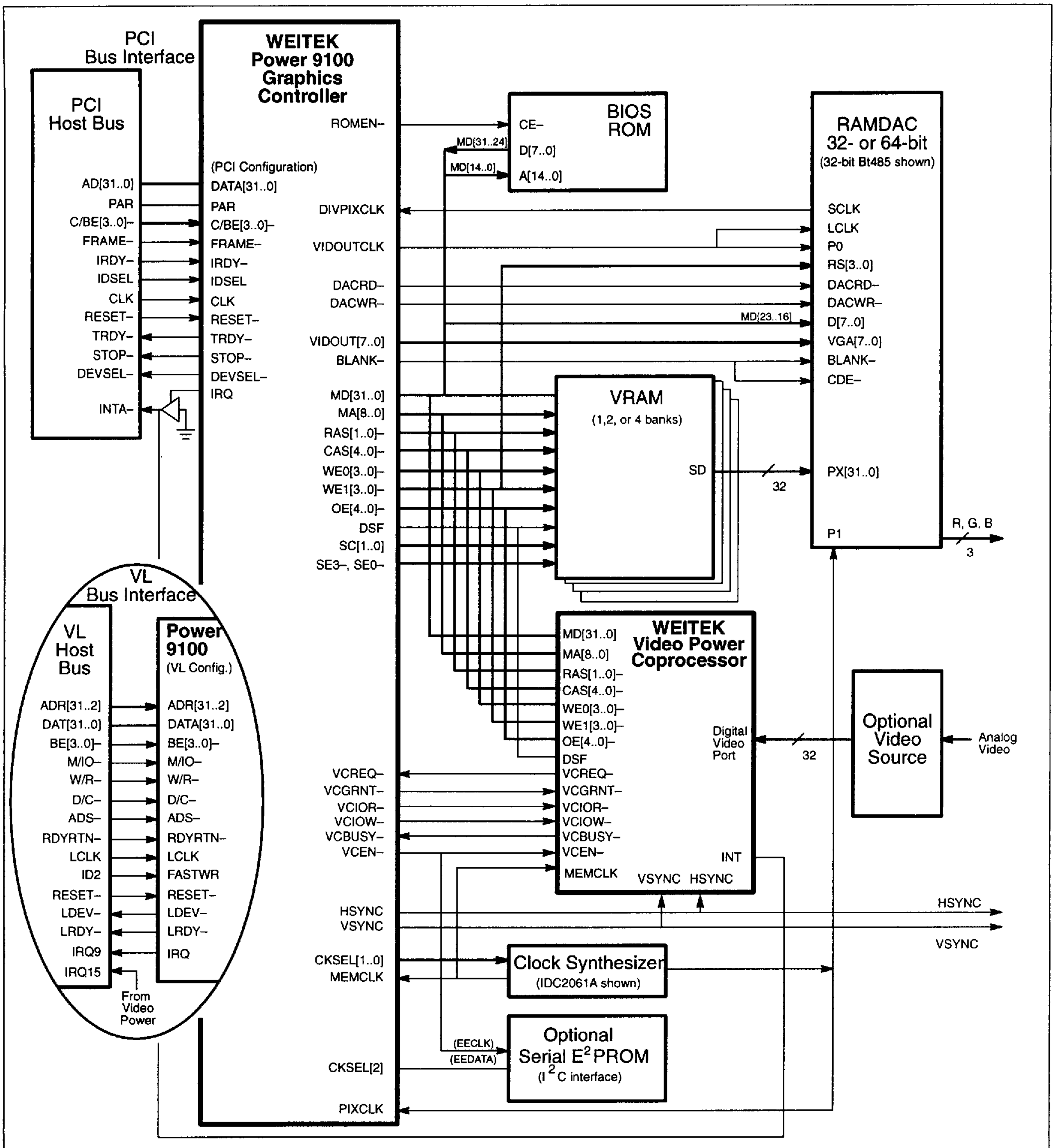


Figure 10. Power 9100 system block diagram

1.5. System Block Diagram and Signal Descriptions, continued

1.5.1. SIGNAL DESCRIPTIONS

Some signals have multiple uses; for example, EECLK and VCEN– share the same pin.

Signal	Type	Description
MD[31..0]	I/O	Memory data bus
MA[8..0]	Output	Memory address bus
RAS[1..0]–	Output	Row address strobes
CAS[4..0]–	Output	Column address strobes. CAS[0]– controls bits [31..16] of bank 0; CAS[4]– controls bits [15..0] of bank 0 (necessary for VGA transfers). Note that in two-bank systems, bank 1 is controlled by CAS[3]–, not CAS[1]–
WE0[3..0]–	Output	Write-enable for the individual bytes in bank 0 (2-bank systems), or banks 0 and 2 (4-bank systems)
WE1[3..0]–	Output	Write-enable for the individual bytes in bank 1 (2-bank systems), or banks 1 and 3 (4-bank systems)
OE[4..0]–	Output	Output Enables. OE[0]– controls the least-significant 16 bits of bank 0; OE[4]– controls the most-significant bits of bank 0 (necessary for VGA transfers). Note that in two-bank systems, bank 1 is controlled by OE[3]–, not OE[1]–
DSF	Output	VRAM special function pin
DACRD–	Output	RAMDAC read control signal
DACWR–	Output	RAMDAC write control signal
MEMCLK	Input	Main chip clock
ROMEN–	Output	BIOS ROM enable
EECLK	Output	EEPROM clock control signal (shared with VCEN–)
EEDATA	I/O	EEPROM data control signal (shared with CKSEL[2])
VDDPLL	Input	Supply voltage for on-chip clock generator
VSSPLL	Output	Ground for on-chip clock generator

Figure 11. Memory control signals

Signal	Type	Description
SE[3]–, SE[0]–	Output	VRAM serial shift enable
SC[1..0]	Output	VRAM serial shift clock
VIDOUT[7..0]	Output	Video data out (SVGA modes). VIDOUT[0] and VIDOUT[1] must be pulled up to VCC with 10 KΩ resistors
VIDOUTCLK	Output	Video data clock out
HSYNC	I/O	Horizontal synchronization. Normally an output; can also be used as an input for external sync
VSYNC	I/O	Vertical synchronization. Normally an output; can also be used as an input for external sync
BLANK–	Output	Blanking interval
PIXCLK	Input	Pixel clock
DIVPIXCLK	Input	Divided pixel clock
CKSEL[2..0]	Output	Frequency synthesizer control (shared with SCLK, SDATA, and EEDATA)

Figure 12. Video control signals

1.5. System Block Diagram and Signal Descriptions, continued

Signal	Type	Description
VCBUSY–	Input	Video coprocessor busy. Video coprocessor is busy; I/O reads and writes will not succeed. This signal should be pulled up to VCC with a 10 KΩ resistor
VCEN–	Output	Video coprocessor enable. When asserted, video coprocessor is active. Resets and deactivates video coprocessor when deasserted (shared with EECLK). This signal must be pulled up to VCC with a 10 KΩ resistor.
VCGRNT–	Output	Video coprocessor bus grant. Signals that the Power 9100 has released the frame buffer
VCIOR–	Output	Video coprocessor I/O read. Requests a read from a video coprocessor register
VCIOW–	Output	Video coprocessor I/O write. Requests a write to a video coprocessor register
VCREQ–	Input	Video coprocessor bus request. Requests that the video coprocessor be given control of the bus. This signal should be pulled up to VCC with a 10 KΩ resistor

Figure 13. Video coprocessor signal description

1.6. Video Coprocessor Interface

The video coprocessor interface allows a separate coprocessor to share the Power 9100's frame buffer and host interface. Such a coprocessor could provide features to accelerate still pictures, video animation, or 3-D rendering.

The video coprocessor interface allows the host to read and write data from the video coprocessor, while the video coprocessor grant and release functions allow the video coprocessor to take and relinquish control of the frame buffer. The video coprocessor pre-empt function lets the Power 9100 regain control to perform high-priority tasks such as memory refresh.

1.7. Related Documents

For availability of Power 9100 documentation, see your WEITEK sales representative. Upcoming titles include:

Power 9100 Application Notes. Practical techniques for Power 9100 design, including examples of complete board designs for both VL and PCI.

Power 9100 Manufacturing Kits. Artwork, programmable logic equations, and manufacturing drawings for tested, cost-effective Power 9100 designs.

Chapter 2. Quick Reference

2.1. Ordering Information

Package Type	Speed Grade	Temperature Range (Case)	Order Number
208-pin PQFP	50 MHz	0 – 85°C	P9100–050–PFP

Figure 14. Ordering information

2.2. Address Formats for Native Mode Registers and Commands

Address Format	Described in Section	Page
RAMDAC access	3.4.5	41
Video coprocessor access	3.4.6	41
System control registers	4.3	47
Video control registers	4.6	72
VRAM control registers	4.7	77
Status register	4.4.2	57
Parameter engine registers	4.4	54
Drawing engine registers	4.5	66
Load coordinate pseudo-registers	5.1	82
Quad command	5.2	83
Blit command	5.3	84
Pixel8 command	5.4	85
Pixel1 command	5.5	87
Next_pixels command	5.6	88

Figure 15. Address formats for native mode registers and commands

2.3. General Address Formats

Prefix*		Address Bits													Specific Address	Section
31	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
		0	0	0	0	0	0	0	0	–	–	–	–	–	System control registers	4.3
		0	0	0	0	0	0	1	0	–	–	–	–	–	Video control registers	4.6
		0	0	0	0	0	0	1	1	–	–	–	–	–	VRAM control registers	4.7
		0	0	0	0	0	1	0	0	0	–	–	–	–	RAMDAC control	3.4.5
		0	0	0	0	1	–	–	–	–	–	–	–	–	Video Coprocessor Interface	3.4.6
		0	1	0	0	0	0	0	0	0	0	0	1	1	Pixel8 command (Power 9000 format)	5.4
		1	–	–	–	–	–	–	–	–	–	–	–	–	Pixel8 command (Power 9100 format)	5.4
		0	1	0	0	0	0	0	0	0	0	1	0	1	Next_pixels command	5.6
		0	1	0	0	0	0	0	1	–	–	–	–	–	Pixel1 command	5.5
		0	1	0	0	0	0	1	1	–	–	–	–	–	Parameter engine control registers	4.4.3
		0	1	0	0	0	1	–	–	–	–	–	–	–	Drawing engine pixel processing registers	4.5
		0	1	1	0	0	0	0	–	–	–	–	–	–	Device coordinate registers	4.4.1
		0	1	1	0	0	1	–	–	–	–	–	–	–	Load coordinates pseudo-registers	5.1

* See figures 50 and 51.

Figure 16. Decoding of addresses for write operations

Prefix*		Address Bits													Specific Address	Section
31	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
		0	0	0	0	0	0	0	0	–	–	–	–	–	System control registers	4.3
		0	0	0	0	0	0	1	0	–	–	–	–	–	Video control registers	4.6
		0	0	0	0	0	0	1	1	–	–	–	–	–	VRAM control registers	4.7
		0	0	0	0	0	1	0	0	0	–	–	–	–	RAMDAC control	3.4.5
		0	0	0	0	1	–	–	–	–	–	–	–	–	Video Coprocessor Interface	3.4.6
		0	1	0	0	0	0	0	0	0	0	0	0	0	Status register	4.4.2
		0	1	0	0	0	0	0	0	0	0	0	0	1	Blit command	5.3
		0	1	0	0	0	0	0	0	0	0	0	1	0	Quad command	5.2
		0	1	0	0	0	0	1	1	–	–	–	–	–	Parameter engine control registers	4.4.3
		0	1	0	0	0	1	–	–	–	–	–	–	–	Drawing engine pixel processing registers	4.5
		0	1	1	0	0	0	0	–	–	–	–	–	–	Device coordinate registers	4.4.1
		0	1	1	0	0	1	–	–	–	–	–	–	–	Load coordinates pseudo-registers	5.1

* See figures 50 and 51.

Figure 17. Decoding of addresses for read operations

2.4. Native Mode Register Summary

Subgroup	Register	Name/Function	Accessed via	Effect of reset	See section
System Control Registers					
	sysconfig	System configuration. Specifies system configuration information.	System control register access address format; see figure 58	set to zero (Host Reset)	4.3.1
	interrupt	Interrupt. Records interrupt conditions.		set to zero	4.3.2
	interrupt_en	Interrupt enable. Enables interrupts upon occurrence of interrupt conditions.		set to zero	4.3.3
	alt_read_bank	Alternate bus read bank select		set to zero	4.3.4
	alt_write_bank	Alternate bus write bank select		set to zero	4.3.5
Parameter Engine Registers					
Device coordinate	X[0],Y[0] X[1],Y[1] X[2],Y[2]] X[3],Y[3]	Device coordinate. Supplies screen coordinates for drawing operation.	Device coordinate register access address format; see figure 65	Not changed	4.4.1
Status	status	Status. Read only. Records status of drawing engine and coordinate register clip checks.	Status register access address format; see figure 67	Not changed	4.4.2
Control and condition	oor	Out of range. Read only. Records out of range x,y values.	Parameter engine control and condition register access address format; see figure 70	Not changed	4.4.3
	cindex	Index. Supplies current index into x and y coordinates		Not changed	
	w_off.x/y	Window offset. Supplies offset of current window on the display.		Not changed	
	p_w_min p_w_max	Parameter engine window minimum, parameter engine window maximum. Read only. Record the contents of the drawing engine window minimum (w_min) and window maximum (w_max) registers.		Not changed	
	xclip yclip	Xclip, yclip. Read only. Record results of clip checks on x and y coordinates.		Not changed	
	xedge_lt xedge_gt yedge_lt yedge_gt	Vertex relationship. Read only. Record results of vertex relationship checking.		Not changed	

Figure 18. Native register summary (1 of 3)

2.4. Native Register Summary, continued

Subgroup	Register	Name/Function	Accessed via	Effect of reset	See section
Drawing Engine Registers					
Pixel processing	color[0] color[1] color[2] color[3]	Color register 0, foreground Color register 1, background Color register 2 Color register 3	Drawing engine pixel processing register access address format; see figure 79	Not changed	4.5.1
	pmask	Plane mask. Specifies plane mask.		Not changed	4.5.2
	draw_mode	Draw mode. Specifies control for writing within a picked window and selects the destination buffer for drawing operations.		Not changed	4.5.3
	pat_originx pat_originy	X pattern origin, y pattern origin. Specify x and y screen coordinates for pattern origin.		Not changed	4.5.4
	raster	Raster. Specifies minterms, transparency and other parameters for a raster operation.		Not changed	4.5.5
	pixel8_reg	Pixel8. Stores excess pixel8 operation data bits.		Not changed	4.5.6
	p_w_min b_w_min	Window minimum. Specifies minimum x,y values for window.		Not changed	4.5.7
	p_w_max b_w_min	Window maximum. Specifies maximum x,y values for window.		Not changed	4.5.7
	pattern[1] pattern[2] pattern[3] pattern[4]	Pattern. Specify pattern.		Not changed	4.5.8
	user[0] user[1] user[2] user[3]	User defined registers		When modes switch, the value is preserved; value not preserved on reset.	4.5.9

Figure 56, continued. Native register summary (2 of 3)

2.4. Native Register Summary, continued

Subgroup	Register	Name/Function	Accessed via	Effect of reset	See section
Video Control Registers					
Horizontal	hrzc	Horizontal counter. Read only.	Video control register access address format; see figure 85	Set to zero	4.6
	hrzt	Horizontal length.		Set to FFFh	
	hrzsr	Horizontal sync rising edge.		Set to FFFh	
	hrzbr	Horizontal blank rising edge.		Set to FFFh	
	hrzbf	Horizontal blank falling edge.		Set to FFFh	
	prehrzc	Horizontal counter preload.		Set to zero	
Vertical	vrtc	Vertical counter. Read only.		Set to zero	
	vrtt	Vertical length.		Set to FFFh	
	vrtsr	Vertical sync rising edge.		Set to FFFh	
	vrtbr	Vertical blank rising edge.		Set to FFFh	
	vrtbf	Vertical blank falling edge.		Set to FFFh	
	prevrtc	Vertical counter preload.		Set to zero	
Repaint	srtctl	Screen repaint timing control.		Set to zero	
	srtctl2	Screen repaint timing control.		Set to FFFh	
	qsfcounter	QSF counter.		Set to zero	
VRAM Control Registers					
	mem_config	Memory configuration.	VRAM control register access,address format; see figure 90	Set to zero	4.7.1
	rfperiod	Refresh period.		Set to 0x3FF	4.7.2
	rfcount	Refresh counter.		Set to zero	4.7.3
	rlmax	RAS low maximum.		Set to 0x3FF	4.7.4
	rlcur	RAS low current.		Set to zero	4.7.5
	pu_config	Power-up configuration		Preserves 32-bit value on the frame buffer data bus at rising edge of reset	3.3.3

Figure 56, continued. Native register summary (3 of 3)

2.5. Command Summary

Command	Function	Described in section
Load Coordinates	Load quad coordinates via shorthand method	5.1
Quad	Draw a quadrilateral.	5.2
Blit	Copy a rectangular area of the display form one screen location to another.	5.3
Pixel8	Transfer one word of pixel data from a linear host memory array to a rectangular display memory array in the frame buffer.	5.4
Pixel1	Transfer up to 32 pixels from a a linear host memory array to a rectangular display memory array in the frame buffer.	5.5
Next_pixels	Advance the drawing area for a pixel8 or pixel1 operation.	5.6

Figure 19. Command summary

2.6. VGA/WEITEK Register Groups

Group	Port	Function	Use	Mode	Fields	Section
General	94	Enable	Color, monochrome	Read, write	VGA (Motherboard)	12.4
	102	Enable	Color, monochrome	Read, write	VGA (Adaptor)	12.4
	3BA	Data	Monochrome only	Read, write	VGA	12.4
	3C2	Data	Color, monochrome	Read, write	VGA	12.4
	3C3	Enable	Color, monochrome	Read, write	VGA (Motherboard)	12.4
	3C7	Data	Color, monochrome	Read, write	VGA	12.4
	3CA	Data	Color only	Read only	VGA	12.4
	3CC	Data	Color, monochrome	Read only	VGA	12.4
	3CD	Data	Color, monochrome	Read, write	Power 9100	12.4
	3DA	Data	Color only	Read, write	VGA	12.4
	46E8	Enable	Color, monochrome	Read, write	VGA (Adaptor)	12.4
Sequencer	3C4	Index	Color, monochrome	Read, write	VGA, Power 9100	12.5
	3C5	Data	Color, monochrome	Read, write	VGA, Power 9100	12.5
CRT controller	3B4	Index	Monochrome only	Read, write	VGA	12.6
	3B5	Data	Monochrome only	Read, write	VGA, Power 9100	12.6
	3D4	Index	Color only	Read, write	VGA	12.6
	3D5	Data	Color only	Read, write	VGA, Power 9100	12.6
Graphics controller	3CE	Index	Color, monochrome	Read, write	VGA, Power 9100	12.7
	3CF	Data	Color, monochrome	Read, write	VGA, Power 9100	12.7
Attribute controller	3C0	Index and data	Color, monochrome	Write only	VGA, Power 9100	12.8
	3C1	Index and data	Color, monochrome	Read only	VGA, Power 9100	12.8

Figure 20. VGA/WEITEK register groups

2.7. Major Differences between the Power 9100 and the Power 9000

Registers	New/Changed/ Replaced	Notes	Reference
Configuration registers	New	These registers specify device and vendor identifiers; enable the RAMDAC pallet, memory space access, I/O space access, interrupts; specify base addresses; enable VGA; and select P9100 or SVGA mode.	Section 3.3
Color (color[0] through color[3])	New	color[0] replaces Power 9000 foreground register; color[1] replaces Power 9000 background register. Both are used in two-color patterns. Color[2] and color[3] are used in new four-color patterns. Each register is 32 bits wide	Section 4.5.1
Pattern (pattern[0] through pattern[3])	New	Replace Power 9000 pattern registers. (Each register is 32 bits wide.)	Section 4.5.8
alt_read_bank alt_write_bank	New	These registers specify the high-order address bits (in register bits 22–14) when alternate aperture frame buffer banking logic is used to read/write directly from the frame buffer	Section 4.3.4 and4.3.5
User registers (4)	New	These registers are reserved for the system software to use.	Section 4.5.9
SVGA	New	The Power 9100 has all of the registers found on the W5286 on its SVGA unit; they are identical to the W5286 registers.	Chapter 12
System configuration (sysconfig)	Changed	New fields: bits 30–29 specify additional shift control bits 28–26 specify 8, 16, 24, or 32 bits per pixel for the drawing engine Deleted field: bit 25 is reserved	Section 4.3.1
Memory configuration (mem_config)	Changed	Replaces Power 9000 mem_config register	Figure 91
Raster	Changed	New fields: bit 15 enables pixel1 transparent mode bit 14 specifies pattern depth (2-color or 4-color pattern) bit 13 specifies solid color disable/enable in pattern register Changed fields: bit 17 enables transparent pattern bits 7–0 minterms (bits 15–0 on Power 9000) bits 12–8 now always zero	Section 4.5.5
Foreground (fground) Background (bground)	Replaced	Replaced by new Power 9100 color registers.	
Pattern	Replaced	Replaced by new Power 9100 pattern registers.	

Figure 21. Register differences between the Power 9100 and the Power 9000

Chapter 3. Memory Map

3.1. Overview

The host controls the Power 9100 by setting registers, loading pseudo-registers, and issuing commands. Power 9100 *registers*, most of which are 32 bits wide, contain the parameters for Power 9100 operations and maintain status information. *Pseudo-registers* minimize the number of parameters that must be specified per drawing operation. *Commands* initiate the drawing operations whose parameters are defined by register and pseudo-register settings.

The Power 9100 is memory-mapped: The address is divided into control fields that determine the actions to be performed. The sections of memory reserved for Power 9100 addresses must be marked as non-cacheable; because the Power 9100 alters its own register contents, memory caching would give invalid data. Not all memory locations in the native register memory space are used. Do not access addresses with undefined functionality.

The Power 9100 performs all tasks in the order is which they are issued.. The host must ensure that requests for drawing operations are accepted, either by checking Power 9100 status to verify that the previous drawing operation has completed before issuing a new request, or, for operations that return status, by checking the status of a requested operation before issuing a new request. The status register contents reflect status conditions, as defined in section 4.4.2. The host processor is also responsible for proper byte alignment of data it sends across the bus to the Power 9100.

The SVGA unit is enabled separately and uses standard VGA I/O mapping.

This chapter describes all Power 9100 address formats. Subsequent chapters describe the Power 9100 configuration registers, control registers, pseudo-registers, and commands, and define the specific address format for accessing each. For illustrations and examples of Power 9100 operation, see Chapter 5.

3.1. Overview, continued

3.1.1. BIG-ENDIAN AND LITTLE-ENDIAN MODES

The Power 9100 data bus connects to the host bus so that transfer of a 32-bit word preserves the significance of the data bits (bit 0 is the least significant bit and bit 31 is the most significant bit). Because the Power 9100 stores all data in big-endian mode, it is necessary to convert the mode when transferring data to and from a host system that uses little-endian mode. The H, B, and b bits enable swapping of half-words, bytes, and bits respectively. This is used most often when accessing the frame buffer. Figure 22 illustrates the swapping effect of setting the H, B, and b bits. The H, B and b bits are derived from different registers depending on the address format used.

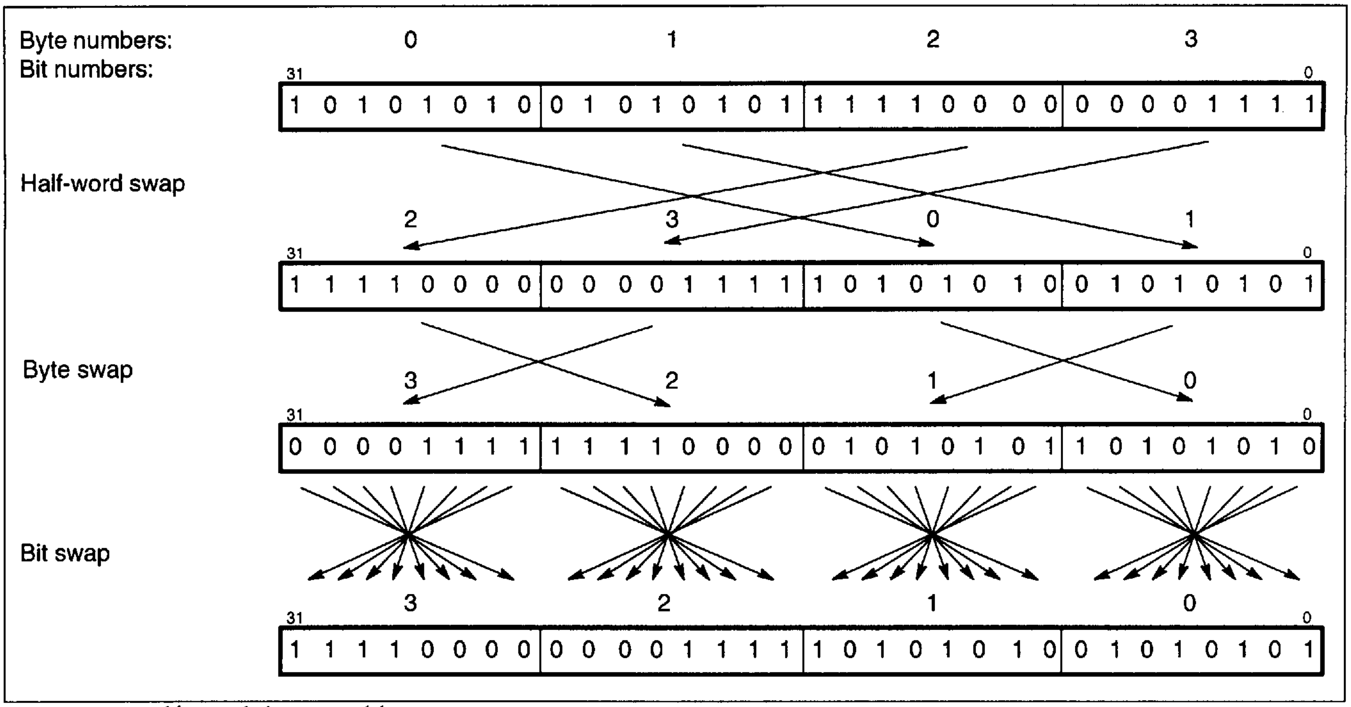


Figure 22. Half-word, byte, and bit swapping

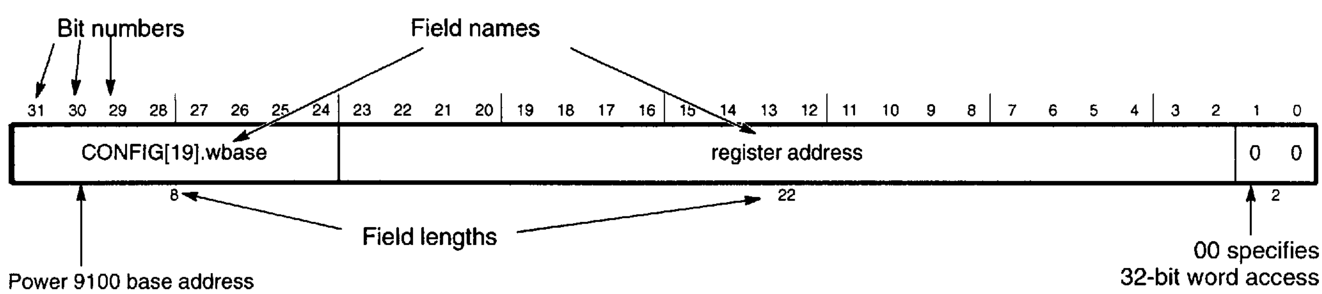
3.2. Conventions and Notation

The conventions and notation defined in this section are used throughout this chapter.

All discussions of registers and address formats define all 32 bits. Illustrations appear as shown in the example in figures 23 and 24, with bit numbers identified across the top of the illustration and field sizes called out below. The field contents are identified in the illustration and defined in a table that accompanies the illustration.

The prefix "c." on a field name indicates that the most-significant bit in the field is a field write control bit. When writing to a register, it is not always desirable to replace the entire register contents. For example, clearing a bit in the interrupt register is a function that needs to be done without altering the information contained in other register fields. When writing to a register, setting a field write control bit to 1 replaces that field with the new data; setting a field write control bit to 0 leaves the contents of the field unaltered. On a register read, field write control bits are always 1, which makes it easy to save and restore all of the fields of a register.

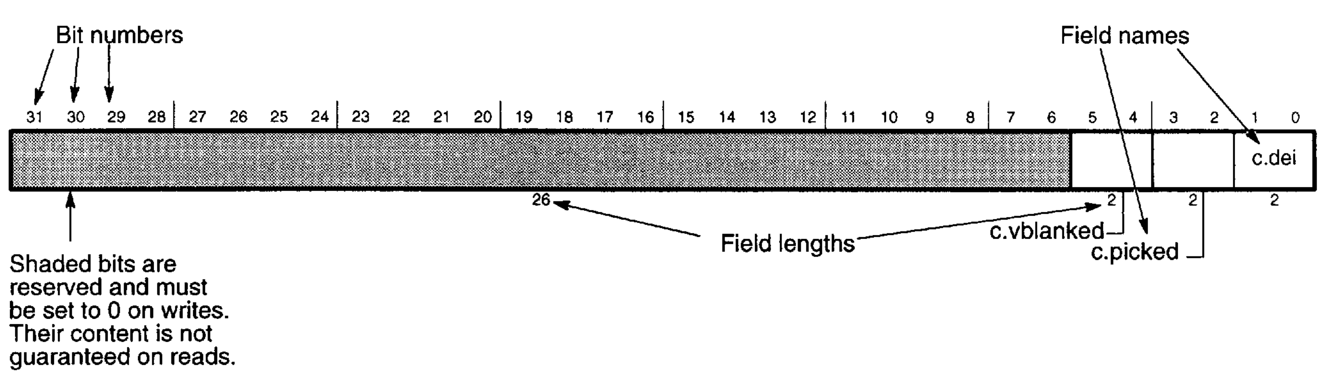
Sample Address Format Illustration



Field	Bits	Contents
field name	X–X	Description of field

Figure 23. Sample address format illustration

Sample Register Data Format Illustration



Field	Bits	Contents
field name	X–X	Description of field

Figure 24. Sample register data format illustration

3.3. Configuration Registers

3.3.1. PCI BUS MODE

The PCI bus specification requires that a group of configuration registers be present in the configuration address space. (See figure25.) These registers are used to control memory and I/O mapping as well as various other control functions. The Power 9100 fully supports the requirements of the PCI specification. These registers are accessed during configuration bus cycles when **IDSEL** is asserted according to the PCI spec.

Configuration registers that are not implemented follow the absent PCI configuration register convention — ignore all writes, read as all zeroes. All operations may be 8-, 16-, or 32-bits.

Byte 3 (bits 31–24)	Byte 2 (bits 23–16)	Byte 1 (bits 15–8)	Byte 0 (bits 7–0)	Offset
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
				0Ch
Display Memory Base Address				10h
				14h
				18h
				1Ch
				20h
				24h
				28h
				2Ch
Expansion ROM Base Address				30h
				34h
				38h
		Interrupt Pin	Interrupt Line	3Ch
	Power 9100 Device Specific			40h

Shaded registers are not implemented and read back 0's.

Figure 25. PCI configuration space header

3.3. Configuration Registers, continued

3.3.2. VL BUS MODE

When operating in the VL bus mode, the same functionality is supported by mapping the configuration registers into the I/O address space using register indexing. Two 32-bit words of I/O address space are used to access the configuration registers.The first is used as an index into the configuration address space, the second is used as the data transfer register. These configuration registers are present independent of the mode (VGA or native) that the Power 9100 is currently operating in. The index and data registers can be placed at one of several locations in the I/O address space. The actual location of these registers is controlled by the setting of the PU_CONFIG.CFGBA field. The value of this field is determined during the reset sequence (see section 3.3.3).

Configuration Index and Data Registers Address Format (VL Bus Mode)

31–16	15	14	13	12	11	10	9	8	7	6	5–3	2–0
	1	0	0	1	0	0	0	1	0	0	.cfgba	reg_type
16	10										13	3

Field	Bits	Contents
reserved	31–16	Reserved; these bits are ignored.
I/O base address	15–6	I/O base address, 91xxh
pu_config.cfgba	5–3	See figure 27, bits [29..27]
reg_type	2–0	000 = index register 100 = data register

Figure 26. Configuration index and data registers address format (VL bus mode)

3.3. Configuration Registers, continued

3.3.3. POWER-UP CONFIGURATION

The deassertion of RESET– forces the system to sample the frame buffer data bus to determine its power-up configuration. The VRAM MD[31..0] bus logic has built-in pull-down resistors causing the bus to settle to a low state. By placing (~10KΩ) pull-up resistors on the bus, selected bits can be pulled high. These initial settings of the data bus are preserved in the read-only PU_CONFIG register.

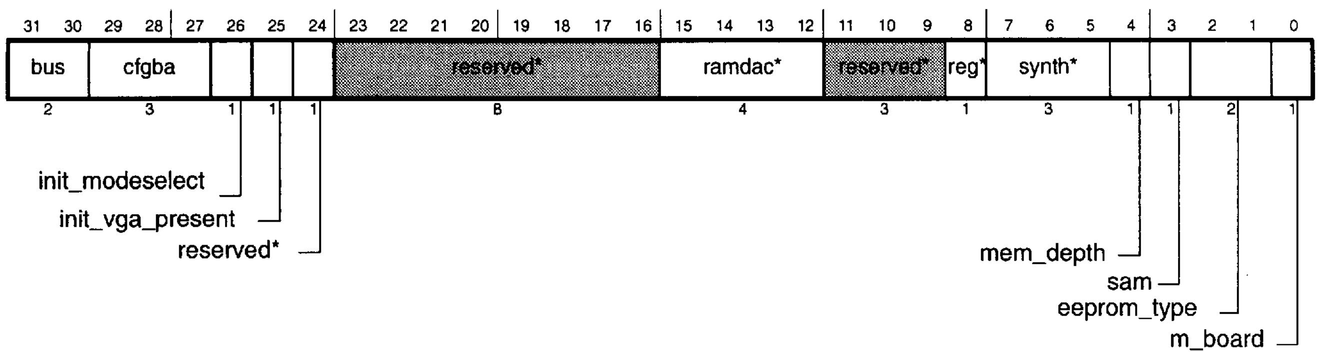
The Power9100's power-up configuration register is used to configure the controller and inform software about a board's configuration, and to identify a board's resources without referring to any additional external information, such as a file on a disk or an EEPROM. (The number of banks of memory, not specified in this register, can be determined through software.)

The power-up configuration register will be further redefined to support new clock generators and DACs as the need arises. Please contact WEITEK if you want to make additional bit assignments. We will make bit assignments as needed. We strongly suggest that you do not build a board requiring an additional encoding without contacting WEITEK.

Figure 27 illustrates and defines the power-up configuration register. The fields that affect the hardware are identified. The other fields are read-only bits defined by the pull-up resistors. The hardware fields are copied to a CONFIG register, where, in some cases, they become read/writeable.

3.3. Configuration Registers, continued

Power-up Configuration Bits (PU_CONFIG register) (Read-Only)
Native mode address offset = 0198h (see figure 90)



*Does not affect the chip's internal state. Used by software to determine system configuration information.

Field	Affects HW?	Bits	Contents
bus	✓	31–30	The bus signalling protocol. (Copied to CONFIG[64].BUS[7..6]) 00 = Reserved (internal Testing mode) 01 = PCI Bus 10 = VESA Local Bus 11 = Reserved
cfgba	✓	29–27	I/O address for configuration register index and data registers, VL mode only. (Copied to CONFIG[64][5..3]) pu_config.cfgba / config register index / config register data 000 / 9100h / 9104h 001 / 9108h / 910Ch 010 / 9110h / 9114h 011 / 9118h / 911Ch 100 / 9120h / 9124h 101 / 9128h / 912Ch 110 / 9130h / 9134h 111 / 9138h / 913Ch
init_modeselect	✓	26	The initial chip mode.(Copied to CONFIG[65].MODESELECT[1] 0 = Native mode 1 = Emulation mode (VGA)
init_vga_present	✓	25	Copied to CONFIG[10].VGA_PRESENT[7]. See figure 34. In PCI mode this setting changes the PCI sub-class code. In VL mode, this setting may affect BIOS enabling.

Figure 27. Power-up configuration bits (PU_CONFIG register)

3.3. Configuration Registers, continued

Field	Affects HW?	Bits	Contents
reserved	✓	24	Bit 24 must be set to zero or the WEITEK software will not work.
reserved		23–16	Software convention only Bits 23–16 cannot be used with certain types of RAMDACs and are reserved for future use. (Note: One customer is using bits [19..16] as a board identification field, having determined that the RAMDAC in their design is compatible with this usage.)
ramdac*		15–12	Software convention only RAMDAC type. 0000 = Bt 485-style RAMDAC, as well as variations identifiable through the RAMDAC status register (identify variations through the upper four bits of the status register*) 1000 = IBM525 (IBM 64-bit RAMDAC)
reserved		11–9	Software convention only Must be set to zero. Do not use these bits or WEITEK software will not work
reg		8	Software convention only External registers available. 0 = External registers are not implemented on the board 1 = External registers are implemented on the board and can be accessed through the RAMDAC address space when CKSEL[2] is set to one
synth		7–5	Software convention only Frequency synthesizer. 000 = ICD2061A, ICS9161, or compatible, or RAMDAC generates all clocks 001 = Fixed MEMCLK; RAMDAC generates PIXCLK. (This selection is currently intended to support IBM RGB525 designs wherein the MEMCLK is used as a reference clock for the RAMDAC PLL. It is assumed that the memory clock is running at 50 MHz, and, thus, the RAMDAC reference clock is also assumed to be 50 MHz.)
mem_depth	✓	4	Depth of memory chips. 0 = 256K 1 = 128K
sam	✓	3	VRAM SAM size. 0 = Full-sized shift registers 1 = Half-sized shift registers
eeprom_type	✓	2–1	00 = AT24C01 (ATMEL 128x8) or compatible (or not installed)
m_board	✓	0	Software convention only Motherboard VGA address decode control. 0 = On motherboard 1 = On add–in board

*At this time, WEITEK software does not check the RAMDAC status register. If you are considering using a Bt 485-compatible RAMDAC with additional features, contact WEITEK first so that we can determine whether the status register alone will be enough to determine your RAMDAC's unique requirements.

Figure 27, continued. Power-up Configuration bits (PU_CONFIG register)

3.3. Configuration Registers, continued

3.3.4. VENDOR ID REGISTERS

00h CONFIG[0] Vendor ID Register[7..0] (Read-Only)

7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	0

8

Field	Bits	Definition
	7–0	Low-order byte of Vendor ID register Value of this byte is 0Eh

Figure 28. CONFIG [0] vendor IDregister (read-only)

01h CONFIG[1] Vendor ID Register[15..8] (Read-only)

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0

8

Field	Bits	Definition
	7–0	High-order byte of Vendor ID register Value of this byte is10h

Figure 29. CONFIG[1] vendor ID register (read-only)

3.3. Configuration Registers, continued

3.3.5. DEVICE ID REGISTERS

02h CONFIG[2] Device ID Register[7..0] (Read-only)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

8

Field	Bits	Definition
00h	7–0	Low-order byte of Device ID register Value of this byte is 00h

Figure 30. CONFIG[2] device ID register (read-only)

03h CONFIG[3] Device ID Register[15..8] (Read-Only)

7	6	5	4	3	2	1	0
1	0	0	1	0	0	0	1

1

Field	Bits	Definition
91h	7–0	High-order byte of Device ID register Value of this byte is 91h

Figure 31.CONFIG[3] device ID register (read-only)

3.3. Configuration Registers, continued

3.3.6. COMMAND REGISTERS

CONFIG[5] command register[15..8] reads back 00h. The Power 9100 does not have an SERR– pin. Fast back-to-back transactions are only allowed to the same agent.

04h CONFIG[4] Command Register[7..0]

7	6	5	4	3	2	1	0
1	0	vps	0	0	0	me	ioe
1	1	1	1	1	1	1	1

Field	Bits	Definition
wait_cycle_control	7	Controls address data stepping 1 = supports data stepping (hardwired)
parity_error_response	6	Controls parity error checking 0 = no parity checking (hardwired)
vps VGA_palette_snoop	5	Controls palette snooping 0 = snoop disabled, Power 9100 responds to palette accesses normally (default). 1 = snoop enabled, Power 9100 does not claim palette write bus cycles, but performs the write to the palette Read cycles are not affected by the VGA palette snoop bit.
memory_write_and_in-validate_enable	4	Not implemented 0 (hardwired)
special_cycles	3	Special cycles are ignored 0 (hardwired)
bus_master	2	Slave device only 0 (hardwired)
me mem_enable	1	memory space enable. 0 = disabled, Power 9100 doesn't respond to any memory accesses. (default) 1 = enabled, Power 9100 responds to memory space accesses.
ioe io_enable	0	I/O space enable. 0 = disabled, Power 9100 does not respond to any I/O space accesses except for the configuration index and data registers for VL bus mode. (PCI bus default) 1 = enabled, Power 9100 responds to I/O space accesses. (VL bus default)

Figure 32. CONFIG[4] command register

3.3. Configuration Registers, continued

3.3.7. STATUS REGISTERS

CONFIG[6] status register[7..0] reads back 00h. The Power 9100 cannot accept fast back-to-back transactions from different agents.

07h CONFIG[7] Status Register [15..8](Read-Only)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0
1	1	1	1	1	2		1

Field	Bits	Definition
	7..6	Not implemented; always reads back 0.
DEVSEL timing	2–1	01b = medium DEVSEL timing
	0	Not implemented; always reads back 0

Figure 33. CONFIG[7] status register (read-only)

3.3. Configuration Registers, continued

3.3.8. REVISION ID REGISTER

The optional PCI revision ID register, CONFIG[8], is not implemented; it reads back 00h. The Power 9100 revision level is located in the sysconfig register. (See section 4.3.1.)

3.3.9. CLASS CODE REGISTERS

CONFIG[9] programming interface register reads back 00h. CONFIG[10] and CONFIG[11] are defined in figures 34 and 35, respectively.

0Ah CONFIG[10] Sub-class code Register(Read-Only)

7	6	5	4	3	2	1	0
vp/be	0	0	0	0	0	0	0
1	7						

Field	Bits	Definition
vp/be vga present (PCI bus mode)	7	PCI bus mode: 0 = VGA present in emulation mode (sub-class code = 00h, VGA compatible controller) 1 = VGA absent in emulation mode (sub-class code = 80h, other display controller) This bit is set by PU_CONFIG[25] on reset (see figure 27).
BIOS enable (VL bus mode)		VL bus mode: Silicon revision N4C–A2 ignores this bit; later revisions operate as follows: 0 = BIOS enabled on VL bus 1 = BIOS disabled on VL bus, supported on ISA bus
	6–0	Reads back 0

Figure 34. CONFIG[10] sub-class code register (read-only)

0Bh CONFIG[11] Base Class Code Register(Read-Only)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1
8							

Field	Bits	Definition
	7–0	Display controller (base class code = 03h, display controller)

Figure 35. CONFIG[11] base class code register (read-only)

3.3. Configuration Registers, continued

3.3.10. DISPLAY MEMORY BASE ADDRESS REGISTERS

CONFIG[16], CONFIG[17], and CONFIG[18] read back 00h and ignore writes.

13h CONFIG[19] Display Memory Base Address Register

7	6	5	4	3	2	1	0
wbase							
8							

Field	Bits	Definition
wbase	7–0	Base for linear display memory and native register access protocol. Address is relocatable every 16 MB. See figures 46 and 51.

Figure 36. CONFIG[19] display memory base address register

3.3.11. EXPANSION ROM BASE ADDRESS REGISTERS

30h CONFIG[48] Expansion ROM Base Address Register

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ren
7							1

Field	Bits	Definition
	7–1	Reads back 0; writes are ignored
ren rom_enable	0	0 = BIOS ROM decoding disabled (default for PCI) 1 = BIOS ROM decoding enabled Silicon version N4C–A2: default for VL bus mode. Silicon versions N4C–A4 and higher: in VL bus mode, this bit is set to the inverse of PU_CONFIG[25] (see figure 27).

Figure 37. CONFIG[48] expansion ROM base address register

3.3. Configuration Registers, continued

31h CONFIG[49] Expansion ROM Base Address Register

7	6	5	4	3	2	1	0
rb0	0	0	0	0	0	0	0
1	7						

Field	Bits	Definition
rb0 rom_base[0]	7	Bit 0 of the expansion ROM base address (default is 0 after reset)
	6–0	Reads back 0; writes are ignored

Figure 38. CONFIG[49] expansion ROM base address register

32h CONFIG[50] Expansion ROM Base Address Register

7	6	5	4	3	2	1	0
rom_base[8..1]							
8							

Field	Bits	Definition
rom_base[8..1]	7–0	Bits 8..1 of the expansion ROM base address (default is 0Ch after reset)

Figure 39. CONFIG[50] expansion ROM base address register

33h CONFIG[51] Expansion ROM Base Address Register

7	6	5	4	3	2	1	0
rom_base[16..9]							
8							

Field	Bits	Definition
rom_base[16..9]	7–0	Bits 16..9 of the expansion ROM base address (default is 00h after reset)

Figure 40. CONFIG[51] expansion ROM base address register

3.3. Configuration Registers, continued

3.3.12. PCI INTERRUPT REGISTER

3Ch CONFIG[60] PCI Interrupt Line Register

7	6	5	4	3	2	1	0
IRQ line							

8

Field	Bits	Contents
IRQ line	7–0	PCI compliant interrupt line register value

Figure 41. CONFIG[60] PCI interrupt line register

3.3.13. PCI INTERRUPT PIN REGISTER

3Dh CONFIG[61] PCI Interrupt Pin Register (Read-Only)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

8

Field	Bits	Contents
interrupt pin	7–0	01h; Power 9100 is connected to INTA– interrupt pin

Figure 42. CONFIG[61] PCI interrupt pin register

3.3. Configuration Registers, continued

3.3.14. POWER 9100 DEVICE SPECIFIC REGISTERS

40 h CONFIG[64] Register (Read-Only)

7	6	5	4	3	2	1	0
bus		cfgba			0	0	edi
2		3			2		1

Field	Bits	Contents
bus	7–6	Read-only copy of PU_CONFIG.BUS (see figure) 11 = Reserved 10 = VESA Local Bus 01 = PCI Bus 00 = Reserved (internal Testing mode)
cfgba	5–3	Read–only copy of PU_CONFIG.CFGBA (VL mode, see figure 27)

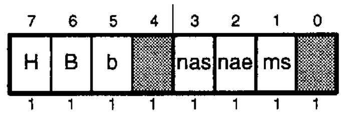
pu_config.cfgba	config register index	config register data
000	9100h	9104h
001	9108h	910Ch
010	9110h	9114h
011	9118h	911Ch
100	9120h	9124h
101	9128h	912Ch
110	9130h	9134h
111	9138h	913Ch

Field	Bits	Contents
	2–1	Reads back 0
edi eedain (eedata in)	0	Current state of CKSEL[2] pin.

Figure 43. CONFIG[64] register (read-only)

3.3. Configuration Registers, continued

41h CONFIG[65] Register

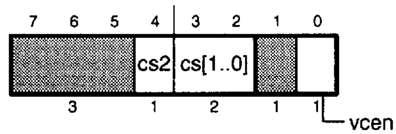


Field	Bits	Contents
H	7	Half-word swap for native register access protocol through 000A0000–000AFFFF
B	6	Byte swap for native register access protocol through 000A0000–000AFFFF
b	5	Bit swap for native register access protocol through 000A0000–000AFFFF
reserved	4	Reserved; must be set to 0
nas na_select	3	Native mode access select. Select native mode registers or frame buffer to respond to address range 000A0000–000AFFFF This bit is set to 0 (default) on reset. 0 = Frame Buffer 1 = Native-mode registers
nae na_enable	2	Native mode access enable. Respond to address range 000A0000–000AFFFF. This bit is set to 0 (default) on reset. 0 = disabled. 1 = enabled.
ms modeselect	1	Native/emulation mode select. This bit is set by PU_CONFIG[26] on reset (see figure 27). 0 = Power 9100 is in native mode. 1 = Power 9100 is in emulation (VGA) mode.
reserved	0	Reserved; must be set to 0

Figure 44. CONFIG[65] register

3.3. Configuration Registers, continued

42h CONFIG[66] Register



Field	Bits	Contents
reserved	7–5	Reserved;must be set to 0
cs2 cksel[2]	4	In native mode, sets the current state of open collector pin CKSEL[2]. (set to 0 on reset)(CKSEL[2] pin is also used as EEDATA input pin. For reading, use CONFIG[64][0]). In emulation mode, CKSEL[2] is controlled by the sequencer index register 12h. See section 12.5.12.
cs[1..0] cksel[1..0]	3–2	In native mode, sets the current state of output pins CKSEL[1..0]. (set to 0 on reset) In emulation mode, CKSEL[1..0] are controlled by the miscellaneous output register. See section 12.4.3.
reserved	1	Reserved; must be set to 0
vcen	0	The inverse of the level of the VCEN– pin. This bit is set to 0 (default) on reset. (Also used as EECLK)

Figure 45. CONFIG[66] register

3.4. Address Formats

The following sections define each of the general address formats. Formats for accessing specific registers, pseudo-registers, and commands are all variations on these general formats, and are defined in chapters 4 and 5.

3.4.1. LINEAR FRAME BUFFER ADDRESSING

The pixel address format, illustrated in figure 46, allows direct access to the pixel map. Half-word, byte, and bit swapping for endian control is specified in the system configuration (sysconfig) register (see section 4.3). Data in the frame buffer is stored in big-endian format.

When the host issues a pixel address, the Power 9100 accesses the display memory as if it were normal memory. Read accesses do not modify the pixel data; the current settings of the pixel processing registers do not affect the pixel data. Accessing non-existent pixels yields undefined results. Pixel accesses complete within a few clock cycles; the worst case includes the time required for a VRAM refresh, a shift register load, a row miss, and the actual read or write.

3.4.2. ALTERNATE FRAME BUFFER APERTURE MAPPING

The aperture address format, illustrated in figure 47, allows indirect access to the pixel map. Half-word, byte, and bit swapping for endian control is specified in the system configuration (sysconfig) register (see section 4.3). Data in the frame buffer is stored in big-endian format. Access to the frame buffer through this mechanism is controlled by CONFIG[65].na_enable and CONFIG[65].na_select (figure 44). By setting the alt_read_bank (section 4.3.4) and alt_write_bank (section 4.3.5) registers, this address format can be used to access a contiguous 64 KB block of frame buffer memory on any 64 KB boundary. The contents of the alt_read_bank and alt_write_bank registers are merged with the offset from the address field to compute the display memory address.

Linear Frame Buffer Pixel Address Format

31–24	23	22–0
CONFIG[19].wbase	1	pixel address
8	1	23

Field	Bits	Contents
CONFIG[19].wbase	31–24	See figure 36
	23	Specifies frame buffer access (0 specifies register location; see figure 51).
pixel address	22–0	Address of pixel in the display memory

Figure 46. Linear frame buffer pixel address format

Alternate Frame Buffer Aperture Address Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	pixel within bank
16																16

Field	Bits	Contents
aperture address	31–16	Maps to memory space 000A 0000h – 000A FFFFh
pixel within bank	15–0	Address of pixel within a 64 KB bank of display memory
Note: CONFIG[65].na_enable = 1 and CONFIG[65].na_select = 0 to access frame buffer		

Figure 47. Alternate frame buffer aperture address format

3.4. General Address Formats, continued

3.4.3. ROM BIOS ACCESS

The ROM BIOS access format, illustrated in figure 49, allows direct access to the ROM BIOS. Data in the ROM BIOS is stored in little-endian format and is not subject to H, B, and b swapping. Access to the ROM BIOS through this mechanism is controlled by the expansion ROM address registers (see section 3.3.11). Both the CONFIG[4].mem_enable field and the CONFIG[48].rom_enable fields must be set to one in order to enable BIOS ROM access.

3.4.4. NATIVE REGISTER ACCESS PROTOCOL

Except for configuration registers, frame buffer memory and ROM BIOS access to all remaining native mode items are controlled through the native register access protocol. See figure 48 for a list of the items that are accessible.

[Editor's Note: This address format was previously called Coprocessor Access protocol.]

There are two access methods: direct and indirect. In both access modes, data swapping is available. The indirect access format is only available if CONFIG[65].NA_ENABLE is set to one.

Figure 50 shows the indirect native register access protocols. With this protocol the H, B, and b swapping are specified by CONFIG[65].H, CONFIG[65].B, and CONFIG[65].b.

Figure 51 shows the direct native register access protocol. With this protocol the H, B and b swapping is directly specified within the address itself.

Address Format	Described in Section	Page
RAMDAC access	3.4.5	41
Video coprocessor access	3.4.6	41
System control registers	4.3	47
Video control registers	4.6	72
VRAM control registers	4.7	77
Status register	4.4.2	57
Parameter engine registers	4.4	54
Drawing engine registers	4.5	66
Load coordinate pseudo-registers	5.1	82
Quad command	5.2	83
Blit command	5.3	84
Pixel8 command	5.4	85
Pixel1 command	5.5	87
Next_pixels command	5.6	88

Figure 48. Address formats for native mode registers and commands

ROM BIOS Access Format

31–15	14–0
ROM base address	ROM address
17	15

Field	Bits	Contents
ROM base address	31–15	Location of 32 KB ROM See figures 38 (CONFIG[49]), 39 (CONFIG[50]), 40 (CONFIG[51])
ROM address	14–0	Address within 32 KB ROM

Note: CONFIG[4].mem_enable = 1 and CONFIG[48].rom_enable = 1 to enable BIOS ROM access

Figure 49. ROM BIOS access format

3.4. General Address Formats, continued

Indirect Native Register Access Protocol

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	native address														
17																	15														
Prefix																															

Field	Bits	Definition
Prefix	31–15	Indirect access prefix used in native register address formats.
native address	14–0	See figures 52, 53, 54, 55, 58, 65, 67, 70, 79, 85, 90, 93, 94, 95, 96, 98, and 100
Note: CONFIG[65].na_enable = 1 and CONFIG[65].na_select = 1; see figure 44		

Figure 50. Indirect native access protocol

Direct Native Register Access Protocol

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONFIG[19].wbase								0	0	0	0	0	H	B	b	0	native address														
8								5					1	1	1	1	15														
Prefix 17																															

Field	Bits	Definition
Prefix	31–15	Direct access prefix used in native register address formats.
CONFIG[19].wbase	31–24	Base address for direct native register access (see figure 36).
	23–19	Must be 0
H	18	Half-word swap
B	17	Byte within half-word swap
b	16	Bit within byte swap
	15	Must be 0
native address	14–0	See figures 52, 53, 54, 55, 58, 65, 67, 70, 79, 85, 90, 93, 94, 95, 96, 98, and 100

Figure 51. Direct native access protocol

3.4. General Address Formats, continued

3.4.5. RAMDAC ADDRESS FORMAT

The RAMDAC address format, illustrated in figure 52, is an extension to the native register set and is only accessible in native mode. In emulation mode, the RAMDAC is accessible through standard VGA protocol.

RAMDAC reads and writes are initiated immediately and completed within a few clock cycles; the longest possible wait time for a RAMDAC access is the time required to complete a VRAM refresh, plus the time required to complete a shift register reload, plus the time required for the access. The 8-bit data transfer to or from the RAMDAC occurs directly across the MD[24..16] bits to the host bus, subject to the native mode register byte swapping in access mode. (See chapter 8).

3.4.6. VIDEO COPROCESSOR INTERFACE

The video coprocessor address format, illustrated in figure 53, is an extension of the native mode register set and allows direct access to the video coprocessor. Video coprocessor reads and writes are initiated immediately and completed within a few clock cycles. The video coprocessor interface is only enabled when operating in native mode. The video coprocessor interface control pins are shared with the VGA video bus. Video Power responds to 64 32-bit register locations. Unused registers read back all zeroes. This interface can be used for other purposes if Video Power is not present.

RAMDAC Address Format

31 … 15	14	13	12	11	10	9	8	7	6	5 … 2	1	0
prefix	0	0	0	0	0	1	0	0	0	ramdac addr	0	0
15	9									4	2	

Field	Bits	Definition
prefix	31–15	Prefix. See figures 50 and 51.
decoding address	14–6	See figures 54 and 55
ramdac addr	5–2	Addresses RAMDAC registers; each RAMDAC is product specific Bit 5 = RS[3] Bit 4 = RS[2] Bit 3 = RS[1] Bit 2 = RS[0]
32-bit access	1–0	Must be 00

Figure 52. RAMDAC address format

Video Coprocessor Interface Address Format

31 … 15	14	13	12	11	10	9 … 2	1	0
prefix	0	0	0	0	1	reg	0	0
17	5					8	2	

Field	Bits	Definition
prefix	31–15	Prefix. See figures 50 and 51.
decoding address	14–10	See figures 54 and 55
reg	9–2	The register of the video coprocessor
32-bit access	1–0	Must be 00

Figure 53. Video coprocessor interface address format

3.4. General Address Formats, continued

Prefix*		Address Bits													Specific Address	Section
31	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
		0	0	0	0	0	0	0	0	–	–	–	–	–	System control registers	4.3
		0	0	0	0	0	0	1	0	–	–	–	–	–	Video control registers	4.6
		0	0	0	0	0	0	1	1	–	–	–	–	–	VRAM control registers	4.7
		0	0	0	0	0	1	0	0	0	–	–	–	–	RAMDAC control	3.4.5
		0	0	0	0	1	–	–	–	–	–	–	–	–	Video Coprocessor Interface	3.4.6
		0	1	0	0	0	0	0	0	0	0	0	1	1	Pixel8 command (Power 9000 format)	5.4
		1	–	–	–	–	–	–	–	–	–	–	–	–	Pixel8 command (Power 9100 format)	5.4
		0	1	0	0	0	0	0	0	0	0	1	0	1	Next_pixels command	5.6
		0	1	0	0	0	0	0	1	–	–	–	–	–	Pixel1 command	5.5
		0	1	0	0	0	0	1	1	–	–	–	–	–	Parameter engine control registers	4.4.3
		0	1	0	0	0	1	–	–	–	–	–	–	–	Drawing engine pixel processing registers	4.5
		0	1	1	0	0	0	0	–	–	–	–	–	–	Device coordinate registers	4.4.1
		0	1	1	0	0	1	–	–	–	–	–	–	–	Load coordinates pseudo-registers	5.1
* See figures 50 and 51.																

Figure 54. Decoding of addresses for write operations

Prefix*		Address Bits													Specific Address	Section
31	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
		0	0	0	0	0	0	0	0	–	–	–	–	–	System control registers	4.3
		0	0	0	0	0	0	1	0	–	–	–	–	–	Video control registers	4.6
		0	0	0	0	0	0	1	1	–	–	–	–	–	VRAM control registers	4.7
		0	0	0	0	0	1	0	0	0	–	–	–	–	RAMDAC control	3.4.5
		0	0	0	0	1	–	–	–	–	–	–	–	–	Video Coprocessor Interface	3.4.6
		0	1	0	0	0	0	0	0	0	0	0	0	0	Status register	4.4.2
		0	1	0	0	0	0	0	0	0	0	0	0	1	Blit command	5.3
		0	1	0	0	0	0	0	0	0	0	0	1	0	Quad command	5.2
		0	1	0	0	0	0	1	1	–	–	–	–	–	Parameter engine control registers	4.4.3
		0	1	0	0	0	1	–	–	–	–	–	–	–	Drawing engine pixel processing registers	4.5
		0	1	1	0	0	0	0	–	–	–	–	–	–	Device coordinate registers	4.4.1
		0	1	1	0	0	1	–	–	–	–	–	–	–	Load coordinates pseudo-registers	5.1
* See figures 50 and 51.																

Figure 55. Decoding of addresses for read operations

Chapter 4. Native Mode Registers

4.1. Overview

The 32-bit Power 9100 registers specify parameters for an operation, control the operation, and maintain status information. Figure 56 summarizes the Power 9100 registers. All registers except those designated "read only" are read/write registers.

The register descriptions in this section define the register functions, present the address format for accessing each register, and provide a complete definition of each register field.

4.2. Native Register Summary

Subgroup	Register	Name/Function	Accessed via	Effect of reset	See section
System Control Registers					
	sysconfig	System configuration. Specifies system configuration information.	System control register access address format; see figure 58	set to zero (Host Reset)	4.3.1
	interrupt	Interrupt. Records interrupt conditions.		set to zero	4.3.2
	interrupt_en	Interrupt enable. Enables interrupts upon occurrence of interrupt conditions.		set to zero	4.3.3
	alt_read_bank	Alternate bus read bank select		set to zero	4.3.4
	alt_write_bank	Alternate bus write bank select		set to zero	4.3.5
Parameter Engine Registers					
Device coordinate	X[0],Y[0] X[1],Y[1] X[2],Y[2]] X[3],Y[3]	Device coordinate. Supplies screen coordinates for drawing operation.	Device coordinate register access address format; see figure 65	Not changed	4.4.1
Status	status	Status. Read only. Records status of drawing engine and coordinate register clip checks.	Status register access address format; see figure 67	Not changed	4.4.2
Control and condition	oor	Out of range. Read only. Records out of range x,y values.	Parameter engine control and condition register access address format; see figure 70	Not changed	4.4.3
	cindex	Index. Supplies current index into x and y coordinates		Not changed	
	w_off.x/y	Window offset. Supplies offset of current window on the display.		Not changed	
	p_w_min p_w_max	Parameter engine window minimum, parameter engine window maximum. Read only. Record the contents of the drawing engine window minimum (w_min) and window maximum (w_max) registers.		Not changed	
	xclip yclip	Xclip, yclip. Read only. Record results of clip checks on x and y coordinates.		Not changed	
	xedge_lt xedge_gt yedge_lt yedge_gt	Vertex relationship. Read only. Record results of vertex relationship checking.		Not changed	

Figure 56. Native register summary (1 of 3)

4.2. Native Register Summary, continued

Subgroup	Register	Name/Function	Accessed via	Effect of reset	See section
Drawing Engine Registers					
Pixel processing	color[0] color[1] color[2] color[3]	Color register 0, foreground Color register 1, background Color register 2 Color register 3	Drawing engine pixel processing register access address format; see figure 79	Not changed	4.5.1
	pmask	Plane mask. Specifies plane mask.		Not changed	4.5.2
	draw_mode	Draw mode. Specifies control for writing within a picked window and selects the destination buffer for drawing operations.		Not changed	4.5.3
	pat_originx pat_originy	X pattern origin, y pattern origin. Specify x and y screen coordinates for pattern origin.		Not changed	4.5.4
	raster	Raster. Specifies minterms, transparency and other parameters for a raster operation.		Not changed	4.5.5
	pixel8_reg	Pixel8. Stores excess pixel8 operation data bits.		Not changed	4.5.6
	p_w_min b_w_min	Window minimum. Specifies minimum x,y values for window.		Not changed	4.5.7
	p_w_max b_w_min	Window maximum. Specifies maximum x,y values for window.		Not changed	4.5.7
	pattern[1] pattern[2] pattern[3] pattern[4]	Pattern. Specify pattern.		Not changed	4.5.8
	user[0] user[1] user[2] user[3]	User defined registers		When modes switch, the value is preserved; value not preserved on reset.	4.5.9

Figure 56, continued. Native register summary (2 of 3)

Subgroup	Register	Name/Function	Accessed via	Effect of reset	See section
Video Control Registers					
Horizontal	hrzc	Horizontal counter. Read only.	Video control register access address format; see figure 85	Set to zero	4.6
	hrzt	Horizontal length.		Set to FFFh	
	hrzsr	Horizontal sync rising edge.		Set to FFFh	
	hrzbr	Horizontal blank rising edge.		Set to FFFh	
	hrzbf	Horizontal blank falling edge.		Set to FFFh	
	prehrzc	Horizontal counter preload.		Set to zero	
Vertical	vrtc	Vertical counter. Read only.		Set to zero	
	vrtt	Vertical length.		Set to FFFh	
	vrtsr	Vertical sync rising edge.		Set to FFFh	
	vrtbr	Vertical blank rising edge.		Set to FFFh	
	vrtbf	Vertical blank falling edge.		Set to FFFh	
	prevrtc	Vertical counter preload.		Set to zero	
Repaint	srtctl	Screen repaint timing control.		Set to zero	
	srtctl2	Screen repaint timing control.		Set to FFFh	
	qsfcounter	QSF counter.		Set to zero	
VRAM Control Registers					
	mem_config	Memory configuration.	VRAM control register ac-cess,address format; see fig-ure 90	Set to zero	4.7.1
	rfperiod	Refresh period.		Set to 0x3FF	4.7.2
	rfcount	Refresh counter.		Set to zero	4.7.3
	rlmax	RAS low maximum.		Set to 0x3FF	4.7.4
	rlcur	RAS low current.		Set to zero	4.7.5
	pu_config	Power-up configuration		Preserves 32-bit value on the frame buffer data bus at rising edge of reset	3.3.3

Figure 56, continued. Native register summary (3 of 3)

4.3. System Control Registers

The system control registers, summarized in figure 57, control all but the video portions of the Power 9100 (which are covered in section 4.6). They can be read or written at any time. Figure 58 defines the specific format for accessing these registers.

Register	Reference	Function
sysconfig	4.3.1	System configuration. Specifies shift control, swapping for endian control on pixel accesses, buffer selection for pixel accesses, and the current version of the Power 9100. This register is only reset by a host reset.
interrupt	4.3.2	Interrupt. Indicates the occurrence of various Power 9100 interrupt conditions. This register is reset by a mode change (native mode).
interrupt_en	4.3.3	Interrupt enable. Enables/disables the assertion of the interrupt signal upon the occurrence of specific interrupt conditions. This register is reset by a mode change (native mode).
alt_read_bank	4.3.4	Alternate bus read bank select. Read/write. Controls banking of the frame buffer. This register is set to 0 by a mode change (native mode).
alt_write_bank	4.3.5	Alternate bus write bank select. Read/write. Controls banking of the frame buffer. This register is set to 0 by a mode change (native mode).

Figure 57. System control registers

Address Format for System Control Register Access

Bits	31–15	14	13	12	11	10	9	8	7	6–2	1	0
Field	prefix	0	0	0	0	0	0	0	0	register	0	0
Width	17	8								5	2	

Field	Bits	Contents		
prefix	31–15	See figures 50 and 51.		
decoding address	14–7	Decoding address for read/write operations; see figures 54 and 55.		
register	6–2	Entry	Offset	Register Selected
		00001	0004h	sysconfig
		00010	0008h	interrupt
		00011	000Ch	interrupt_en
		00100	0010h	alt_write_bank
		00101	0014h	alt_read_bank
		00110–11111		reserved
32-bit access	1–0	Must use 00.		

Figure 58. Address format for system control register access

4.3. System Control Registers, continued

4.3.1. SYSTEM CONFIGURATION REGISTER

The system configuration (sysconfig) register provides various system configuration controls, as defined in figure 60. Bits 13 through 11 determine half-word, byte, and bit swapping control.

The shift control fields are used by the drawing engine to determine the size, in bytes, of the horizontal display. The pixel_size field and the shift control fields are set independently from one another. To set the shift control fields:

1. Determine the number of pixels in a horizontal scan line.
2. Multiply the number of pixels by the number of bytes (as set in the pixel_size field, sysconfig[28..26]).
3. Set the four shift control fields so that the total adds up to the total number of bytes in the scan line.

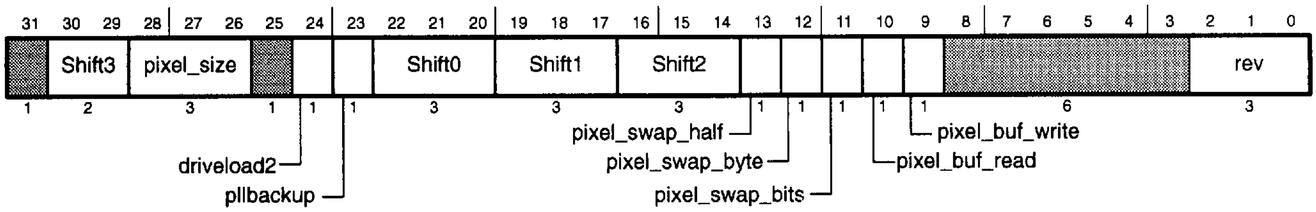
There are many combinations of fields that work for a given display. Refer to figure 59 for examples.

Screen Resolution	Horizontal Bytes	Shift0	Shift1	Shift2	Shift3
640x480x8 bits	640	101	011	000	00
		011	000	101	00
1280x1024x24 bits	3840	110	101	100	10
		111	100	101	01

Figure 59. Examples of shift control field settings (sysconfig register)

4.3. System Control Registers, continued

System Configuration Register (sysconfig)



Field	Bits	Contents
reserved	31	Reserved; must be set to 0
Shift3	30–29	Shift Control 3. 00 = SHIFT_0 (no add) 01 = SHIFT_1024 (add 1024) 10 = SHIFT_2048 (add 2048) 11 = SHIFT_4096 (add 4096)
pixel_size	28–26	Pixel size for drawing engine 010 = 8 bits per pixel (8bpp) 011 = 16 bits per pixel (16bpp) 111 = 24 bits per pixel (24bpp) 101 = 32 bits per pixel (32bpp)
reserved	25	Reserved; must be set to 0
driveload2	24	Frame buffer controller drive load selection (for most applications, set to 0) 0 = Normal drive load 1 = Double drive load
pllbackup	23	Overrides internal PLL to use external clocking circuit (for most applications, set to 0) 0 = Normal operation 1 = External PLL clocking.
Shift0	22–20	Shift Control 0. 000 = SHIFT_0 (no add) 011 = SHIFT_128 (add 128) 100 = SHIFT_256 (add 256) 101 = SHIFT_512 (add 512) 110 = SHIFT_1024 (add 1024) 111 = SHIFT_2048 (add 2048)
Shift1	19–17	Shift Control 1. 000 = SHIFT_0 (no add) 010 = SHIFT_64 (add 64) 011 = SHIFT_128 (add 128) 100 = SHIFT_256 (add 256) 101 = SHIFT_512 (add 512) 110 = SHIFT_1024 (add 1024)
Shift2	16–14	Shift Control 2. 000 = SHIFT_0 (no add) 001 = SHIFT_32 (add 32) 010 = SHIFT_64 (add 64) 011 = SHIFT_128 (add 128) 100 = SHIFT_256 (add 256) 101 = SHIFT_512 (add 512)

Figure 60. System configuration (sysconfig) register (1 of 2)

4.3. System Control Registers, continued

Field	Bits	Contents
pixel_swap_half	13	Half-word swap selection for endian control for frame buffer accesses: 0 = do not swap half words on frame buffer accesses. 1 = swap half words on frame buffer accesses
pixel_swap_byte	12	Byte swap selection for endian control for frame buffer accesses: 0 = do not swap bytes on frame buffer accesses 1 = swap bytes on frame buffer accesses
pixel_swap_bits	11	Bit swap selection for endian control for frame buffer accesses: 0 = do not swap bits on frame buffer accesses 1 = swap bits on frame buffer accesses
pixel_buf_read	10	~~B~~Frame buffer selection for host read pixel accesses 0 = buffer 0 (set to 0 for single buffer display) 1 = buffer 1
pixel_buf_write	9	~~B~~Frame buffer selection for host write pixel accesses 0 = buffer 0 (set to 0 for single buffer display) 1 = buffer 1
reserved	8–3~~0~~	Reserved; must be set to 0
rev	2–0	Read only field. (Immediately after a reset, this field reads back 000; after the first write to the sysconfig register after the reset, this field reads back the correct data.) Device revision 010 = N4C–A2 silicon 100 = N4E–A4 silicon 101 = X1F–A5 silicon

Figure 60, continued System configuration (sysconfig) register (2 of 2)

4.3. System Control Registers, continued

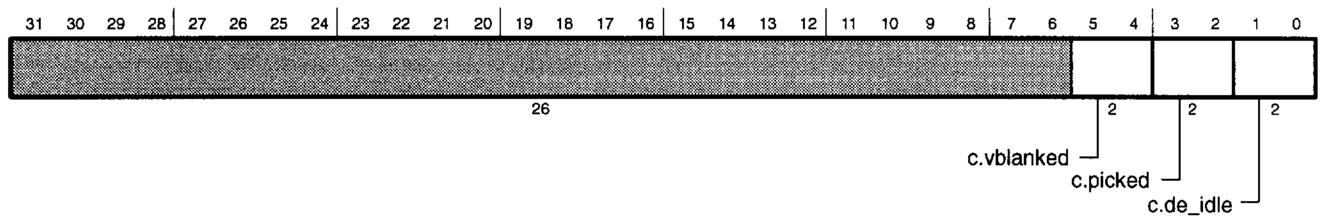
4.3.2. INTERRUPT REGISTER

The interrupt register is a read/write register that accumulates status information. The Power 9100 sets bits in the interrupt register to indicate the occurrence of specific interrupt conditions, as shown in figure 61. The Power 9100 generates a hardware interrupt (that is, asserts the INTR signal) only when the corresponding bit is set in the interrupt enable (interrupt_en) register (see section 4.3.3).

The interrupt register bits are "sticky;" they remain set after the conditions they reflect are gone. They must be cleared by a host register transfer. Each condition bit is preceded by a field write control bit which must be set to 1 to write into the field or 0 to leave the field as it stands when writing to the register to clear interrupt bits.

If the host sets an interrupt bit in the interrupt register and enables that interrupt via the interrupt enable register, the Power 9100 generates the interrupt.

Interrupt Register



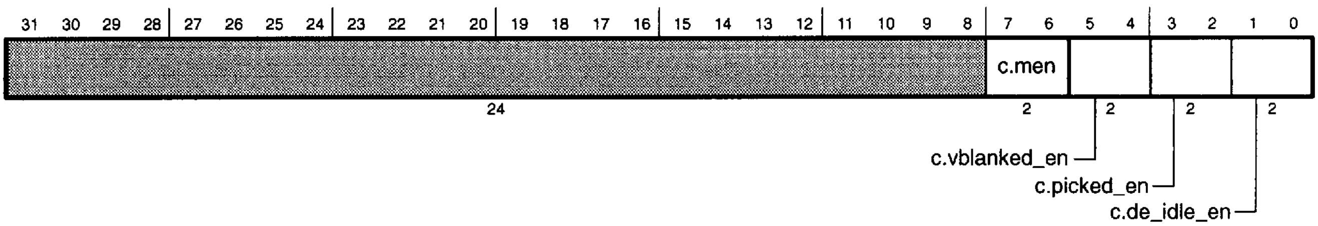
Field	Bits	Contents
reserved	31–6	Reserved; must be set to 0
c.vblanked	5–4	Vblank: Bit 5: field write control bit Bit 4: 0 = no vblank occurred 1 = vblank occurred
c.picked	3–2	Pick: Bit 3: field write control bit Bit 2: 0 = no pick occurred 1 = pick occurred
c.de_idle	1–0	Drawing engine idle: Bit 1: field write control bit Bit 0: 0 = drawing engine busy 1 = drawing engine idle

Figure 61. Interrupt register

4.3.3. INTERRUPT ENABLE REGISTER

The interrupt enable (interrupt_en) register specifies the conditions under which the Power 9100 asserts the INTR interrupt signal. Figure 62 defines this register. Note that each interrupt enable bit is preceded by a field write control bit which must be set to 1 to write into the field (or 0 to leave the field as it stands) when writing to the register. Also, if the host sets an interrupt bit in the interrupt register and enables that interrupt by setting the corresponding bit in the interrupt enable register, the Power 9100 generates the interrupt.

Interrupt Enable Register (interrupt_en)



Field	Bits	Contents
reserved	31–8	Reserved; must be set to 0
c.men	7–6	Master enable: Bit 7: field write control bit Bit 6: 0 = disable all interrupts 1 = enable interrupts as specified by individual interrupt enables (bits 5–0)
c.vblanked_en	5–4	Vblank: Bit 5: field write control bit Bit 4: 0 = do not generate interrupt when a vblank occurs 1 = generate interrupt when a vblank occurs
c.picked_en	3–2	Pick: Bit 3: field write control bit Bit 2: 0 = do not generate interrupt when a pick occurs 1 = generate interrupt when a pick occurs
c.de_idle_en	1–0	Drawing engine status: Bit 1: field write control bit Bit 0: 0 = do not generate interrupt when drawing engine goes idle 1 = generate interrupt when drawing engine goes idle

Figure 62. Interrupt enable (interrupt_en) register

4.3. System Control Registers, continued

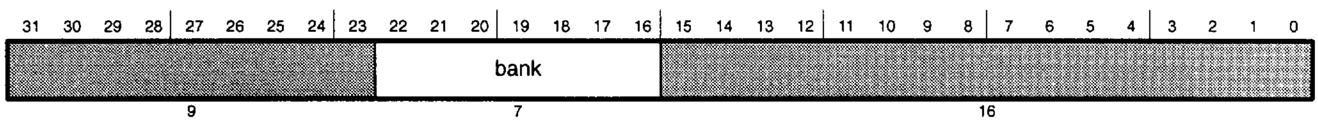
4.3.4. ALTERNATE READ BANK REGISTER

The alternate read bank register (alt_read_bank) specifies the seven high order address bits when the host uses the alternate aperture frame buffer banking logic to read from the frame buffer. The lower 16 bits are described in figure 47. See figure 63 for details.

4.3.5. ALTERNATE WRITE BANK REGISTER

The alternate write bank register (alt_write_bank) specifies the seven high order address bits when the host uses the alternate aperture frame buffer banking logic to write from the frame buffer. The lower 16 bits are described in figure 47. See figure 63 for details.

alt_write_bank and alt_read_bank Registers



Field	Bits	Contents
reserved	31–23	Reserved; must be set to 0
bank	22–16	Bank for alternate frame buffer aperture
reserved	15–0	Reserved; must be set to 0

Figure 63. alt_write_bank and alt_read_bank registers

4.4. Parameter Engine Registers

The parameter engine registers, summarized in figure 64, supply the coordinates for a drawing operation, provide control for the drawing operation, and maintain status information. The host can read or write these registers at any time. A system reset leaves them in an undefined state.

Register Category	Register	Reference	Function
Device coordinate	X[0],Y[0] X[1],Y[1] X[2],Y[2]] X[3],Y[3]	Section 4.4.1	Device coordinate. Supplies screen coordinates for drawing operation.
Status	status	Section 4.4.2	Status. Read only. Records status of drawing engine and coordinate register clip checks.
Control and condition	oor	Section 4.4.3	Out of range. Read only. Records out of range x,y values.
	cindex		Index. Supplies current index into x and y coordinates
	w_off.x/y		Window offset. Supplies offset of current window on the display.
	p_w_min p_w_max		Parameter engine window minimum, parameter engine window maximum. Read only. Record the contents of the drawing engine window minimum (w_min) and window maximum (w_max) registers.
	xclip yclip		Xclip, yclip. Read only. Record results of clip checks on x and y coordinates.
	xedge_lt xedge_gt yedge_lt yedge_gt		Vertex relationship. Read only. Record results of vertex relationship checking.

Figure 64. Parameter engine registers

4.4. Parameter Engine Registers, continued

4.4.1. DEVICE COORDINATE REGISTERS

The device coordinate registers supply the four screen coordinates required for each drawing operation. There are four X registers and four Y registers, one for each x and y value required. X and y values can be supplied as 32-bit or 16-bit values, as indicated by the YX field in the address format. When supplied as 16-bit values, the x and y values are packed and transferred in a single 32-bit read or write.

Figure 65 defines the specific user register address format for accessing these registers.

The eight device coordinate registers are 13 bits wide. They can be loaded individually or in xy pairs. When loaded individually, the first 13 bits of a 32-bit data value are used; the others are ignored. Similarly, there are unused bits when xy pairs are loaded.

4.4. Parameter Engine Registers, continued

Address Format for Device Coordinate Register Access

Bits	31–15	14	13	12	11	10	9	8	7–6	5	4–3	2	1	0
Field	prefix	0	1	1	0	0	0	0	reg	rel	yx		0	0
Width	17	7							2	1	2	1	2	

Field	Bits	Contents
prefix	31–15	See figures 50 and 51.
decoding address	14–8	Decoding address for read/write operations; see figures 54 and 55.
reg device coordinate register	7–6	00 X[0]/Y[0] 01 X[1]/Y[1] 10 X[2]/Y[2] 11 X[3]/Y[3]
rel	5	0 Perform absolute screen addressing 1 Perform window-relative screen addressing (write only)
yx	4–3	00 Not used 01 Read or write 32-bit X value 10 Read or write 32-bit Y value 11 Read or write 16-bit X value (high 16 bits) and 16-bit Y value (low 16 bits)
reserved	2	Reserved; must be set to 0
32-bit access	1–0	Must use 00.

Figure 65. Address format for device coordinate register access

Device Coordinate Registers

Bits	31–29	28–16	15–13	12–0
Field		coordinate b		coordinate a
Width	3	13	3	13

Field	Bits	Contents
reserved	31–29	Reserved; must be set to 0
coordinate b	28–16	yx = 11: *x* screen coordinate of xy pair (see figure 65 for yx setting) yx = 10: reserved, must be set to 0 yx = 01: reserved, must be set to 0 yx = 00: reserved, must be set to 0
reserved	15–13	Reserved; must be set to 0
coordinate a	12–0	yx = 11: *y* screen coordinate of xy pair (see figure 65 for yx setting) yx = 10: *y* screen coordinate yx = 01: *x* screen coordinate yx = 00: reserved, must be set to 0

Figure 66. Device coordinate registers

4.4. Parameter Engine Registers, continued

4.4.2. STATUS REGISTER

The status register is a read-only register whose contents reflect the current status of the drawing engine and coordinate register clip checks. Access to the status register can be obtained by issuing a quad or blit command. Figure 67 defines the specific address format for accessing the status register. Figure 68 defines the status register. The Power 9100 updates this register every time the x and y coordinates are changed.

Bit 31 must be zero to successfully initiate another quad draw operation or a blit operation; bit 30 must be zero when issuing the first in a series of pixel1 or pixel8 operations or writing to any drawing engine register (see section 5.7.5).

Address Format for Status Register Access

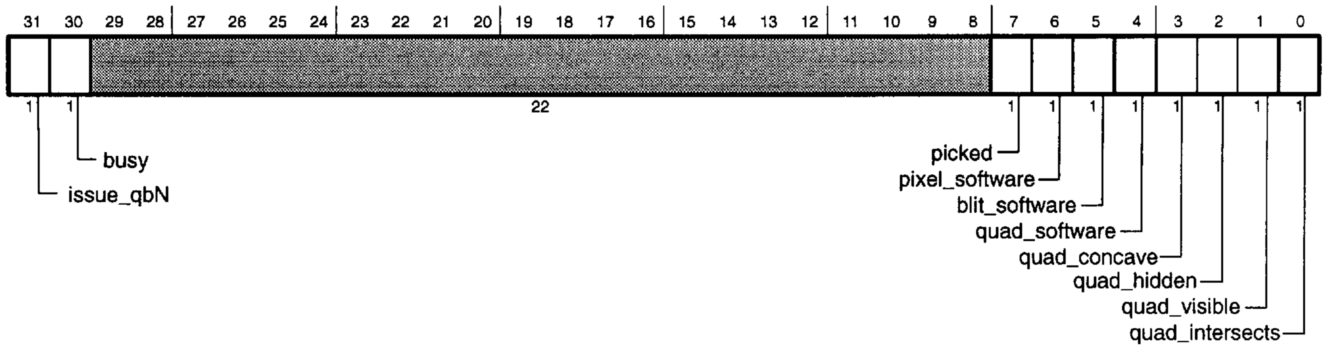
31–15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
prefix	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	13													2	

Field	Bits	Contents
prefix	31–15	See figures 50 and 51.
decoding address	14–2	Decoding address for read operation; see figure 55.
32-bit access	1–0	Must use 00.

Figure 67. Address format for status register access

4.4. Parameter Engine Registers, continued

Status Register



Field	Bits	Contents
issue_qbN	31	Quad/blit initiate another: 0 = may start another quad operation if previous was also a quad operation or may start another blit operation if previous was also a blit operation 1 = may not start quad or blit operation
busy	30	0 = drawing engine not busy 1 = drawing engine busy
reserved	29–8	Reserved; must be set to 0
picked	7	Pick detected. (This field is a copy of the picked field in the interrupt register.)
pixel_software	6	Exception encountered for pixel command; the operation cannot be performed and must be done in software.
blit_software	5	Exception encountered for blit command; the operation cannot be performed and must be done in software.
quad_software	4	Exception encountered for quad command; the operation cannot be performed and must be done in software.
quad_concave	3	The requested quad is concave.
quad_hidden	2	The source coordinates for a quad draw are entirely outside the clipping window.
quad_visible	1	The source coordinates for a quad draw are entirely inside the clipping window.
quad_intersects	0	The source coordinates for a quad draw straddle the clipping window.

Figure 68. Status register

4.4. Parameter Engine Registers, continued

4.4.3. CONTROL AND CONDITION REGISTERS

The parameter engine control and condition registers control and monitor Power 9100 operations, as summarized in figure 69. Figure 70 defines the specific user register address format for accessing these registers.

Out of Range Register. The read-only out of range register (oor), illustrated in figure 71, identifies *x* and *y* values that are out of range of the drawing engine for the current drawing operation. The Power 9100 updates this register every time the *x* and *y* coordinate registers are modified.

Index Register. The index register (cindex) supplies the current index into the *x* and *y* coordinates as a 2-bit binary integer. The value is used for load coordinate computation (see section 5.1). This is a read/write register.

Window Offset Register. The window offset register (w_off.xy) supplies the offset of the current window on the display as two packed 16-bit two's complement binary integers, representing the *x* and *y* coordinates. This information is required by coordinate computation operations to calculate addressing relative to the window. It has no effect on clipping. This is a read/write register.

Pixel Window Minimum and Maximum Registers. The pixel valued window minimum (p_w_min) and window maximum (p_w_max) registers are read through this address format. See section 4.5.7 for a complete description of the window maximum and minimum registers.

Clip Registers. The read-only clip registers (xclip and yclip) define the results of clip checks, as summarized and illustrated in figures 73 and 74. The Power 9100 updates these registers for every new set of *x* and *y* values loaded into the coordinate registers.

Vertex Relationship Checking Registers. The read-only vertex relationship checking registers (xedge_lt, xedge_gt, yedge_lt, and yedge_gt) define the results of checks on vertices to verify that the requested drawing operation is acceptable, as summarized and illustrated in figures 75 through 78. The Power 9100 updates these registers for every new set of *x* and *y* values loaded into the coordinate registers.

Register	Reference	Description
oor	Figure 71	Out of range. Records the x,y values that are out of range for the current drawing operation.
cindex	Section 5.1	Index. Supplies the current index into the x,y coordinates for load coordinate computation.
w_off.xy	Figure 72	Window offset. Supplies the offset of the current window on the display for coordinate computation of window-relative addressing. (This provides a translation only, no clip or scale; it is an offset only.)
p_w_min p_w_max	Section 4.5.7	Parameter engine window minimum, parameter engine window maximum. Read only. Record the contents of the drawing engine window minimum (w_min) and window maximum (w_max) registers.
xclip yclip	Figures 73, 74	Clip check. Records the results of clip checks.
xedge_lt xedge_gt yedge_lt yedge_gt	Figures 75–78	Vertex relationship checking registers. These read-only registers define the results of checks on vertices in order to verify the acceptability of the requested drawing operation.

Figure 69. Control and condition registers

4.4. Parameter Engine Registers, continued

Address Format for Control and Condition Register Access

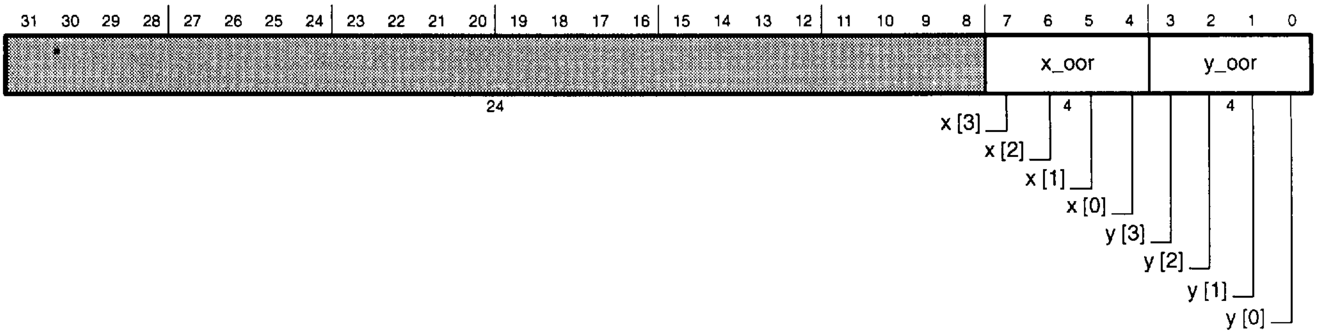
31–15	14	13	12	11	10	9	8	7	6–2	1	0
prefix	0	1	0	0	0	0	1	1	register	0	0
17	8								5	2	

Field	Bits	Contents		
prefix	31–15	See figures 50 and 51.		
decoding address	14–7	Decoding address for read/write operations; see figures 54 and 55.		
register	6–2	Entry	Offset	Register Selected
		00000	2180h	Reserved
		00001	2184h	oor
		00010	2188h	not used
		00011	218Ch	cindex
		00100	2190h	w_off.xy
		00101	2194h	p_w_min (Read only) (to write, refer to figure 79)
		00110	2198h	p_w_max (Read only) (to write, refer to figure 79)
		00111	219Ch	not used
		01000	21A0h	yclip
		01001	21A4h	xclip
		01010	21A8h	xedge_lt
		01011	21ACh	xedge_gt
		01100	21B0h	yedge_lt
		01101	21B4h	yedge_gt
		01110–11111		Reserved
32-bit access	1–0	Must use 00		

Figure 70. Address format for control and condition register access

4.4. Parameter Engine Registers, continued

Out of Range (oor) Register

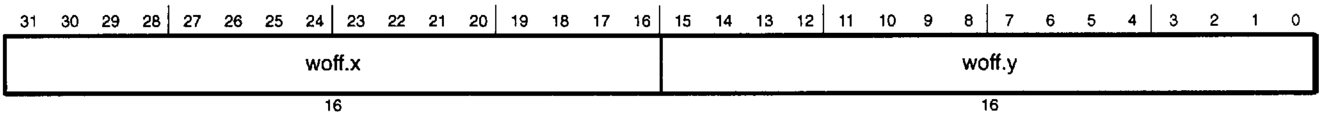


Field	Bits	Contents
reserved	31–8	Reserved; must be set to 0
x_oor	7–4	Each bit indicates whether or not one x value is out of range of the drawing engine (msb = x[3]...lsb = x[0]): 0 = The x value is not out of range 1 = The x value is out of range
y_oor	3–0	Each bit indicates whether or not one y value is out of range of the drawing engine (msb = y[3]...lsb = y[0]): 0 = The y value is not out of range 1 = The y value is out of range

Figure 71. Out of range (oor) register

4.4. Parameter Engine Registers, continued

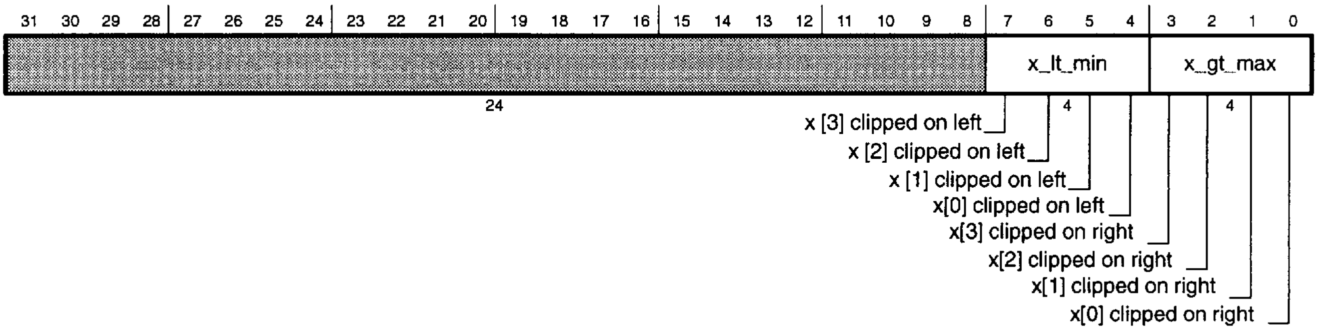
w_off.xy Register



Field	Bits	Contents
woff.x	31–16	X offset of current window (two's compliment binary integer)
woff.y	15–0	Y offset of current window (two's compliment binary integer)

Figure 72. Window offset (w_off.xy) register

Xclip Register

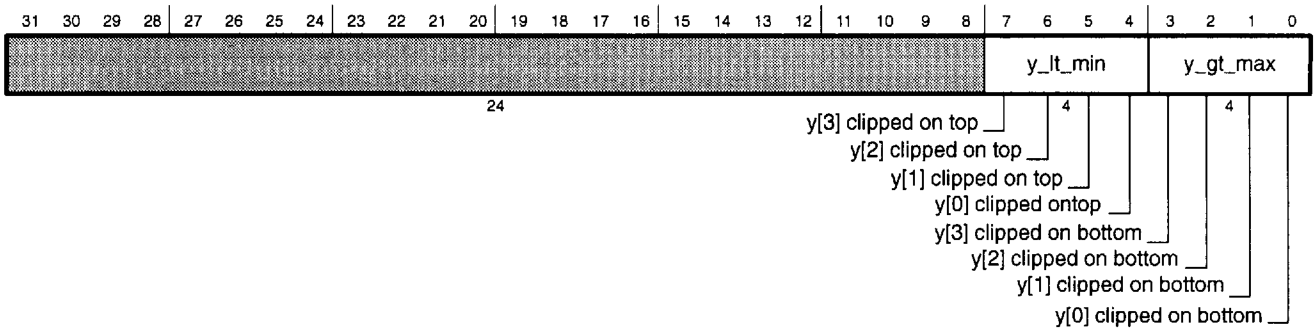


Field	Bits	Contents
reserved	31–8	Reserved; must be set to 0
x_lt_min	7–4	Each bit indicates whether the x value of one of the vertices falls outside and to the left of the clipping rectangle. x[n] < w_min.x 0 = The x value is not less than w_min.x 1 = The x value is less than w_min.x
x_gt_max	3–0	Each bit indicates whether the x value of one of the vertices falls outside and to the right of the clipping rectangle. x[n] > w_min.x 0 = The x value is not greater than w_max.x 1 = The x value is greater than w_max.x

Figure 73. Xclip register

4.4. Parameter Engine Registers, continued

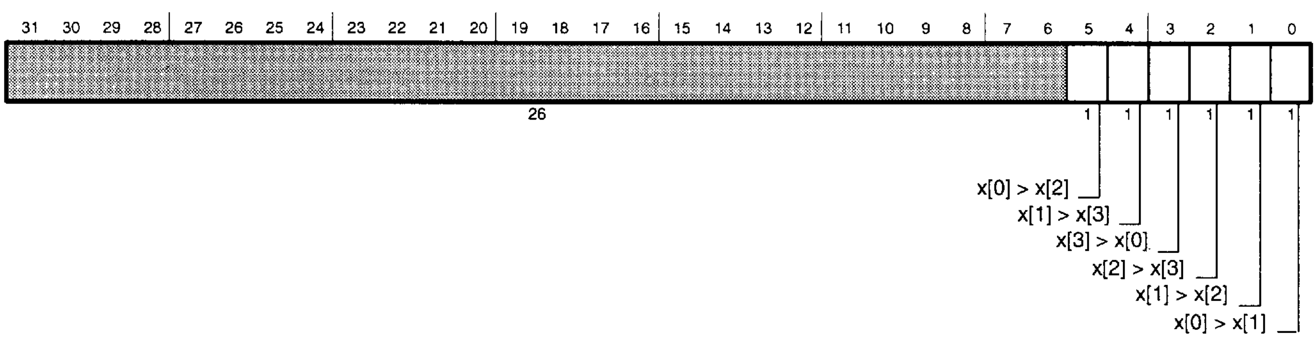
Yclip Register



Field	Bits	Contents
reserved	31–8	Reserved; must be set to 0
y_lt_min	7–4	Each bit indicates whether the y value of one of the vertices falls outside and to the top of the clipping rectangle. y[n] < w_min.y 0 = The y value is not less than w_min.y 1 = The y value is less than w_min.y
y_gt_max	3–0	Each bit indicates whether the y value of one of the vertices falls outside and to the top of the clipping rectangle. y[n] > w_min.y 0 = The y value is not greater than w_max.y 1 = The y value is greater than w_max.y

Figure 74. Yclip register

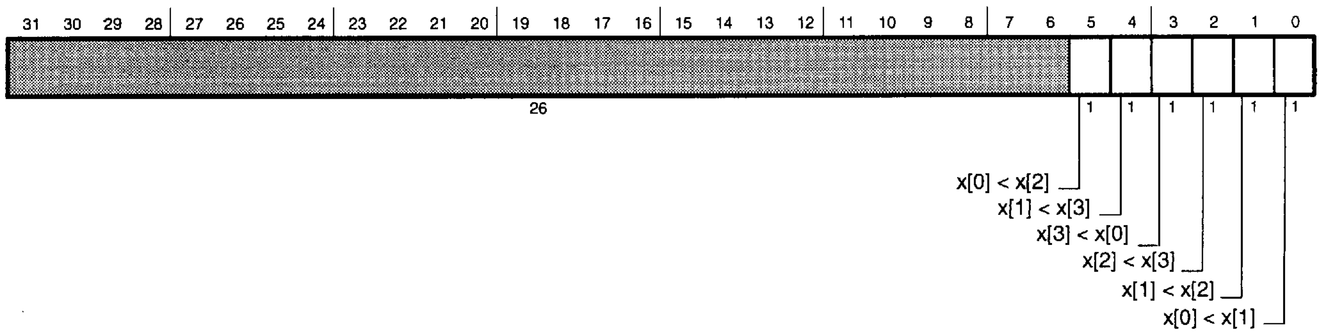
Xedge_lt Register



Field	Bits	Contents
reserved	31–26	Reserved; must be set to 0
	5–0	When the condition specified in the bit definition is met, the bit is set to one; when the condition is not met, the bit is set to zero.

Figure 75. Xedge_lt register

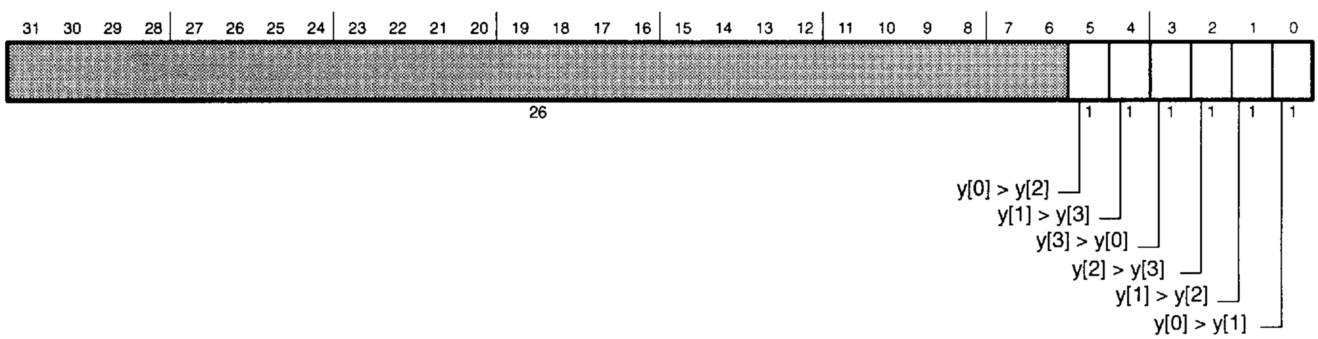
Xedge_gt Register



Field	Bits	Contents
reserved	31–6	Reserved; must be set to 0
xedge_gt	5–0	When the condition specified in the bit definition is met, the bit is set to one; when the condition is not met, the bit is set to zero.

Figure 76. Xedge_gt register

Yedge_lt Register

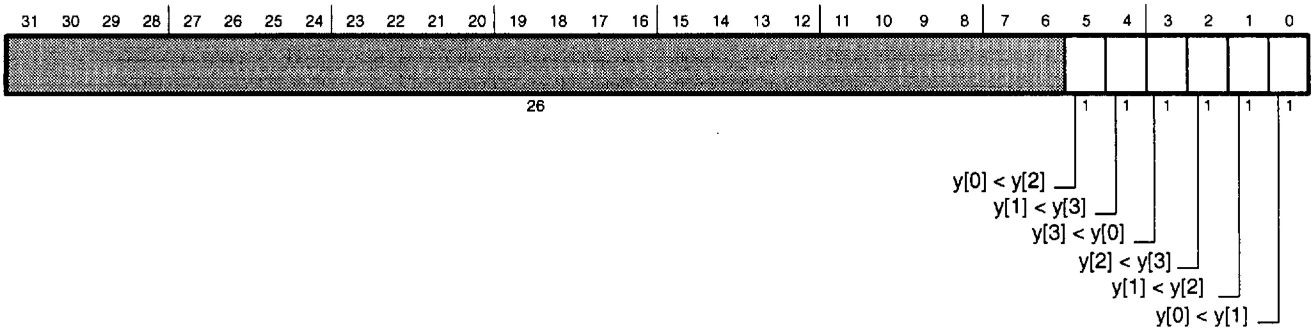


Field	Bits	Contents
reserved	31–6	Reserved; must be set to 0
yedge_lt	5–0	When the condition specified in the bit definition is met, the bit is set to one; when the condition is not met, the bit is set to zero.

Figure 77. Yedge_lt register

4.4. Parameter Engine Registers, continued

Yedge_gt Register



Field	Bits	Contents
reserved	31–6	Reserved; must be set to 0
yedge_gt	5–0	When the condition specified in the bit definition is met, the bit is set to one; when the condition is not met, the bit is set to zero.

Figure 78. Yedge_gt register

4.5. Drawing Engine Registers

The drawing engine registers are pixel processing registers that provide storage and define the foreground and background colors, plane mask, pattern and pattern origins, pixel drawing window limits, and raster operation parameters. These registers are accessed as illustrated in figure 79. The host can read or write these registers only when the drawing engine is idle (when the busy bit in the status register is set to zero); attempting to do so when the drawing engine is busy produces undefined results.

4.5. Drawing Engine Registers, continued

Address Format for Drawing Engine Pixel Processing Register Access

31–15	14	13	12	11	10	9	8–2	1	0
prefix	0	1	0	0	0	1	register	0	0
17	6						7	2	

Field	Bits	Contents		
prefix	31–15	See figures 50 and 51.		
decoding address	14–9	Decoding address for read/write operations; see figures 54 and 55.		
register	8–2	Entry	Offset	Register Selected
		0000000	2200h	color[0]
		0000001	2204h	color[1]
		0000010	2208h	Plane mask (pmask)
		0000011	220Ch	Draw mode (draw_mode)
		0000100	2210h	Pattern origin x (pat_originx)
		0000101	2214h	Pattern origin y (pat_originy)
		0000110	2218h	Raster (raster)
		0000111	221Ch	Pixel8 (pixel8_reg)
		0001000	2220h	Pixel Window min (p_w_min) (Write only) (to read, refer to figure 70)
		0001001	2224h	Pixel Window max (p_w_max) (Write only) (to read, refer to figure 70)
		0001010–0001101		Reserved
		0001110	2238h	color[2]
		0001111	223Ch	color[3]
		0010000–0011111		Reserved
		0100000	2280h	Pattern[0]
		0100001	2284h	Pattern[1]
		0100010	2288h	Pattern[2]
		0100011	228Ch	Pattern[3]
		0100100	2290h	User[0] (user defined register)
		0100101	2294h	User[1] (user defined register)
		0100110	2298h	User[2] (user defined register)
		0100111	229Ch	User[3] (user defined register)
		0101000	22A0h	Byte Window min (b_w_min)
		0101001	22A4h	Byte Window max (b_w_max)
		0101010–1111111		Reserved
32-bit access	1–0	Must be 00.		

Figure 79. Address format for drawing engine pixel processing register access

4.5. Drawing Engine Registers, continued

4.5.1. COLOR REGISTERS

There are four 32-bit color registers: color[0], color[1], color[2], and color[3]. These registers contain the colors used by the pattern and pixel1 logic. When operating in 8-bpp and 16-bpp the requested color must be replicated four and two times, respectively, to fill the entire register. In 24-bit mode, byte[0] is repeated in byte[3].

Color[0] is the foreground register; color[1] is the background register. Both are used in two-color patterns and in the pixel1 command. Four-color patterns use all four color registers (color[0..3]).

4.5.2. PLANE MASK REGISTER

The plane mask register (pmask) is a 32-bit read-write register. When operating in 8-bpp and 16-bpp, the requested mask must be replicated four and two times, respectively, to fill the entire register. This register is used by the drawing engine to disable and enable individual bit writes and utilizes the write-per-bit feature of the VRAM. A zero disables the write, masking the bit. A one enables the write. This register works in 8-, 16-, and 32-bit modes only.

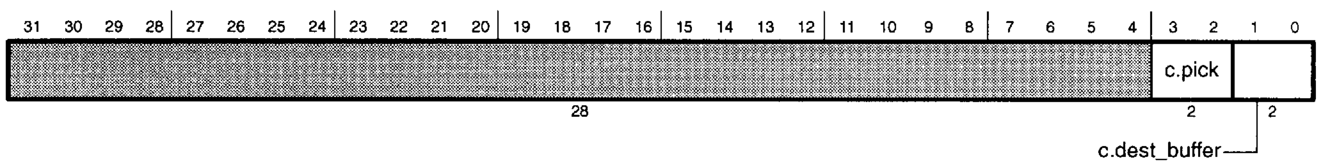
4.5.3. DRAW_MODE REGISTER

The draw_mode register controls writing within a picked window and selects the destination buffer for drawing operations, as defined in figure 80.

4.5.4. PATTERN ORIGIN REGISTERS

The two pattern origin registers (pat_originx and pat_originy) supply the 3-bit origin of the pattern to be used by the quad command to fill polygons on the display screen. The x and y origins are supplied in the lower three bits of each register.

Draw_mode Register



Field	Bits	Contents
reserved	31–4	Reserved; must be set to 0
c.pick	3–2	Pick mode enable. Bit 3: field write control bit Bit 2: 0 = Suppress any write operation outside the window; allow a write inside the window 1 = Suppress all write operations; set the picked bit in the interrupt register (see figure 61) if the suppressed write is within the window
c.dest_buffer	1–0	Destination buffer. Bit 1: field write control bit Bit 0: 0 = Select buffer 0 as the destination buffer for all drawing engine operations 1 = Select buffer 1 as the destination buffer for all drawing engine operations

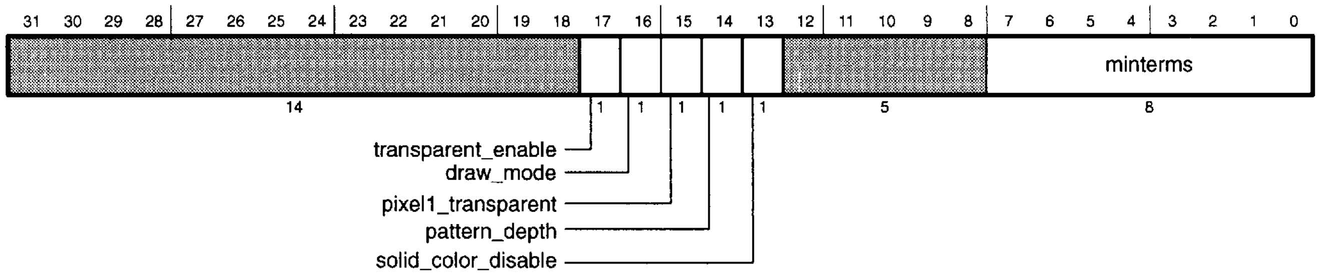
Figure 80. Draw_mode register

4.5. Drawing Engine Registers, continued

4.5.5. RASTER REGISTER

The raster register specifies the parameters for a raster operation, as defined in figure 81.

Raster Register



Field	Bits	Contents
reserved	31–18	Reserved; must be set to 0
transparent_enable	17	Transparent Pattern Enable 0 = Disable transparent pattern 1 = Enable transparent pattern
draw_mode	16	Quad draw mode (see section 5.7.2): 0 = X11 mode 1 = Oversized mode
pixel1_transparent	15	Transparent Pixel1 Enable 0 = Disable pixel1 transparent mode 1 = Enable pixel1 transparent mode
pattern_depth	14	Pattern Depth 0 = 2 color pattern 1 = 4 color pattern (8 bpp only)
solid_color_disable	13	Solid color disable 0 = Solid color enabled, Pattern disabled 1 = Solid color disabled, Pattern enabled
reserved	12–8	Reserved; must be set to 0
minterms[7..0]	7–0	Minterms for Boolean raster operation Bit 7 = Pattern • Source • Destination Bit 6 = Pattern • Source • $\overline{\text{Destination}}$ Bit 5 = Pattern • $\overline{\text{Source}}$ • Destination Bit 4 = Pattern • $\overline{\text{Source}}$ • $\overline{\text{Destination}}$ Bit 3 = $\overline{\text{Pattern}}$ • Source • Destination Bit 2 = $\overline{\text{Pattern}}$ • Source • $\overline{\text{Destination}}$ Bit 1 = $\overline{\text{Pattern}}$ • $\overline{\text{Source}}$ • Destination Bit 0 = $\overline{\text{Pattern}}$ • $\overline{\text{Source}}$ • $\overline{\text{Destination}}$ Taking each bit in the pixel individually, the minterms operate according to how the bits are set in this field. Set one or more bits to enable one or more cases to generate a 1 in the resulting pixel color. For example:

Operation	minterms[7..0]	Comments
Pixel drawing	CCh	Set if source is a 1
Pattern drawing	F0h	Set if pattern is a 1
Source XOR destination	66h	Set if source, destination differ
Source AND destination	88h	Set if source, destination are set

Figure 81. Raster register

4.5. Drawing Engine Registers, continued

4.5.6. PIXEL8 REGISTER

The pixel8 register (pixel8_reg) provides temporary storage for excess bit data from a pixel8 operation (see section 5.4). The data is stored in the lower end of the register. This register is generally of interest to the programmer only when a context switch is being performed in the middle of a pixel8 operation.

4.5.7. WINDOW MINIMUM/WINDOW MAXIMUM REGISTERS

The window clipping registers in the drawing engine are used to control the clipping of drawing operations. A single rectangle is supported by specifying the minimum and maximum X and Y values. Clipping is always enabled and imposes no performance penalty. There are two sets of clipping registers: pixel valued (p_w_min and p_w_max) and byte valued (b_w_min and b_w_max). The pixel valued registers contain the X and Y values encoded in pixels and scan lines. The byte valued registers contain the X and Y values in bytes and scan lines. The pixel valued registers are loaded through the drawing engine address format but are read back through the parameter engine address format (see section 4.5). Figure 82 illustrates the contents of these registers. Both registers must be reloaded (see section 4.4).

4.5.8. PATTERN REGISTERS

The pattern registers are read/write registers by which the host specifies the pattern for quad fill. The pattern registers are a remappable 8x8 bit map that is two bits deep (indexes 0 and 1). The origin is located using the pattern origin XY register. The two-bit pattern from the pattern register selects one of four color registers. The index 1 registers pattern[1][0..7][0..7] are only used with the special four-color pattern in 8-bpp mode. Figure 83 shows the format of the pattern registers.

4.5.9. USER REGISTERS

These four 32-bit read/write registers are available in native mode for use by the system software. The values in these registers are preserved when switching between native mode and VGA mode.

The WEITEK BIOS uses the user[0] register. Contact WEITEK before writing code that uses the user[2..0] registers.

Window Minimum (p_w_min and b_w_min) and Window Maximum (p_w_max and b_w_max) Registers

31–29	28–16	15–13	12–0
(reserved)	x value	(reserved)	y value
3	13	3	13

Field	Bits	Contents
reserved	31–29	Reserved; must be set to 0
x value	28–16	Minimum x value in w_min register; maximum x value in w_max register.
reserved	15–13	Reserved; must be set to 0
y value	12–0	Minimum y value in w_min register; maximum y value in w_max register.

Figure 82. Window minimum (p_w_min and b_w_min) and window maximum (p_w_max and b_w_max) registers

4.5. Drawing Engine Registers, continued

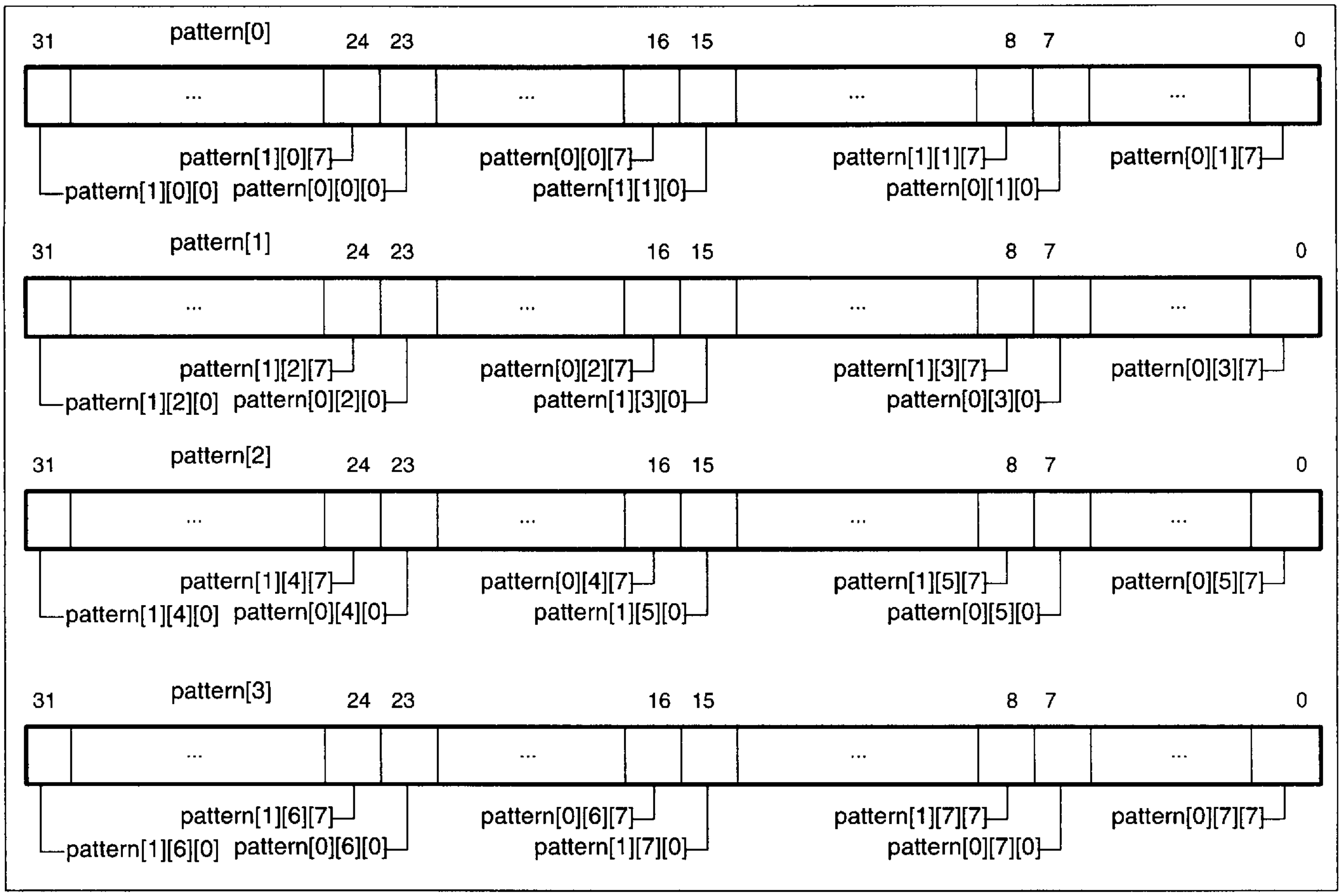


Figure 83. Pattern registers

4.6. Video Control Registers

Figures 86 and 87 summarize the video control registers. They can be accessed at any time. Figure 85 defines the specific address format for accessing these registers.

Most of the registers in the video section operate in terms of CRTC clocks (CRTC_CLKs). The CRTC clock is generated by the programmable divide down of the video clock input to the Power 9100. Usually, the RAMDAC divides the clock. (The divide ratio of the dot clock can also be controlled by mem_config.crtc_freq and mem_config.video_clk_sel.)

QSF counter position. This value determines which bit in the qsfcounter register is used to generate internal shift register reload requests. This field should be set to four less than the log (base 2) of the length of one-half of the shift register of the VRAM, as counted by CRTC_CLK. A value of zero in this field indicates a shift register of half-length 2^{0+4} or 16 CRTC_CLKs, with a total shift register length of 32 CRTC_CLKs.

See also section 8.1.

Memory Configuration	qsfselect value CRTC_CLK= 1/4 dot clock	qsfselect value CRTC_CLK= 1/8 dot clock in 8bpp mode
1	4	3
2	4	3
3	5	4
4	5	4
5	6	5

Figure 84. QSF counter position examples

4.6. Video Control Registers, continued

Address Format for Video Control Register Access

31–15	14	13	12	11	10	9	8	7	6–2	1	0
prefix	0	0	0	0	0	0	1	0	register	0	0
17	8								5	2	

Field	Bits	Contents		
prefix	31–15	See figures 50 and 51.		
decoding address	14–7	Decoding address for read/write operations; see figures 54 and 55.		
register	6–2	Entry	Offset	Register Selected
		00000		Reserved
		00001	0104h	hrzc
		00010	0108h	hrzt
		00011	010Ch	hrzsr
		00100	0110h	hrzbr
		00101	0114h	hrzbf
		00110	0118h	prehrzc
		00111	011Ch	vrtc
		01000	0120h	vrtt
		01001	0124h	vrtsr
		01010	0128h	vrtbr
		01011	012Ch	vrtbf
		01100	0130h	prevrtc
		01101	0134h	sraddr
		01110	0138h	srtctl
		01111	013Ch	sraddr_inc
		10000	0140h	srtctl2
		10001–11111		Reserved
32-bit access	1–0	Must be 00.		

Figure 85. Address format for video control register access

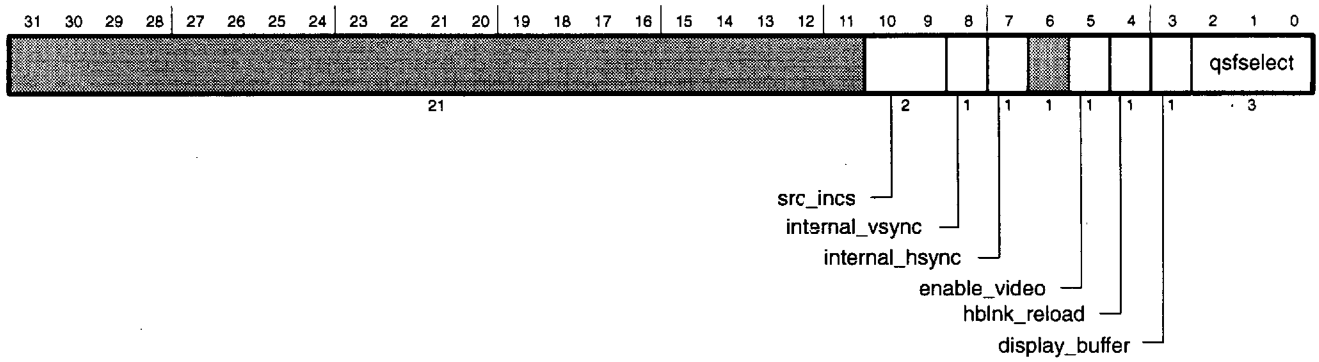
4.6. Video Control Registers, continued

Register	Description/Function
HORIZONTAL TIMING	
hrzc	Horizontal counter. Read only. Specifies the current pixel position along a horizontal sweep; the Power 9100 increments this counter, upon each occurrence of CRTC_CLK. It resets to zero at the assertion of HSYNC. The value occupies the lower 12 bits of the register, which is set by the Power 9100.
hrzt	Horizontal length. Read/write. Specifies the length of a horizontal scan line. The value occupies the lower 12 bits of the register, which is set by the host. The Power 9100 compares the current hrzc value (the current pixel position) to this value to determine when to wrap around. The hrzt register must be a multiple of 8 for all modes except 8-bit-per-pixel with a 64-bit wide RAMDAC, when it must be a multiple of 16. For simplicity, hrzt may be a multiple of 16 for all modes.
hrzsr	Horizontal sync assertion. Read/write. Specifies the the amount of time in CRC_CLKS, the horizontal sync is active. The value occupies the lower 12 bits of the register, which is set by the host.
hrzbr	Horizontal blank deassertion. Read/write. Specifies the location along a horizontal sweep which defines the deassertion of the horizontal blank. The value occupies the lower 12 bits of the register, which is set by the host.
hrzbf	Horizontal blank assertion. Read/write. Specifies the location along a horizontal sweep which defines the assertion of the horizontal blank. The value occupies the lower 12 bits of the register, which is set by the host.
prehrzc	Horizontal counter preload value. Read/write. Specifies the value with which to preload hrzc upon receipt of an internal or external HSYNC or an external VSYNC; allows synchronization of the Power 9100 with external video sources, whatever the combination of internal or external delays. The value occupies the lower 12 bits of the register, which is set by the host. Set this register to zero when using only internal syncs.
VERTICAL TIMING	
vrtc	Vertical counter. Read only. Specifies the current line position along a vertical sweep; the Power 9100 increments this counter upon each occurrence of HSYNC. The value occupies the lower 12 bits of the register, which is set by the Power 9100. In the vertical timing sequence, vrtt defines the number of horizontal lines from the assertion of VSYNC to the deassertion of the next VSYNC (the number of horizontal lines in a complete vertical scan cycle). It resets to zero at the assertion of VSYNC.
vrtt	Vertical length. Read/write. Specifies the number of lines in a vertical sweep. The value occupies the lower 12 bits of the register, which is set by the host. The vrtc register counts from zero to vrtt–1 and then repeats.
vrtsr	Vertical sync assertion. Read/write. Specifies the amount of time, in horizontal scan lines, the vertical sync is active. The value occupies the lower 12 bits of the register, which is set by the host.
vrtbr	Vertical blank deassertion. Read/write. Specifies the location along a vertical sweep which defines the vertical blank deassertion. The value occupies the lower 12 bits of the register, which is set by the host.
vrtbf	Vertical blank assertion. Read/write. Specifies the location along a vertical sweep which defines the vertical blank assertion. The value occupies the lower 12 bits of the register, which is set by the host.
prevrtc	Vertical counter preload value. Read/write. Specifies the value with which to preload vrtc upon receipt of an internal or external VSYNC; allows synchronization of the Power 9100 with external video sources, whatever the combination of internal or external delays. The value occupies the lower 12 bits of the register, which is set by the host. Set this register to zero when using only internal syncs.
SCREEN REPAINT	
srtctl2	Screen refresh timing control. Read/write. Specifies controls for screen refresh, as set by the host. Figure 88 defines this register.
srtctl	Screen refresh timing control. Read/write. Specifies controls for screen refresh, as set by the host. Figure 87 defines this register.
qsfcounter	QSF counter. Read only. Used to determine when to generate a shift register load operation. It is a duplicate of the QSF signal from the VRAMs. It keeps track of which part of the SAM is being shifted out. It is loaded with a zero after every read transfer and incremented by CRTC_CLK.

Figure 86. Video control registers

4.6. Video Control Registers, continued

Screen Repaint Timing control (srtctl) Register

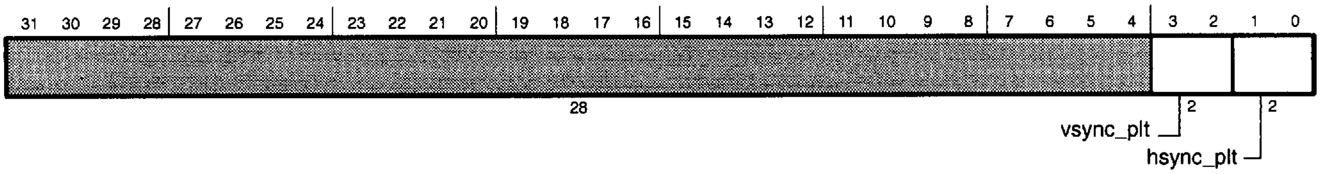


Field	Bits	Contents
reserved	31–11	Reserved; must be set to 0
src_incs	10–9	Amount to increment SRADDR register (see section 8.1.4) 00 = 256 01 = 512 10 = 1024
internal_vsync	8	Internal vertical synchronization: 0 = Select external VSYNC 1 = Select internal VSYNC
internal_hsync	7	Internal horizontal synchronization: 0 = Select external HSYNC 1 = Select internal HSYNC
reserved	6	Reserved; must be set to 0
enable_video	5	Enable video: 0 = Blanks asserted, HSYNC and VSYNC are disabled 1 = Normal operation
hblnk_reload	4	Screen repaint mode (see section 8.1.8): 0 = Normal (split shift mode) 1 = Restricted (HBLNK– reload mode)
display_buffer	3	Buffer for display. Any change to this field takes effect for the next screen refresh. 0 = Buffer 0 1 = Buffer 1
qsfselect	2–0	QSF counter position. See section 4.6

Figure 87. Screen repaint timing control (srtctl) register

4.6. Video Control Registers, continued

Screen Repaint Timing Control (srtctl2) Register



Field	Bits	Contents
reserved	31–4	Reserved; must be set to 0
vsync_plt	3–2	External VSYNC polarity control. Only applies when VSYNC is used as an output (not during external sync). 00 = High true polarity 01 = Low true polarity 10 = Forced low 11 = Forced high
hsync_plt	1–0	External HSYNC polarity control.. Only applies when HSYNC is used as an output (not during external sync). 00 = High true polarity 01 = Low true polarity 10 = Forced low 11 = Forced high

Figure 88. Screen repaint timing control (srtctl2) register

4.7. VRAM Control Registers

The VRAM control registers, which specify the timings and configurations associated with the VRAMs, are summarized in figure 89. Figure 90 defines the specific address format for accessing these registers; aside from the standard general format, bits 8 through 7 always contain the value 3, and bits 6 through 2 select the VRAM control register.

4.7.1. MEMORY CONFIGURATION REGISTER

The memory configuration register (mem_config) defines how the frame buffer is configured, as defined in figure 91. See chapter 7 for definitions of the memory configurations.

Setting mem_config.vram_read_sample = 1 and mem_config.slow_host_hifc = 0 is not supported.

4.7.2. REFRESH PERIOD REGISTER

The refresh period (rfperiod) register defines the length of the refresh period, which is the number of cycles between memory refreshes. The register field containing the refresh period is the least significant 10 bits of the register; the rest of the register must be zeroed. A system reset initializes the contents of this register to 0x3FF.

The values written to the rfperiod register are copied to the rfcount register when the rfcount register contents reach zero. (See section 4.7.3.)

4.7.3. REFRESH COUNT REGISTER

The refresh count (rfcount) register keeps track of the number of cycles between refreshes. The Power 9100 loads this register with the value contained in the rfperiod register, and then decrements the rfcount register contents on each MEMCLK. When the rfcount register contents reach zero, the Power 9100 reloads the register with the current value in the rfperiod register and makes a refresh request. The register field containing the refresh count is the least significant 10 bits of the register; the rest of the register contains zeroes.

4.7.4. RAS LOW REGISTER

The RAS low maximum (rlmax) register specifies the maximum amount of time that the RAS– signal can be asserted. The register field containing the time is the least significant 10 bits of the register; the rest of the register contains zeroes. A system reset initializes the contents of this register to 0x3FF.

The values written to the rlmax register are copied to the rlcur register when the rlcur register contents reach zero. (See section 4.7.5.)

4.7.5. LOW CURRENT REGISTER

The RAS low current (rlcur) register counts down the amount of time that the RAS– signal can be held low when it is asserted. When the Power 9100 asserts the RAS– signal, it also loads the rlcur register with the value in the rlmax register. The Power 9100 then decrements the contents of the rlcur register on each MEMCLK either until RAS– is deasserted or until the rlcur value reaches zero. If the register contents reach zero while RAS– remains asserted, the Power 9100 performs a refresh sequence to precharge the VRAMS. The register field containing the time count is the least significant 10 bits of the register; the rest of the register contains zeroes.

4.7.6. POWER-UP CONFIGURATION REGISTER

The PU_CONFIG register is a read-only copy of the value that is on the 32-bits of the frame buffer data bus when the system reset is deasserted. As well as setting certain default modes within the Power 9100, it can also be used to control certain BIOS settings.

Register	Reference	Function
mem_config	4.7.1	Memory configuration. Read/write. Specifies how the frame buffer is configured.
rfperiod	4.7.2	Refresh period. Read/write. Controls the frequency of memory refresh sequences.
rfcount	4.7.3	Refresh down counter. Read only. Refresh occurs when rfcount reaches zero.
rlmax	4.7.4	RAS low maximum. Read/write. Specifies the maximum amount of time the Power 9100 can assert the RAS– signal.
rlcur	4.7.5	RAS low current. Read only. Controls the maximum amount of time the RAS– signal is asserted.
pu_config	4.7.6 3.3.3	Power-up configuration. Read only. The initial settings of the VRAM data bus are preserved in this register.

Figure 89. VRAM control registers

4.7. VRAM Control Registers, continued

Address Format for VRAM Control Register Access

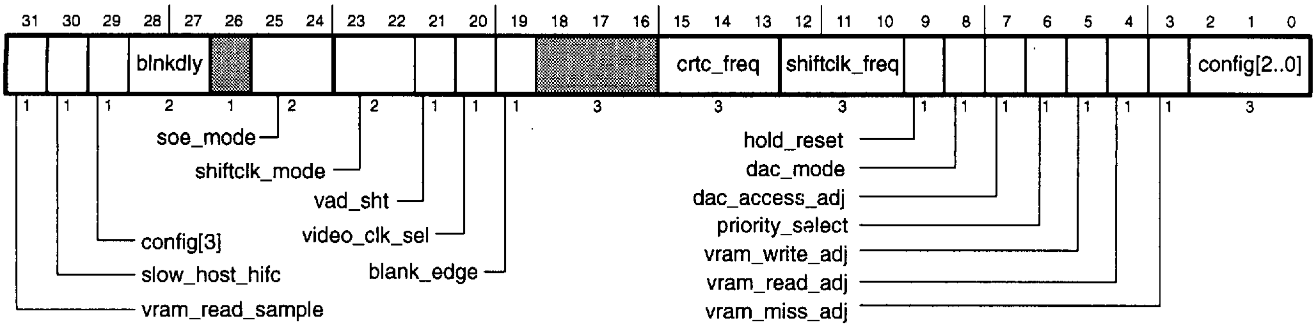
31–15	14	13	12	11	10	9	8	7	6–2	1	0
prefix	0	0	0	0	0	0	1	1	register	0	0
17	8								5	2	

Field	Bits	Contents		
prefix	31–15	See figures 50 and 51.		
decoding address	14–7	Decoding address for read/write operations; see figures 54 and 55.		
register	6–2	Entry	Offset	Register Selected
		00001	0184h	mem_config
		00010	0188h	rfperiod
		00011	018Ch	rfcount
		00100	0190h	rfmax
		00101	0194h	rlcur
		00110	0198h	pu_config
		00111–11111		Reserved
32-bit access	1–0	Must be 00		

Figure 90. Address format for VRAM control register access

4.7. VRAM Control Registers, continued

Memory Configuration (mem_config) Register



Field	Bits	Contents
vram_read_sample	31	Adjustment for read timing template, see Figures 126 and 127.
slow_host_hifc	30	Slow host interface adjust. 1 = add 1 cycle to frame buffer reads from host. Must be set to 1 if bit 31, vram_read_sample, is set to 1.
config[3]	29	High order bit of memory configuration. See Figures 116 through 123
blnkdly	28–27	Controls delay for blank generation in units of shiftclk_freq: 00 = 0 shiftclk_freq delay 01 = 1 shiftclk_freq delay 10 = 2 shiftclk_freq delay 11 = 3 shiftclk_freq delay
reserved	26	Reserved; must be set to 0
soe_mode	25–24	Sets timing patterns for serial output enables: 00 = 1 SAM bank style 01 = 2 SAM bank style 10 = 4 SAM bank style Note: Set this field to 10b before switching to VGA mode to avoid VRAM contention while in VGA mode.
shiftclk_mode	23–22	Sets timing patterns for shift clocks (see figure 134): 00 = 1 SAM bank style 01 = 2 SAM bank style 10 = 4 SAM bank style
vad_sht	21	Additional divide for Video Transfer address (see section 8.1.4): 0 = Don't divide by two 1 = divide by two
video_clk_sel	20	Selects the video_clk source for the video section: 0 = PIXCLK 1 = DIVPIXCLK
blank_edge	19	Clock edge that HSYNC and VSYNC are synchronized to: 1 = Falling edge of video_clk 0 = Rising edge of video_clk
reserved	18–16	Reserved, must be set to 0

Figure 91. Memory configuration (mem_config) register (1 of 2)

4.7. VRAM Control Registers, continued

Field	Bits	Contents
crtc_freq	15–13	Sets the frequency of the internal CRTC divided dot clk. Normally set to 000 because the RAMDAC generates the divided clock. 000 = pixel_clk / 1 001 = pixel_clk / 2 010 = pixel_clk / 4 011 = pixel_clk / 8 100 = pixel_clk / 16
shiftclk_freq	12–10	Sets the frequency of the RAMDAC latch clock state machine. Normally set to 000 because the RAMDAC generates the divided clock. 000 = pixel_clk / 1 001 = pixel_clk / 2 010 = pixel_clk / 4 011 = pixel_clk / 8 100 = pixel_clk / 16
hold_reset	9	1 forces memory and video into reset state 0 for normal operation
dac_mode	8	Controls dac read/write signalling mode, see FIgure 144
dac_access_adj	7	Enable RAMDAC back to back transfer checking, see Figure 144 Note: This feature does not work reliably in current silicon. Back-to-back RAMDAC accesses should be separated with NOPs. The recommended method is to read a Power 9100 register, such as pu_config, which is guaranteed to take an appropriate amount of time.
priority_select	6	Video coprocessor priority: 0 = lower than drawing engine 1 = higher than drawing engine
vram_write_adj	5	Adjustment for VRAM write timing template, see Figure 128.
vram_read_adj	4	Adjustment for read timing template, see Figures 126.
vram_miss_adj	3	Adjustment for row miss timing template, see Figures 124 and 125
config[2..0]	2–0	Low-order 3 bits of memory configuration. See Figures 116 through 123

Figure 91. Memory configuration (mem_config) register (2 of 2)

Chapter 5. Commands

Figure 92 summarizes the commands that initiate functions on the Power 9100; it also summarizes the command functions and indicates the section of this data book that describes each command. All commands are accessed via the native register address format. (See section 3.4.4 and figures 54 and 55.)

Note that pixel8 and blit cannot perform 24-bit processing with patterns. For pixel8 and blit, the X address must be programmed in BYTE.

Command	Function	Described in section
Load Coordinates	Load quad coordinates via shorthand method	5.1
Quad	Draw a quadrilateral.	5.2
Blit	Copy a rectangular area of the display form one screen location to another.	5.3
Pixel8	Transfer one word of pixel data from a linear host memory array to a rectangular display memory array in the frame buffer.	5.4
Pixel1	Transfer up to 32 pixels from a a linear host memory array to a rectangular display memory array in the frame buffer.	5.5
Next_pixels	Advance the drawing area for a pixel8 or pixel1 operation.	5.6

Figure 92. Command summary

5.1. Load Coordinates Command

Using the load coordinates command allows you to provide the minimal number of coordinates required for a drawing operation. You specify the type of quadrilateral to draw and specify only the required coordinates. The load coordinates command enables loading of coordinates for quadrilaterals and degenerate quadrilaterals in a shorthand notation that performs the drawing operation with the fewest data transfers. It also facilitates drawing polylines and meshed quads. The load coordinate command is meaningful for write operations only. Figure 93 defines the specific user register address format for accessing the load coordinates pseudo-registers.

Address Format for Load Coordinates Pseudo-register Access

Bits	31–15	14	13	12	11	10	9	8–6	5	4–3	2	1	0
Field	prefix	0	1	1	0	0	1	vtype	rel	yx		0	0
Width	17	6						3	1	2	1	2	

Field	Bits	Contents		
prefix	31–15	See figures 50 and 51.		
decoding address	14–9	Decoding address for read/write operations; see figures 54 and 55.		
vtype	8–6	<u>Entry</u>	<u>Contents</u>	
		000	Point	(specify one coordinate)
		001	Line	(specify two coordinates)
		010	Triangle	(specify three coordinates)
		011	Quadrilateral	(specify four coordinates)
		100	Rectangle	(specify four coordinates)
		101	not used	
		110	not used	
		111	not used	
rel	5	<u>Entry</u>	<u>Action</u>	
		0	Perform screen addressing relative to the window offset	
		1	Perform screen addressing relative to the previous vertex	
yx	4–3	<u>Entry</u>	<u>Action</u>	
		00	Not used	
		01	Load or read 32-bit X value	
		10	Load or read 32-bit Y value	
		11	Load or read 16-bit X value (high 16 bits) and 16-bit Y value (low 16 bits)	
32-bit access	1–0	Must be 00		

Figure 93. Address format for load coordinates pseudo-register access

5.2. Quad Command

The quad command draws and fills the quadrilateral defined by the vertices specified in the coordinate registers (see section 4.4.1). This command sets the busy bit in the status register and performs the drawing operation, scan line by scan line, applying the current pattern, with the left and right Bresenham engines establishing the outlines of the left and right boundaries as the fill engine fills the quadrilateral.

The vertices are numbered from 0,0 starting at the upper left corner of the display. The quad command requires the four vertices of the quadrilateral (x[0],y[0]; x[1],y[1]; x[2],y[2]; and x[3],y[3]). Four vertices are also required for a triangle, line and point. For a triangle, two vertices specify the same coordinate; for a point, all four vertices specify the same coordinate. (You can also use the load coordinates pseudo-register access, see figure 93).

The quad command is initiated with a read operation, which returns the contents of the status register.

The quad command can fail for either of two reasons: the quad is concave or at least one coordinate is out of range of the drawing engine. If the quad cannot be drawn, quad sets the quad_sw bit in the status register (see section 4.4.2) and does not perform the operation.

Figure 94 defines the address format for initiating the quad command.

Quad Command Address Format

31–15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
prefix	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
17	13													2	

Field	Bits	Contents
prefix	31–15	See figures 50 and 51.
decoding address	14–2	Decoding address for read operation; see figure 55.
32-bit access	1–0	Must be 00

Figure 94. Quad command address format

5.3. Blit Command

The blit command copies a rectangular area of the display from one screen location to another. When the original copied area (the source) and the new area (the destination) overlap, the Power 9100 ensures accuracy by performing the copy as though the entire source were moved offscreen and then moved onscreen to the destination.

The blit command is initiated with a read operation, which returns the contents of the status register.

The blit command requires the vertices of the upper left (x[0],y[0]) and lower right (x[1],y[1]) corners of the source, plus the vertices of the upper left [x[2],y[2]) and lower right (x[3],y[3]) corners of the destination. Both areas must be the same size; specifying different sized areas produces undefined results. If the destination is out of range, blit sets the blit_software bit in the status register (see section 4.4.2) and does not perform the copy. Otherwise, blit determines the direction of the blit operation and initiates the copy.

Figure 95 defines the address format for initiating the blit command.

Blit Command Address Format

31–15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
prefix	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
17							13							2	

Field	Bits	Contents
prefix	31–15	See figures 50 and 51.
decoding address	14–2	Decoding address for read operation; see figure 55.
32-bit access	1–0	Must be 00

Figure 95. Blit command address format

5.4. Pixel8 Command

The pixel8 command transfers one word of pixel data from a linear host memory array to a rectangular display memory array in the frame buffer. This operation is useful for writing pixels of various colors to the screen. The array in host memory must be padded out to a 32-bit boundary.

The pixel8 command requires the left edge of the block to be transferred (x[0]), the point at which to begin the transfer (x[1],y[1]), the right edge of the block to be transferred (x[2]), and the increment by which to increase the *y* coordinate at the end of the scan line (y[3]). The pixel8 operation is initiated with a write operation.

This command handles a busy status by holding the processor. Figure 96 defines the address format for initiating the pixel8 command. The pixel8 command can also be accessed by the alternate I/O register encoding also shown in figure 96.

Each pixel8 command writes one word into the frame buffer unless the word being written overlaps the end of the current scan line. Excess data is saved in the pixel8_reg.

Figure 97 illustrates the pixel8 operation with an example.

X coordinates for the pixel8 command must be scaled by the software so that it correctly represents the appropriate number of bytes.

There are two address formats for the pixel8 command. The first format is provided for backwards compatibility with the Power 9000. The second form allows the use of memory to memory string move instructions since it ignores the destination offset. Enough bits are provided to allow up to 8 KBs to be moved in a single string move instruction; this is enough for any length scan line.

Pixel8 Command Address Format (Power 9000 backwards compatible)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
prefix																	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0
17																	13													2	

Pixel8 Command Address Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
prefix																	1	x	x	x	x	x	x	x	x	x	x	x	x	0	0
17																	13													2	

Field	Bits	Contents
prefix	31-15	See figures 50 and 51.
decoding address	14–2	Decoding address for write operation; see figure 54.
32-bit access	1–0	Must be 00

Figure 96. Pixel8 command address formats

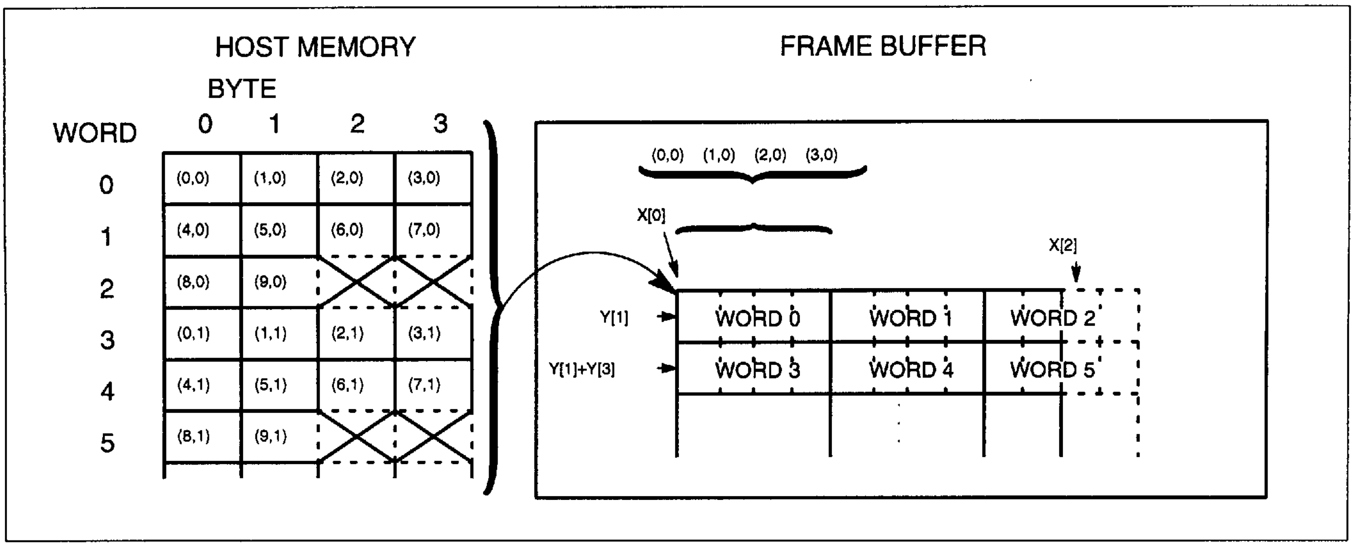


Figure 97. Sample pixel8 host memory to frame buffer transfer

5.5. Pixel1 Command

The pixel1 command expands a monochrome bit pattern to the full color depth. Each bit in the transferred word is expanded to a pixel. The command transfers data for up to 32 pixels from a linear host memory array to a rectangular display memory array in the frame buffer.

The pixel1 command requires the left edge of the block to be transferred (x[0]), the point at which to begin the transfer (x[1],y[1]), the right edge of the block to be transferred (x[2]), and the increment by which to increase the y coordinate at the end of the scan line (y[3]) before beginning the pixel1 command. The pixel1 command specifies the total number of pixels to write (specified in the address as the number of pixels), and provides the pixel data.

5.5.1. PIXEL1 COLOR SELECTION

Each bit of the data presented to the pixel1 command is expanded into a full pixel of color information. If transparent pixel1 mode is disabled (raster.pixel1_transparent = 0) then a 0 bit is expanded into the color contained in the color[0] register and a 1 bit is expanded into the color contained in the color[1] register.

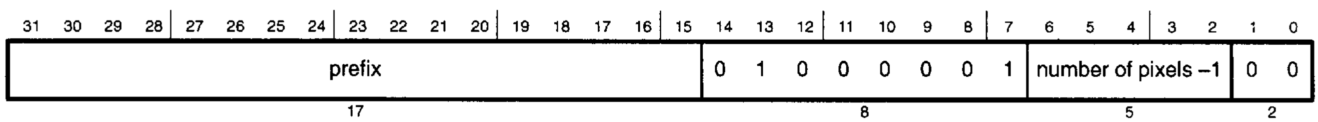
If transparent pixel1 mode is enabled (raster.pixel1_transparent = 1) a zero bit leaves the destination unmodified and a one bit is expanded into the color contained in the color[1] register.

Figure 98 defines the address format for initiating the pixel1 command.

This command handles a busy status by holding the processor.

Unlike the pixel8 command, the pixel1 command does not discard unused bits at the end of the rectangular area. If more bits were specified than fit on the scan line segment of the rectangular area, then the system will automatically continue the pixel1 operation on the next scan line segment with the remaining bits.

Pixel1 Command Address Format



Field	Bits	Contents
prefix	31–15	See figures 50 and 51.
decoding address	14–7	Decoding address for write operation; see figure 54.
number of pixels – 1	6–2	Specifies the number of valid pixels in the data word
32-bit access	1–0	Must be 00

Figure 98. Pixel1 command address format

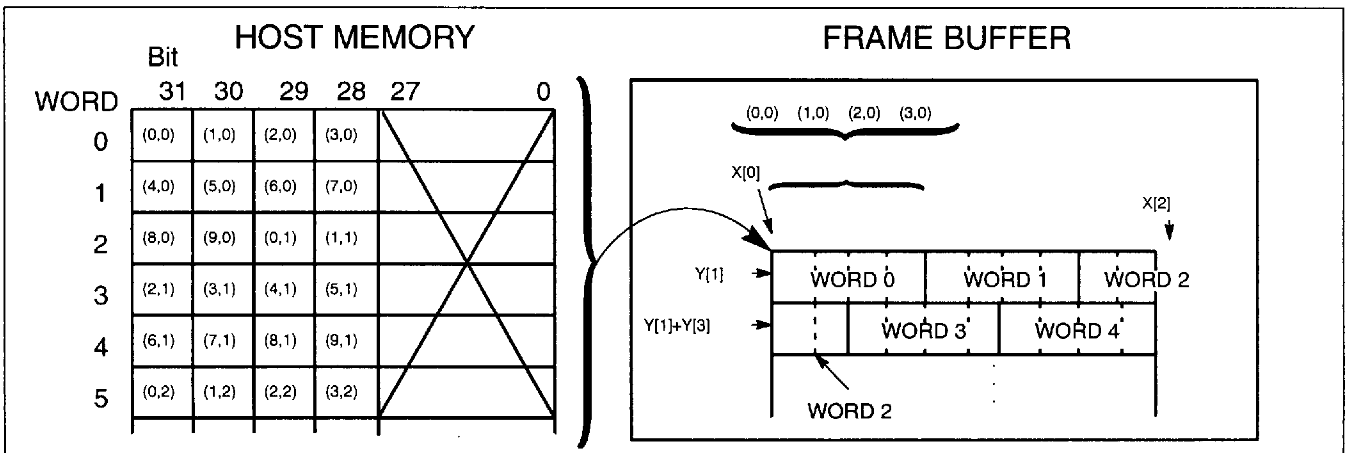


Figure 99. Sample pixel1 host memory to frame buffer data expansion and storage

5.6. Next_pixels Command

The next_pixels command specifies the next area of the frame buffer to load for consecutive pixel8 or pixel1 drawing operations. It advances these drawing operations to the next step in a sequence of operations.

The next_pixels command requires the right edge of the previous block transferred (x[2]) and the top edge of the previous block transferred (y[2]), plus the width of the new transfer (from the data bus).

Figure 100 defines the address format for initiating the next_pixels command.

Next_pixels Command Address Format

31–15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
prefix	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0
17	13													2	

Field	Bits	Contents
prefix	31–15	See figures 50 and 51.
decoding address	14–2	Decoding address for write operation; see figure 54.
32-bit access	1–0	Must be 00

Figure 100. Next_pixels command address format

5.7. Drawing With the Power 9100

The Power 9100 draws quadrilaterals according to the basic rules and in the modes summarized in the following sections.

5.7.1. LEGAL QUADRILATERALS

The Power 9100 draws quadrilaterals and degenerate quadrilaterals (triangles, lines, and points). It handles all quadrilaterals except concave quadrilaterals. A quadrilateral is concave if any scan line crosses more than two boundaries. A request for a concave quadrilateral generates an error. Figure 101 presents examples of non-concave quadrilaterals, which the Power 9100 accepts. Figure 102 presents examples of concave quadrilaterals, which the Power 9100 rejects.

5.7.2. DRAWING MODES

The Power 9100 supports two drawing modes: X11 and oversized. (The drawing mode is selected via the raster register defined in section 4.5.5.)

The X11 drawing mode strictly conforms to the drawing rules employed by X-windows:

1. The boundaries of the quadrilateral are considered to be infinitely thin.
2. Only those pixels whose centers are completely within the boundaries of the quadrilateral are touched.
3. If the center of a pixel is exactly on the boundary, it is touched if and only if the interior is immediately to its right (in the increasing x direction).
4. A pixel with its center on a horizontal boundary is handled as a special case; it is touched if and only if the interior is immediately below (in the increasing y direction) and the boundary is immediately to its right (in the increasing x direction).

In addition to touching the pixels touched according to the X11 rules, the oversized drawing mode also touches the pixels along the boundary of the quadrilateral as drawn by the classic Bresenham algorithm. In this mode, the Power 9100 draws a line or a point as an infinitely thin quadrilateral. In the X11 drawing mode, no pixels would be touched because the quadrilateral has no interior. In oversized mode, however, using the classic algorithm that assumes that a line from [x0,y0] to [x1,y1] touches the same pixels as a line from [x1,y1] to [x0,y0], the pixels are touched. This method also satisfies the X11 definition of a thin line (from Gettys and Scheifler): "If a line is drawn unclipped from [x1,y1] to [x2,y2] and if another line is drawn unclipped from [x1+dx,y1+dy] to [x2+dx,y2+dy], a point [x,y] is touched by drawing the first line only if the point [x+dx,y+dy] is touched by drawing the second line."

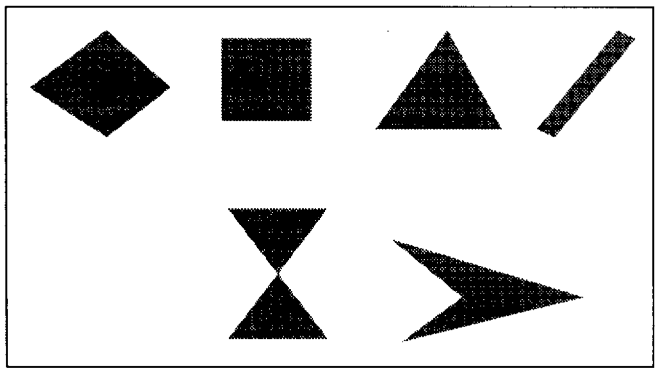


Figure 101. Samples of non-concave (legal) quadrilateral

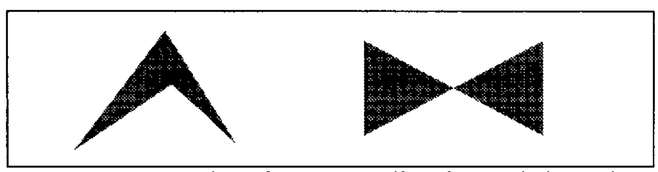


Figure 102. Samples of concave (illegal) quadrilaterals

5.7.3. BASIC QUAD DRAWING METHODS

To draw quadrilaterals, supply all four points of each quadrilateral in the device coordinate registers (to specify a degenerate quadrilateral, repeat one or more coordinates), or specify the quad type and its vertices via the load coordinates pseudo-registers, which enable you to specify fewer coordinates and draw a new quadrilateral sharing the last vertices of the previously drawn quadrilateral. In either case, you issue the quad command to execute the draw.

Unlike the other drawing commands, the X values for the quad command are not scaled into bytes. They are specified in pixels.

5.7.4. NEGATIVE COORDINATES

The Power 9100 uses data in two's complement format. Any portion of a draw operation that is in a negative quadrant, however, is not drawn. This applies to quad draw, blit destination, pixel1, and pixel8. When the source for a blit contains negative coordinates, the results are indeterminate.

5.7.5. FLOW CONTROL

The host processor issues read and write operations which the Power 9100 accepts. Most host buses require that any read or write operation be guaranteed to complete in a certain amount of time (for example, 10 µseconds, or about 500 cycles at 50 MHz). Because some Power 9100 operations, such as clearing the screen, can require hundreds of thousands of cycles, the host processor must be able to issue sequences of these operations in a manner that does not violate the timeout constraint. The Power 9100 handles such problems by a specific system software protocol. Protocol violations are not detected.

Accesses to non-drawing engine registers cause no problems and can be made without regard to the current state of the drawing engine. The answer is always returned within a few cycles.

Accesses to drawing engine registers are valid only when the drawing engine is idle (the status register busy bit is zero).

The sequence to initiate the quad and blit drawing operations comes in two flavors. The first flavor is when the software does not know what the previous drawing operation was or whether it has completed or not. In this case the software must ensure that the drawing engine is idle (with the status register busy bit set to zero) sometime before requesting the initiation of a quad or a blit operation by reading from the appropriate address.

The second flavor comes when the software knows that the previous operation is a quad or a blit and the next operation is also a quad or a blit (warning: this case only applies to quad followed by quad or blit followed by blit but not to quad followed by blit or blit followed by quad). In this case the software can request initiation of the subsequent quad or blit simply by reading the appropriate command address and checking the status.issued_qbN bit. If the drawing engine is busy, the Power 9100 ignores the request for the operation. The software is responsible for reissuing the request . It can either poll by continuously reissuing the request or it can request an interrupt when the drawing engine is ready.

The pixel1 and pixel8 operations wait for the drawing engine to become idle, holding the host bus; sequences of these operations can be executed without checking the status of the drawing engine. It is assumed that any preceding operation is short. When this is not certain, the software should wait for the drawing engine to be idle (with the status register busy bit set to zero) before issuing the command.

Table 103 summarizes which status register bit should be checked before initiating specific operations or register writes.

5.7.6. EXCEPTION HANDLING

The Power 9100 determines whether a requested drawing operation is legal and within range and sets bits in the status register to indicate the result. Software can check the status register to detect an exception.

Before performing:	Check status register:
Parameter engine register write	No status check required
Drawing engine register write	Busy bit (bit 30) for zero
Quad draw	Issue_qbN bit (bit 31) for zero
Blit	Issue_qbN bit (bit 31) for zero
First pixel1 or pixel8 in a series (when switching from reads to writes)	Busy bit (bit 30) for zero

Figure 103. Status register checks

Chapter 6. Host Interface

6.1. Signal Description Conventions

Each signal is identified with a specific type that indicates input/output status, etc.

6.1.1. OUTPUT SIGNALS

Signals of this type are always driven to a valid voltage level. The output driver is not capable of being placed into the high-impedance state. If shown as OUTPUT(HIGH) then the signal is driven HIGH whenever reset is asserted.

6.1.2. INPUT SIGNALS

Signals of this type are never driven by the Power 9100.

6.1.3. TRI-STATED OUTPUT SIGNALS

Signals of this type can be driven high or low or placed into the high-impedance state. Generally these signals are forced into a tri-stated condition during a reset: this is indicated as TRI-STATED(TRI).

6.1.4. BIDIRECTIONAL SIGNALS

Signals of this type are can be either inputs or outputs.

6.2. Host Bus Interface

The Power 9100 directly supports two separate buses: VL and PCI. Other buses, such as 486 CPU, can easily be supported with small amounts of additional glue logic.

The bus mode is configured at reset time and cannot be changed without a complete reset. See section 3.3.3 for more information on reset configuration.

The following sections document the timing and functionality of the pins that are specific to each bus interface.

6.2.1. I/O ADDRESS DECODING

The PCI specification requires that address decoding be done on all 32 bits of address. All address decodes in the Power 9100 I/O or memory are done as a full 32 bits. In VL, the host bus interface logic ensures that the top 16 bits of an I/O address are ignored (that is, treated as zero).

The same situation is not true for memory addresses. The host interface logic assumes that all 32 bits are to be decoded.

6.2.2. PALETTE SNOOPING

Palette snooping is an emulation mode involving DAC write operations. This mode is generally used with the VGA feature connector. When this mode is enabled (CONFIG[4].vga_palette_snoop = 1, see figure 32) and the Power 9100 is in emulation mode (CONFIG[65].modeselect = 1, see figure 44), then write operations to the DAC addresses cause no visible response on the host bus, but are actually performed internally; consequently, the system is responsible for ensuring flow control for these operations. Read operations are responded to normally. In the N4C–A2 silicon, this feature does not work in PCI bus mode because it does not ignore aborted cycles

6.3. PCI Bus Operation

The Power 9100 directly supports the 32-bit version of the PCI bus without any external logic. The Power 9100 never becomes a bus master and therefore does not use all of the PCI signals.

6.3.1. PCI BUS SIGNAL LIST

The PCI bus signal list is shown in figure 104.

6.3.2. PCI BUS TIMINGS

Timings for the PCI bus are divided into groups as shown in figure 105.

6.3.3. PCI BUS CONFIGURATION REGISTERS

The PCI configuration address space is used to access the Power 9100 configuration registers. The configuration registers have been arranged to conform to the PCI configuration register space standard. See section 3.3.1.

Signal	Type	Description
AD[31..0]	Input/Output	The address and data bus. This is tri-stated when RESET– is asserted.
C/BE–[3..0]	Input	Command and byte enable bus.
PAR	Input/Output	Even parity for the AD[31..0] and C/BE–[3..0] buses. The Power 9100 generates correct parity for read transfers. It does not check for correct parity on write transfers.
FRAME–	Input	Cycle frame. Indicates the beginning of a transaction.
TRDY–	Tri-stated (TRI)	Target ready. Indicates that the Power 9100 is ready to accept the transaction.
IRDY–	Input	Initiator ready. Indicates that the bus master is ready.
STOP–	Tri-stated (TRI)	Indicates that the Power 9100 is requesting the current transaction to stop.
IDSEL	Input	An alternate chip select for initialization transfers.
DEVSEL–	Tri-stated (TRI)	Asserted by the Power 9100 to claim a transaction.
IRQ	Tri-stated (TRI)	Interrupt request. Must be buffered, as this signal is not open-collector or active low as required by the PCI specification.
CLK	Input	Bus clock.

Figure 104. PCI bus signal list

Group Type	Description
GROUP 1: Input signals	C/BE–[3..0], FRAME–, IRDY–, and IDSEL. As measured relative to the 1.5V level of the rising edge of BCLK. Setup times are 7 ns and hold times are 0 ns.
GROUP 2: Input/output signals	AD[31..0] and PAR. When operating as input signals they are the same as group 1: 7ns setup and 0ns hold. When operating as outputs: The turn-on time is a minimum of 2ns. The turn-off time is a maximum of 28ns. The output valid time is a minimum of 2ns and a maximum of 11ns..
GROUP 3: Clock signal	A maximum frequency of 33Mhz with a minimum duty cycle of 40% for high and low portions.

Figure 105. Timings for PCI bus

6.3. PCI Bus Operation, continued

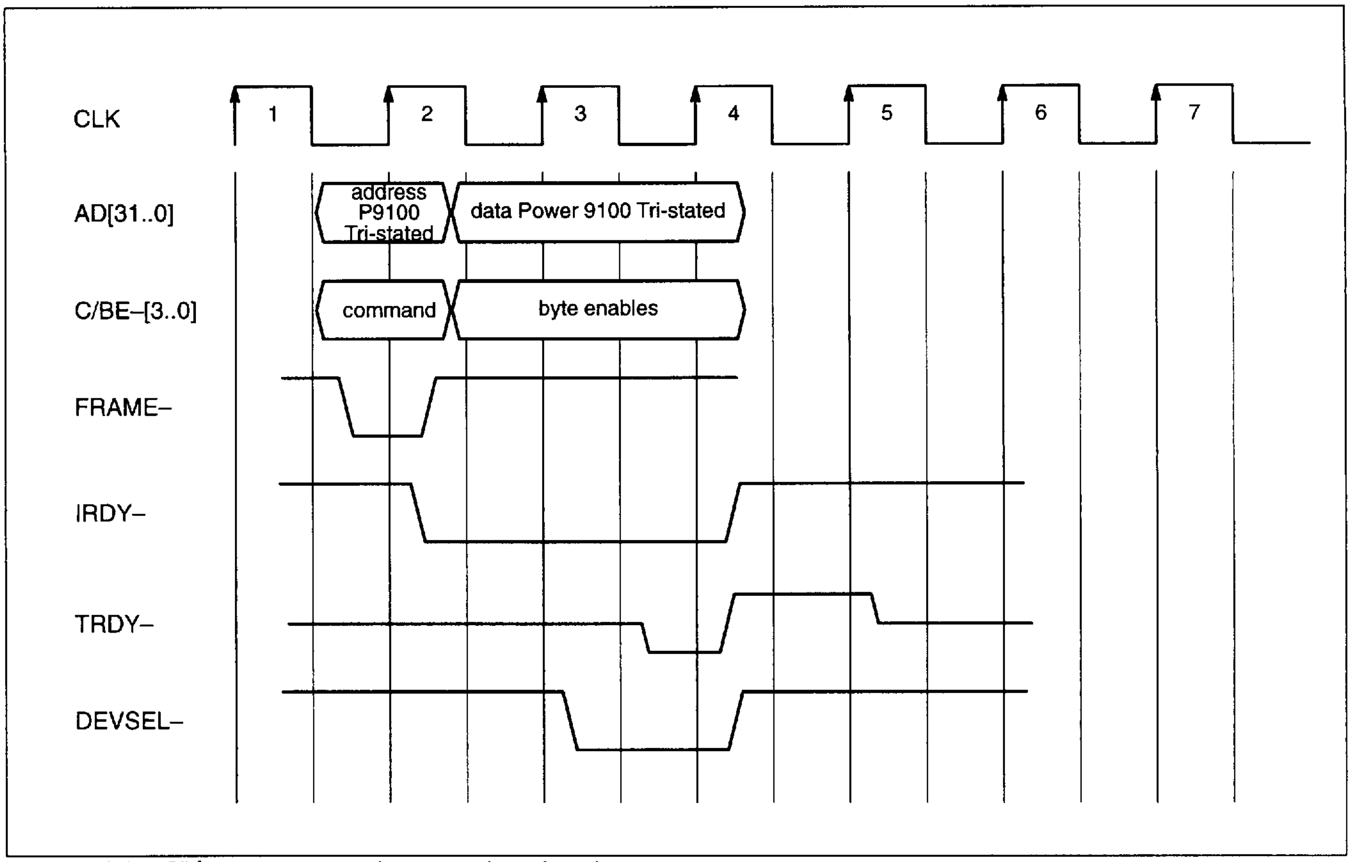


Figure 106. PCI bus memory write operation, 0 wait states

6.3. PCI Bus Operation, continued

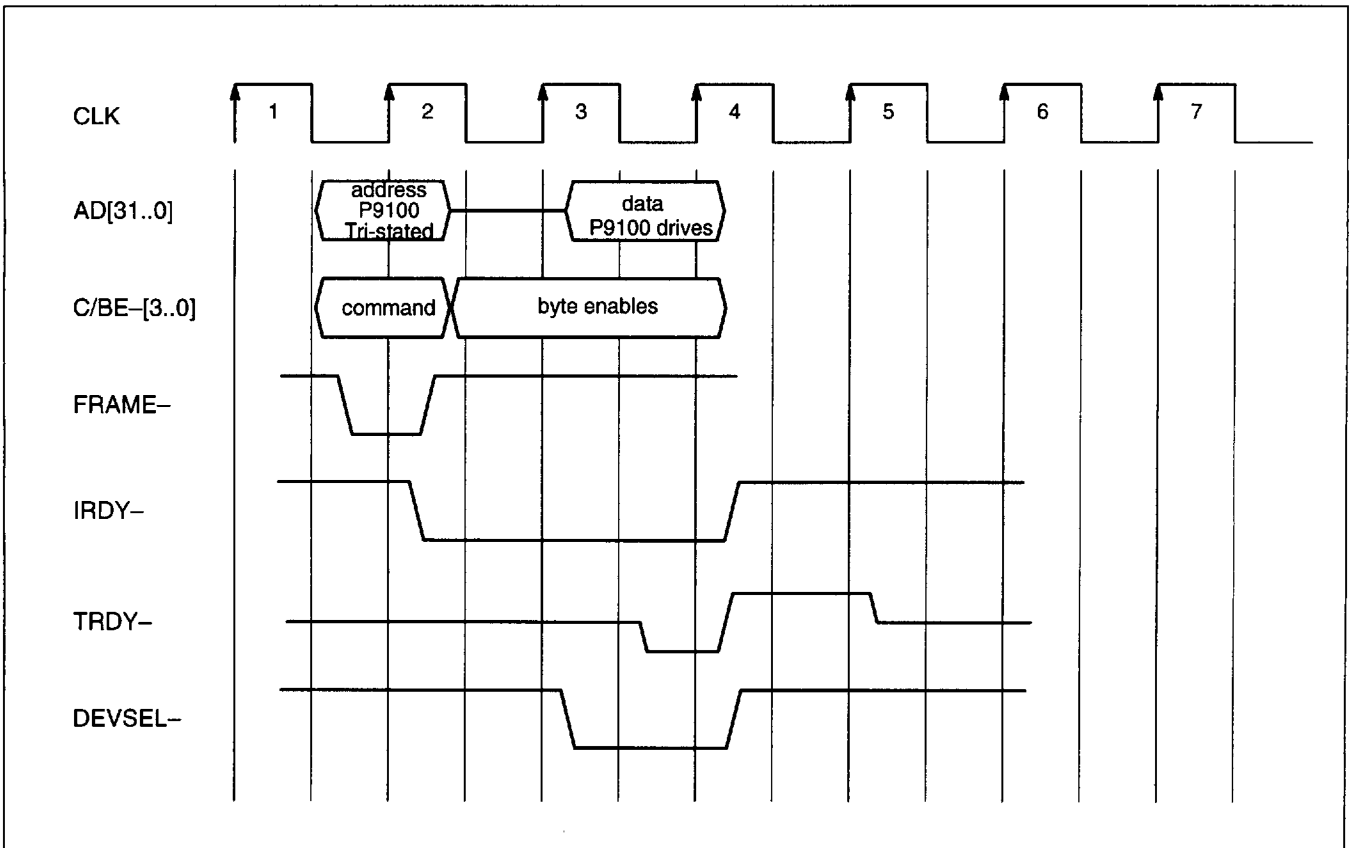


Figure 107. PCI bus memory read operation, 0 wait states

6.3. PCI Bus Operation, continued

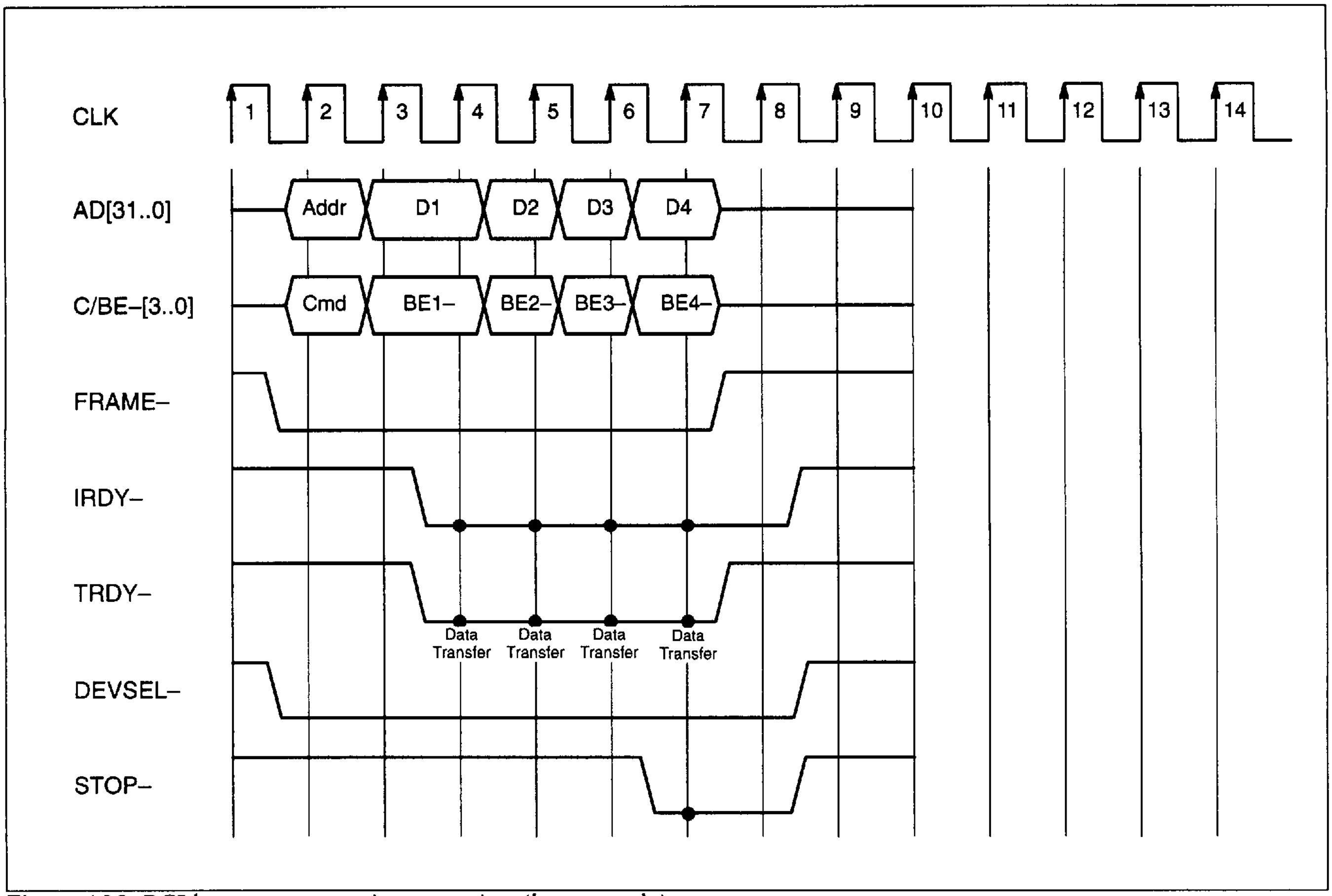
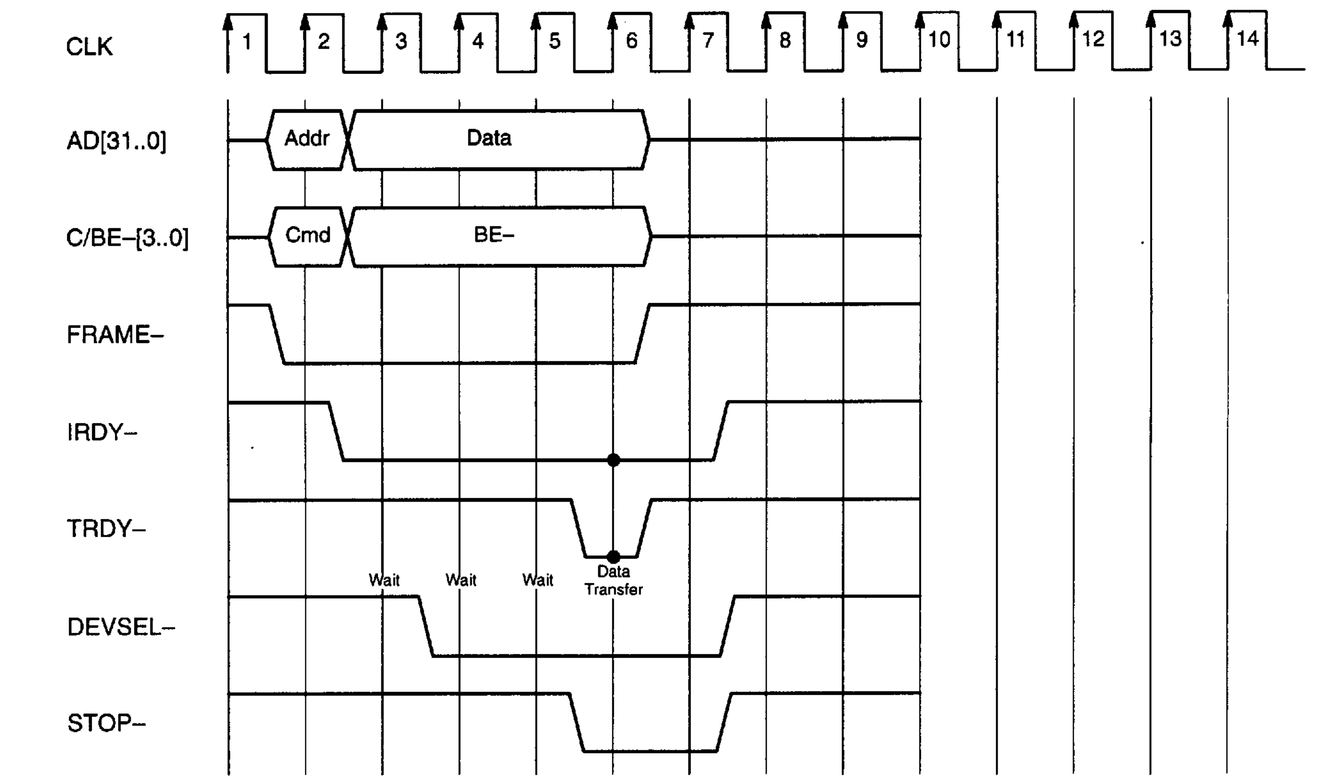


Figure 108. PCI bus memory write operation (burst mode)

6.3. PCI Bus Operation, continued

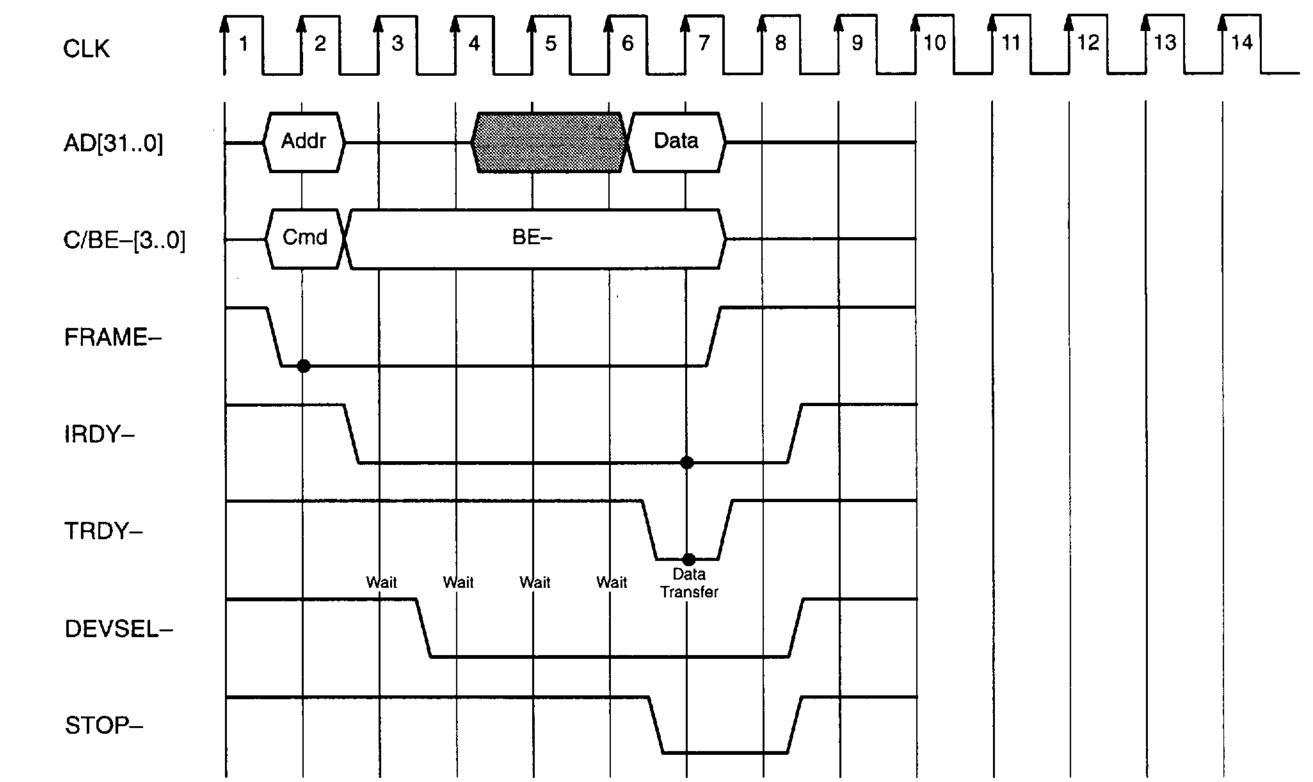


Number of wait states is indicative only. The actual number varies according to the internal state of the Power 9100.

Figure 109. PCI bus I/O write operation (burst not supported during I/O writes)

9663826 0003796 187

6.3. PCI Bus Operation, continued



Number of wait states is indicative only. The actual number varies according to the internal state of the Power 9100.

Figure 110. PCI bus read operation (burst not supported during reads)

6.4. VL Bus Operation

The Power 9100 supports VL bus functionality.

The Power 9100 does not use burst mode on the VL bus. Therefore, the BLAST– and BRDY– signals of the VL bus remain unconnected. Also, fast write mode is not supported.

6.4.1. VL BUS SIGNAL LIST

The VL bus signal list is shown in figure 111.

Signal	Type	Description
ADR[31..2]	Input	The address bus.
DATA[31..0]	Input/Output	The bidirectional data bus.
BE[3..0]–	Input	The byte enable lines.
M/IO–	Input	Memory/IO address indicator.
W/R–	Input	Read or write indicator.
D/C–	Input	Data or code access indicator.
ADS–	Input	Address strobe. This indicates the start of a VL bus cycle.
LDEV–	Output (high)	Local device. This is driven by the Power 9100 to "claim" a VL bus transfer. It is driven combinatorially from ADR[31..2], M/IO– and D/C–.
LRDY–	Tri-stated (TRI)	Local Ready. Indicates that the Power 9100 has completed its portion of the transfer.
RDYRTN–	Input	Ready Return. Indicates when the system considers a read transfer to have completed.
IRQ	Tri-stated (TRI)	Interrupt request.
RESET–	Input	System Reset.
LCLK	Input	Local Bus clock.

Figure 111. VL bus signal list

6.4.2. VL BUS TIMINGS

Timings for the VL bus are divided into groups as shown in figure 112.

Group Type	Description
GROUP 1: input signals to the Power 9100	ADR[31..2], BE[3..0]–, M/IO–, W/R–, D/C–, ADS–, RESET–, RDYRTN–. These signals have a minimum setup time of 7ns. and a minimum hold time of 3ns. The timings are referenced to the 1.5V level of the rising edge of the LCLK signal.
GROUP 2: The LRDY signal	LRDY–. This signal has a maximum output delay of 10ns and a minimum output valid time of 3ns into a maximum load of 100pF. The timings are referenced to the 1.5V level of the rising edge of the LCLK signal. This signal also becomes tri-stated during certain operations. The turn-off time is a maximum of 7ns and is referenced to the 1.5V level of the falling edge of LCLK.
GROUP 3: The Input/Output data bus	DATA[31..0]. During write transfers (i.e., an input to the Power 9100) the timings for this bus are the same as the Group 1 signals: setup = 7ns, hold = 3ns. For read transfers (i.e., an output from the Power 9100), max output delay = 15ns into a 100pF load, turn off time must be no more than 7ns.
GROUP 4: The LDEV signal	LDEV–. This signal is generated combinatorially from ADR[31..2], M/IO– and D/C–it has a maximum output delay of 15ns and a minimum output valid time of 7ns into a maximum load of 20pF. The timing is referenced relative to a change in ADR[31..2], M/IO– and D/C–.
GROUP 5: The clock	LCLK. The bus clock. It has a maximum frequency of 50Mhz (15ns). A minimum duty time of 40% (i.e., minimum clock low/high time is 6ns).

Figure 112. Timings for VL bus

6.4. VL Bus Operation, continued

6.4.3. VL BUS OPERATIONS WAVEFORMS

The LRDY– waveform shows the state of the output driver of the Power 9100. Since this signal is connected to VDD via a pull-up on the system bus, its actual state will be high when there are no buffers trying to actually drive it (show as high-impedance state on the timing diagrams).

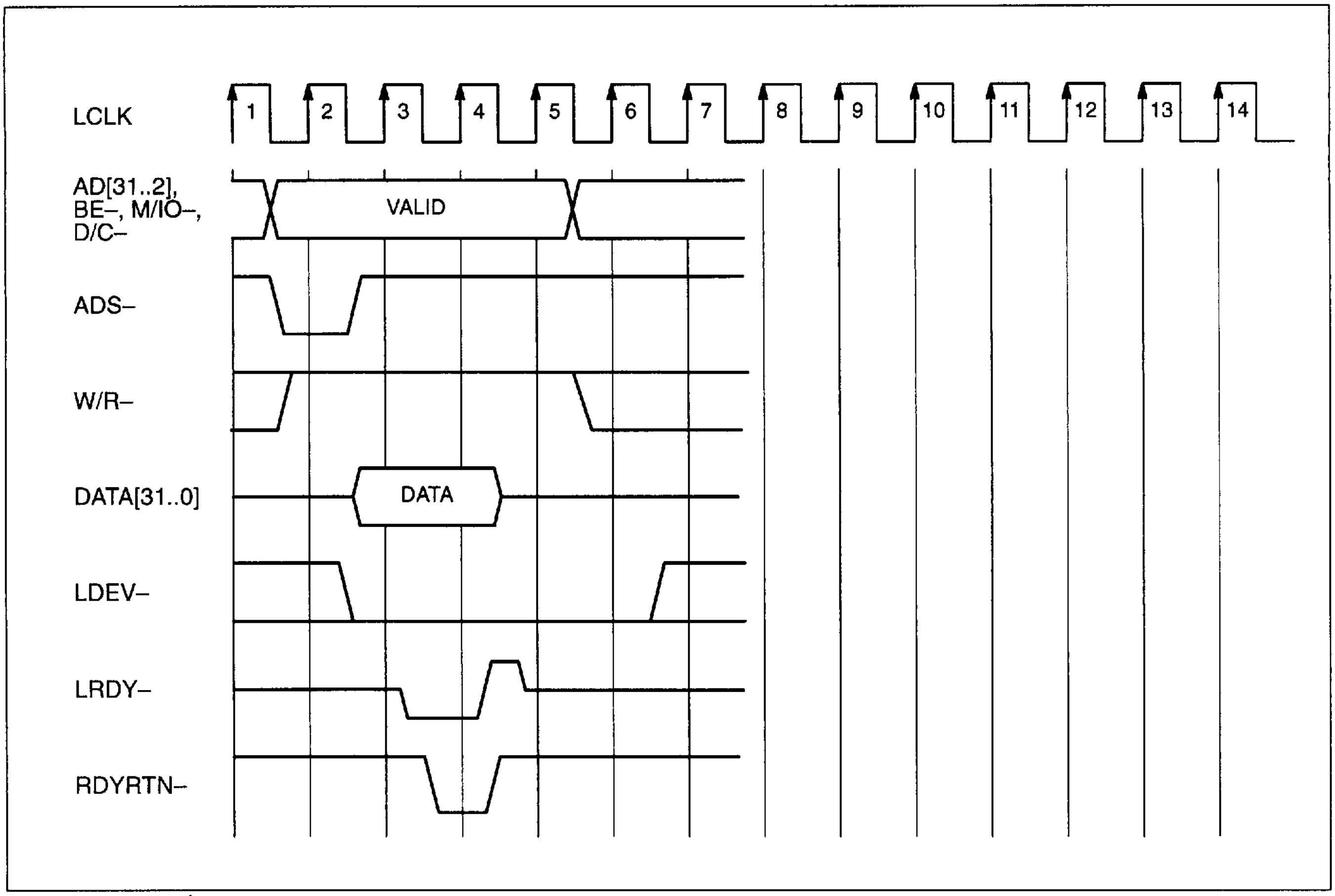
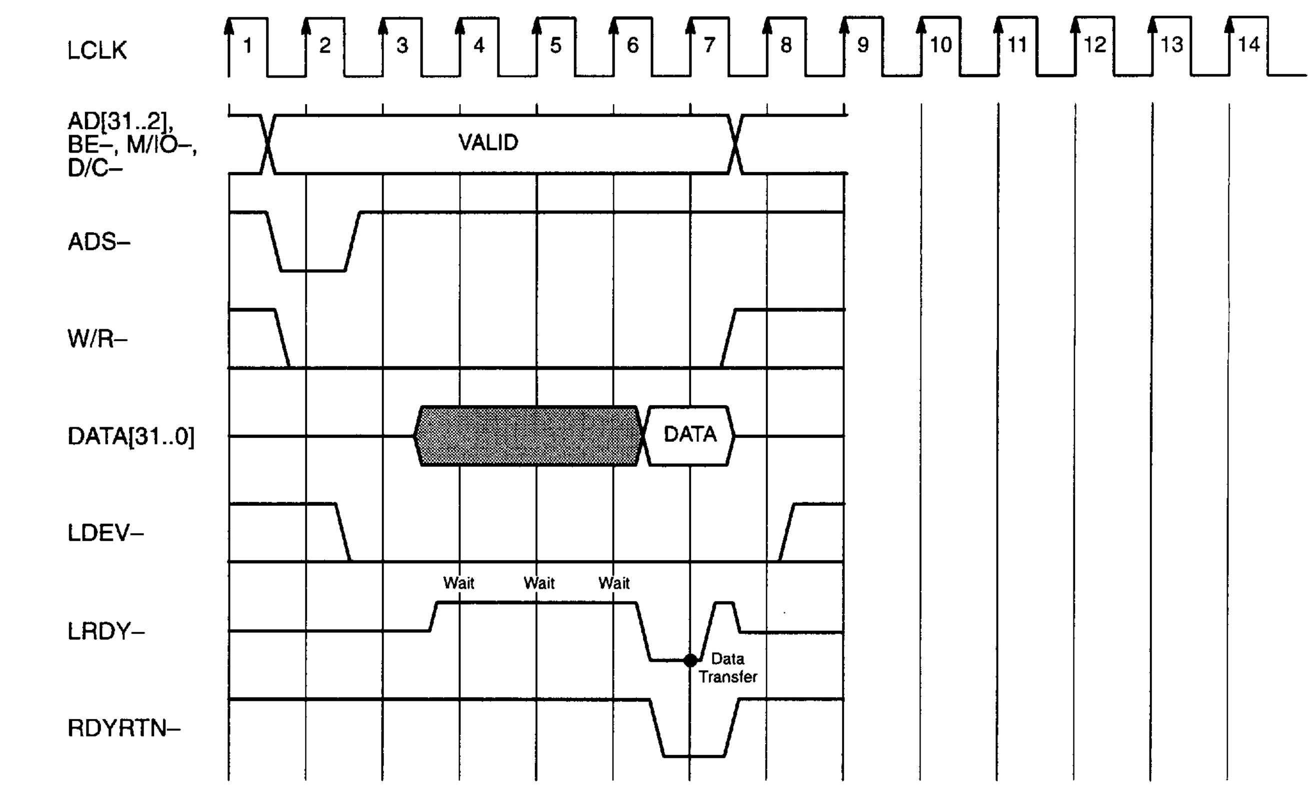


Figure 113. VL bus write operation (no wait states)

6.4. VL Bus Operation, continued



Number of wait states is indicative only. The actual number varies according to the internal state of the Power 9100.

Figure 114. VL bus read operation (non-synchronized RDYRTN–)

6.4. VL Bus Operation, continued

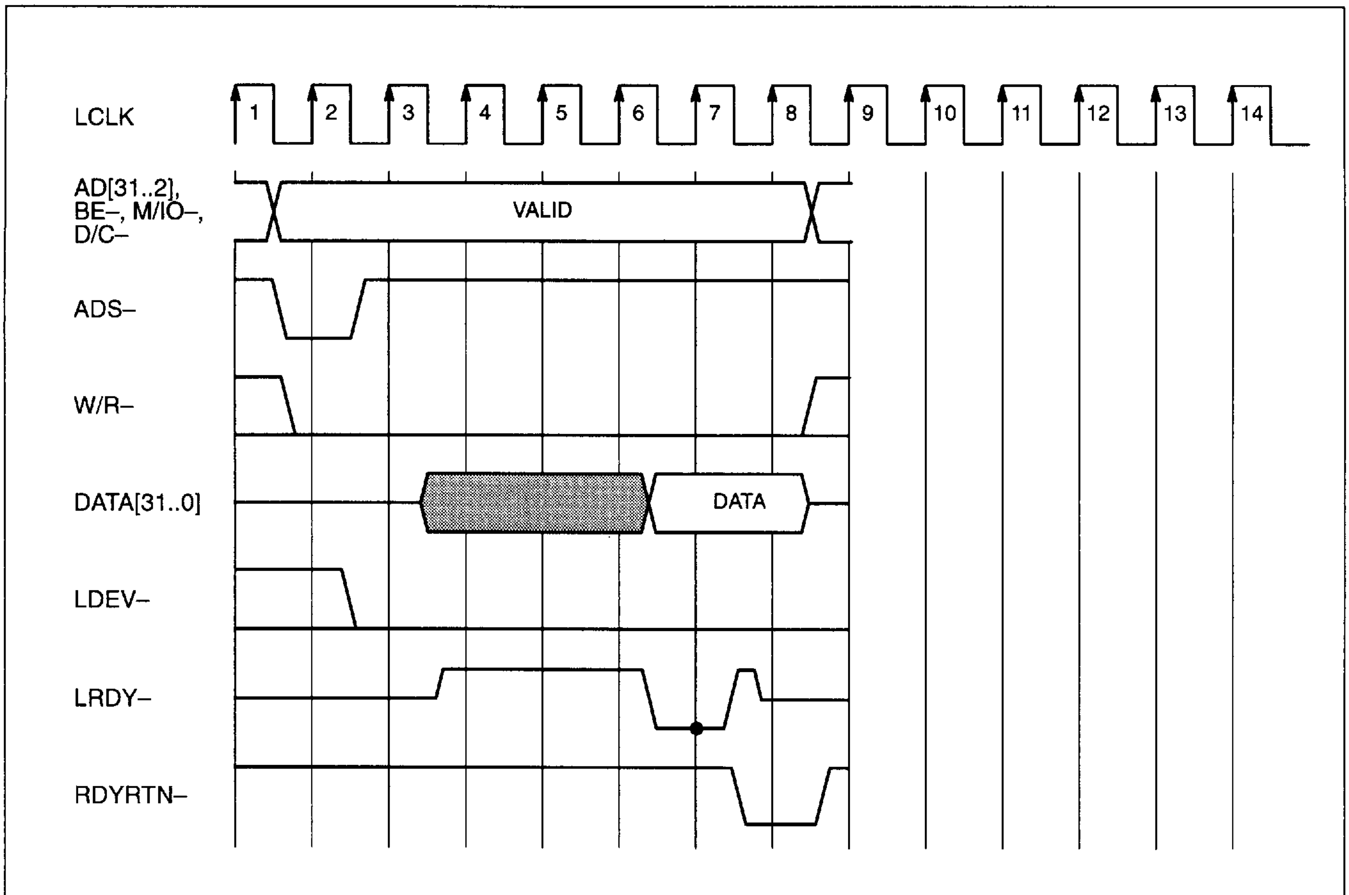


Figure 115. VL bus read operation (resynchronized RDYRTN–)

Chapter 7. Frame Buffer Interface

7.1. Frame Buffer Design Notes

This section presents information necessary to select, configure, and connect VRAMs. The first set of figures (116 through 123) shows how to wire together the memory chips for one-, two-, and four-bank memory configurations.

See figure 60, system configuration register bits [9..10], and figure 87, screen repaint timing control (srtctl) register bit 3, for more information about defining the buffer.

The frame buffer is operated in big-endian mode. The Power 9100 displays MD[31..24] first, followed by MD[23..16], MD[15..8], and MD[7..0]. In a 4 MB design, the banks are displayed in order: 0, 1, 2, 3.

For a 2 MB to 4 MB upgrade board using a 64-bit DAC, use figure 120, mem_config.config = 0111 and connect banks 1 and 2 to sockets or module connectors. Banks 0 and 3 are the base memory.

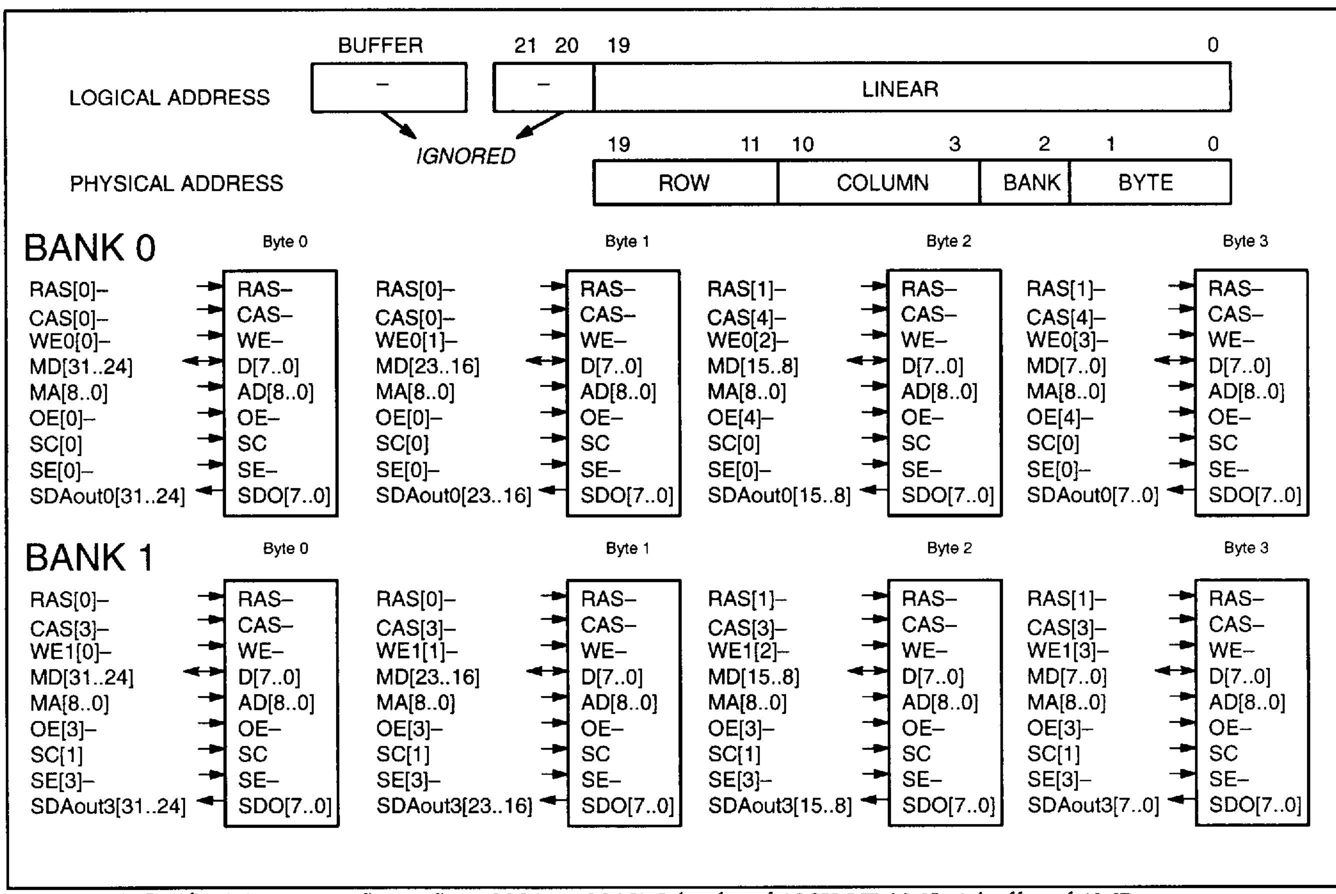


Figure 116. Config 1 (mem_config.config = 0001 or 0010) 2 banks of 128K VRAMS, 1 buffer of 1MB

7.1. Frame Buffer Design Notes, continued

Figure 117 assumes that you are using a 64-bit wide RAMDAC. If you are using a 32-bit wide RAMDAC, use external logic to generate additional serial enable (SE–) signals to address banks 1 and 2. Refer to the *Power 9100/Video Power Board Application Note* for more information.

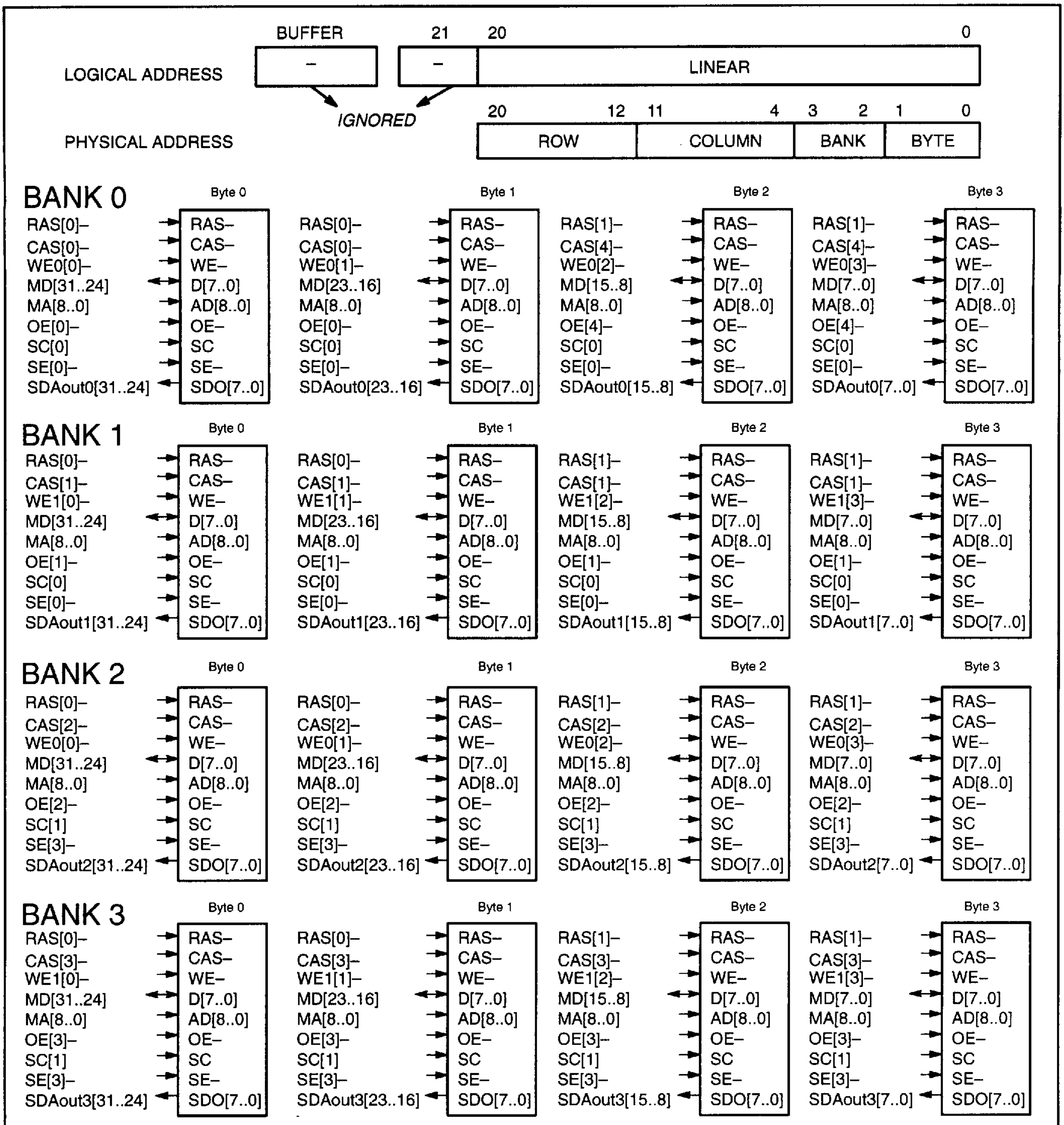


Figure 117. Config 3 (mem_config.config = 0011) 4 banks of 128K VRAMS, 1 buffer of 2MB

7.1. Frame Buffer Design Notes, continued

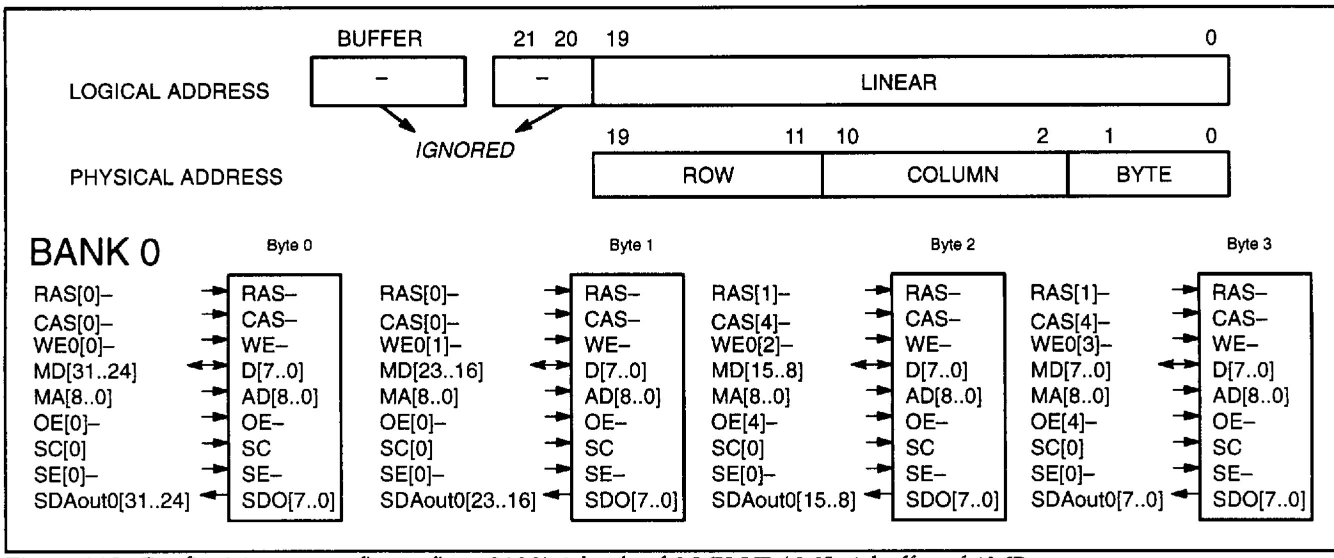


Figure 118. Config 4 (mem_config.config = 0100) 1 bank of 256K VRAMS, 1 buffer of 1MB

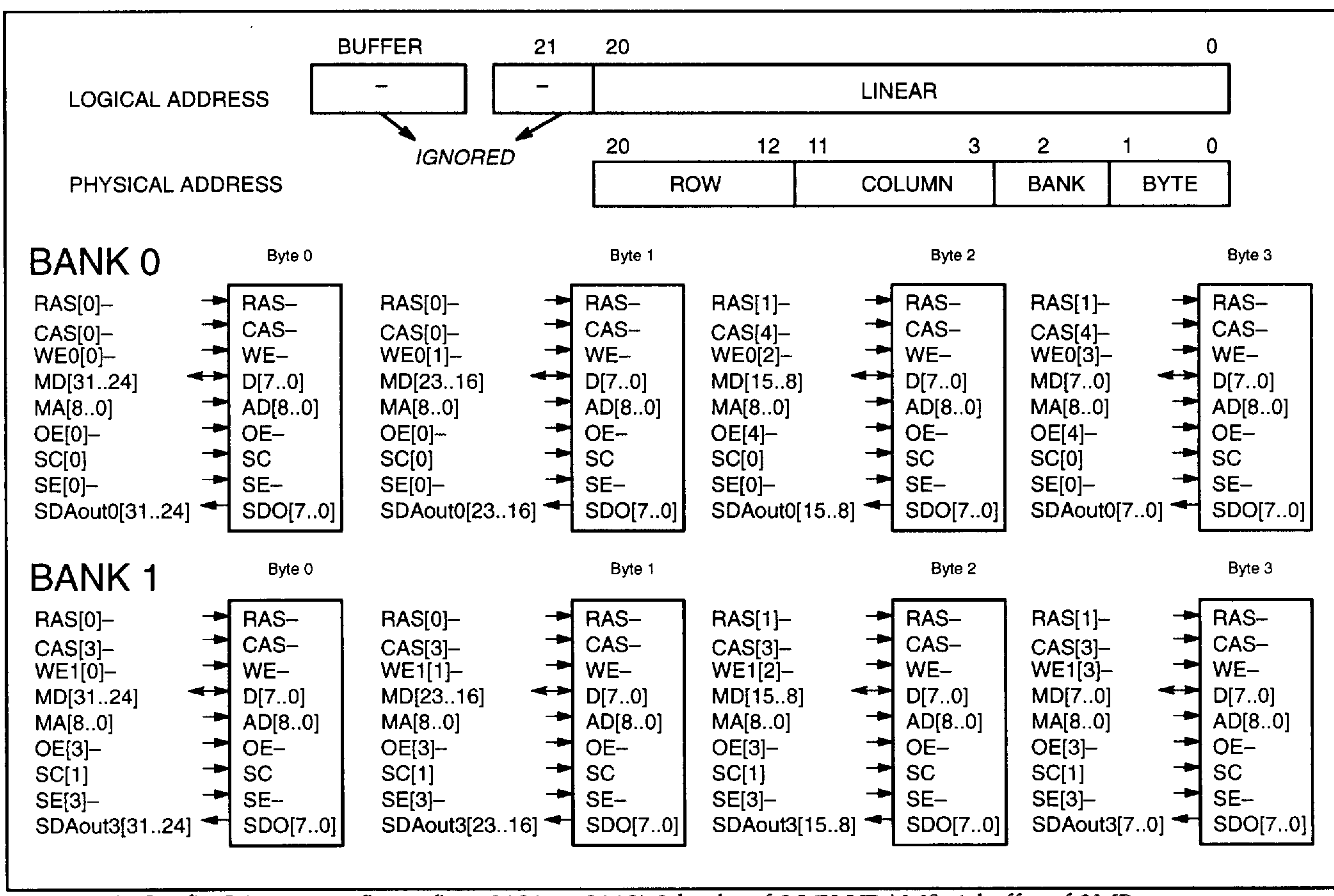


Figure 119. Config 5 (mem_config.config = 0101 or 0110) 2 banks of 256K VRAMS, 1 buffer of 2MB

7.1. Frame Buffer Design Notes, continued

Figure 120 assumes that you are using a 64-bit wide RAMDAC. If you are using a 32-bit wide RAMDAC, use external logic to generate additional serial enable (SE–) signals to address banks 1 and 2. Refer to the *Power 9100/Video Power Board Application Note* for more information.

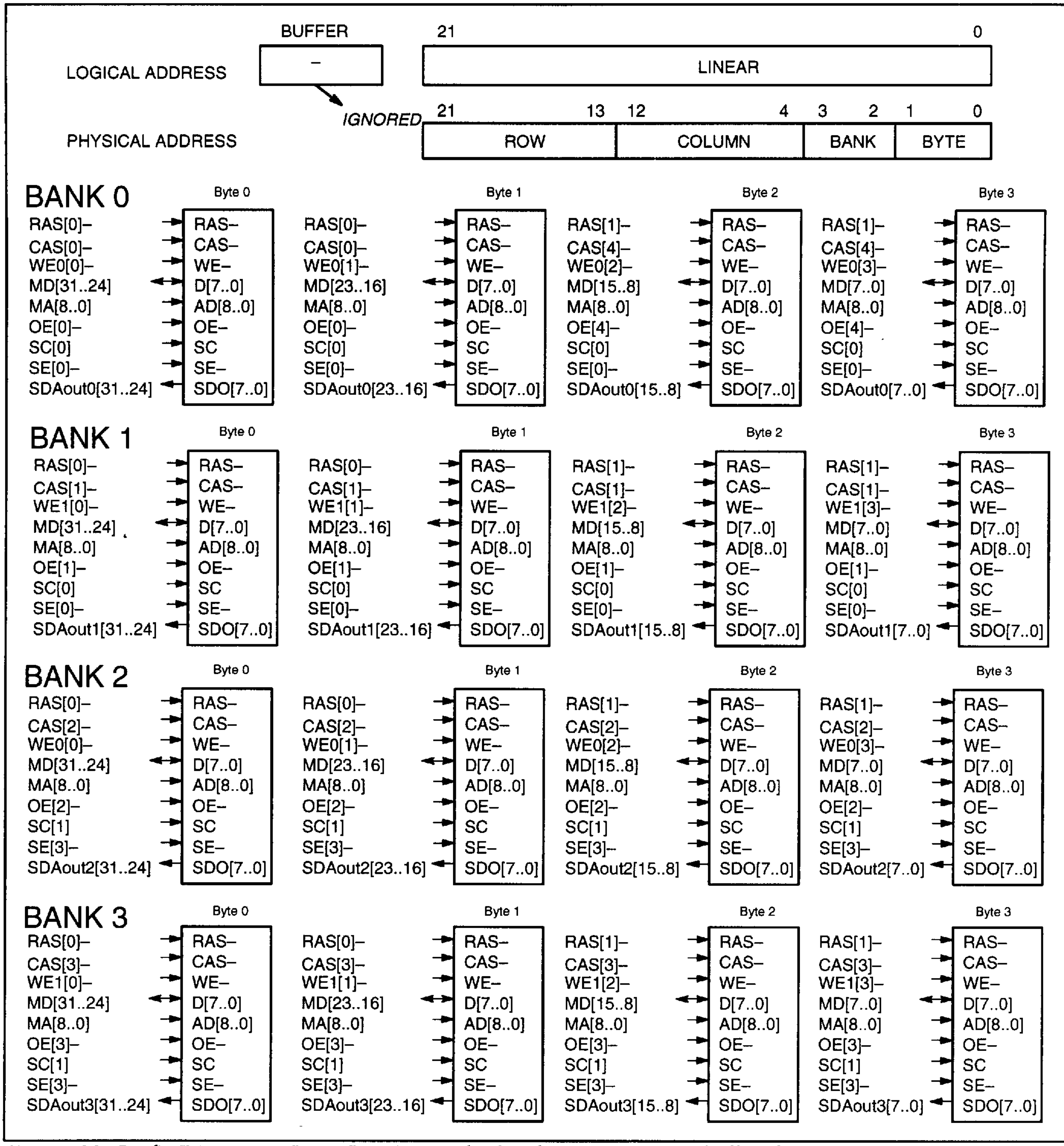


Figure 120. Config 7 (mem_config.config = 0111) 4 banks of 256K VRAMS, 1 buffer of 4MB

7.1. Frame Buffer Design Notes, continued

Figure 121 assumes that you are using a 64-bit wide RAMDAC. If you are using a 32-bit wide RAMDAC, use external logic to generate additional serial enable (SE–) signals to address banks 1 and 2. Refer to the *Power 9100/Video Power Board Application Note* for more information.

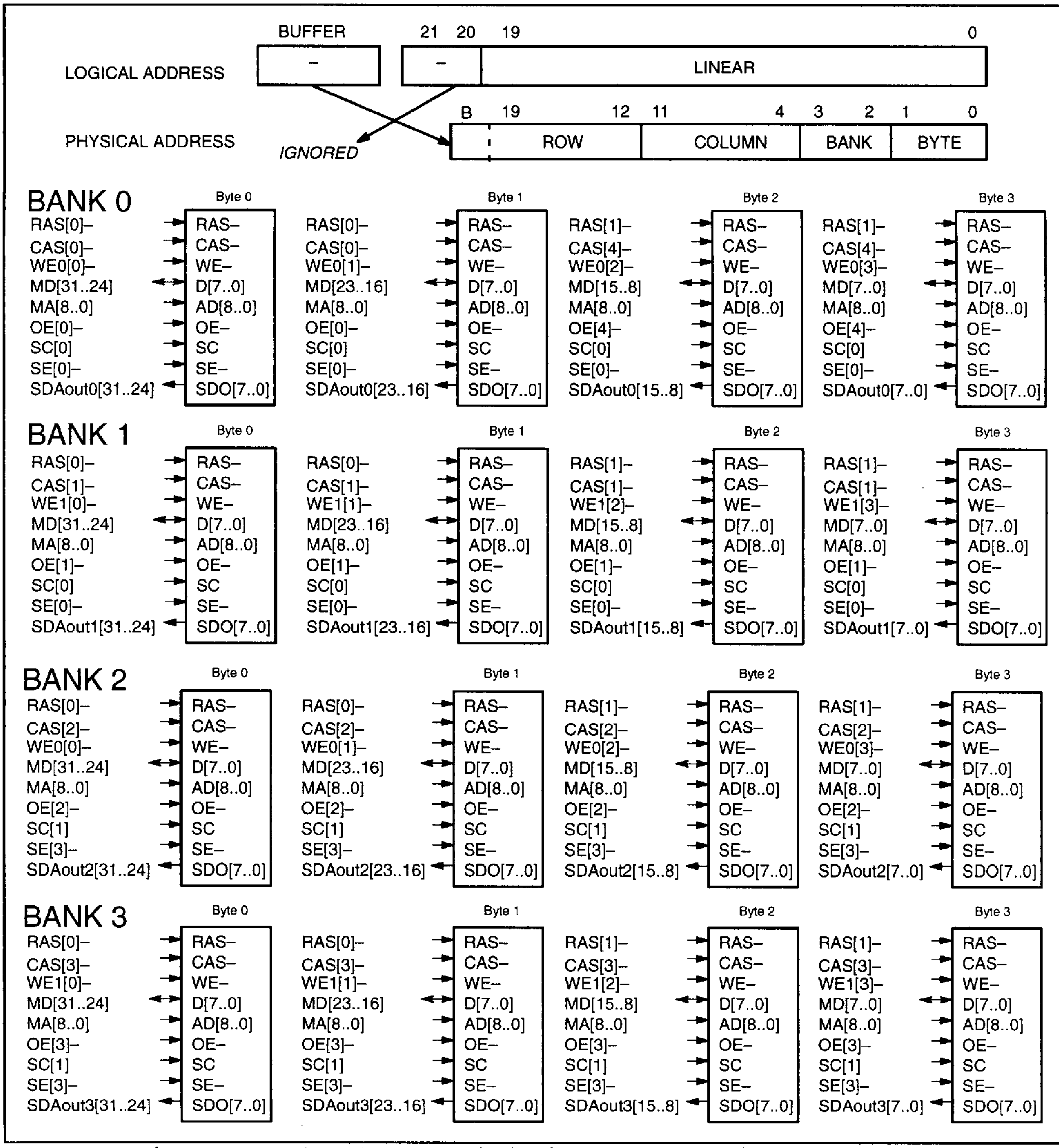


Figure 121. Config 11 (mem_config.config = 1011) 4 banks of 128K VRAMS, 2 buffers of 1MB

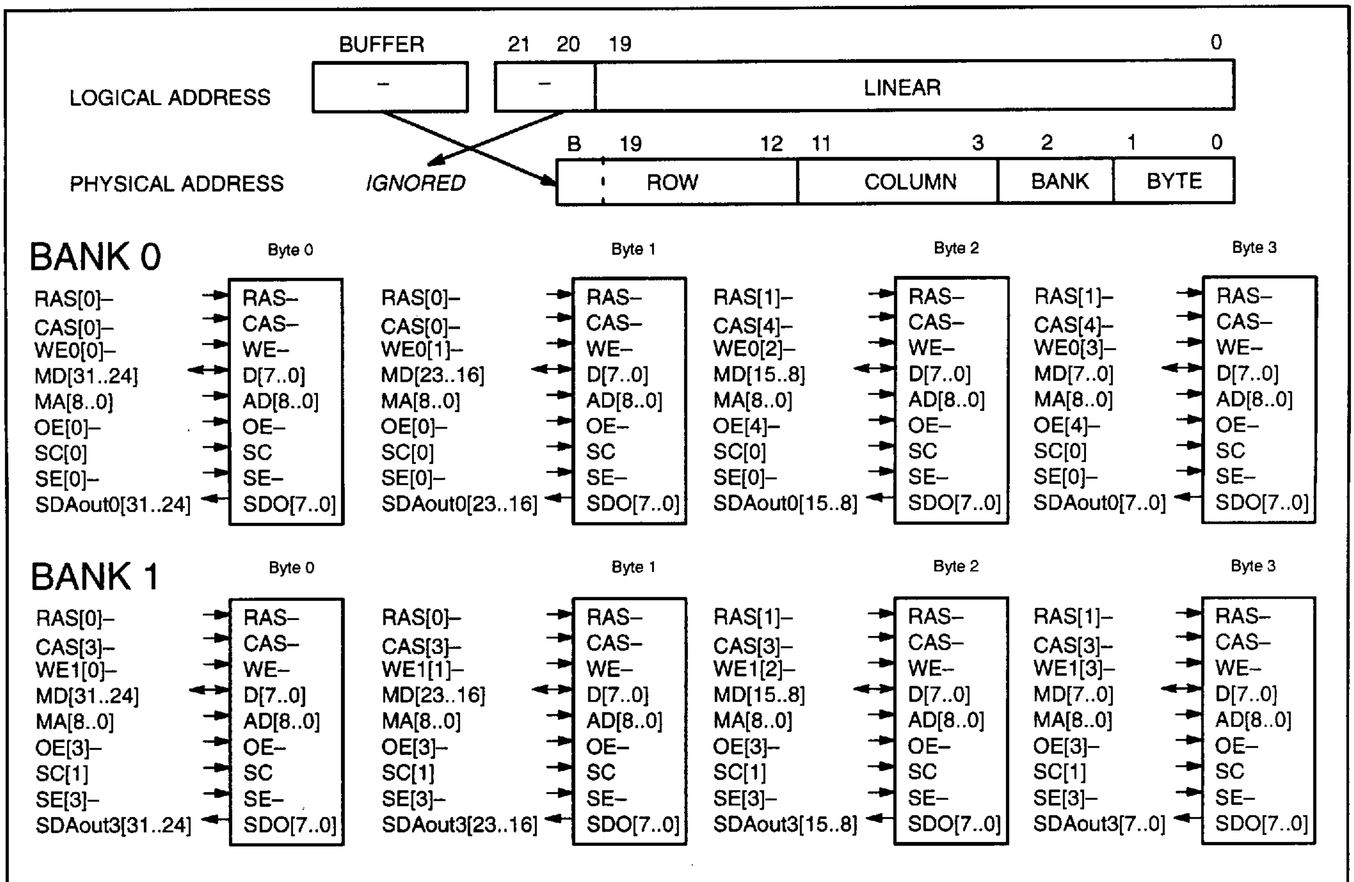


Figure 122. Config 14 (mem_config.config = 1110 or 1101) 2 banks of 256K VRAMS, 2 buffers of 1MB

7.1. Frame Buffer Design Notes, continued

Figure 123 assumes that you are using a 64-bit wide RAMDAC. If you are using a 32-bit wide RAMDAC, use external logic to generate additional serial enable (**SE–**) signals to address banks 1 and 2. Refer to the *Power 9100/Video Power Board Application Note* for more information.

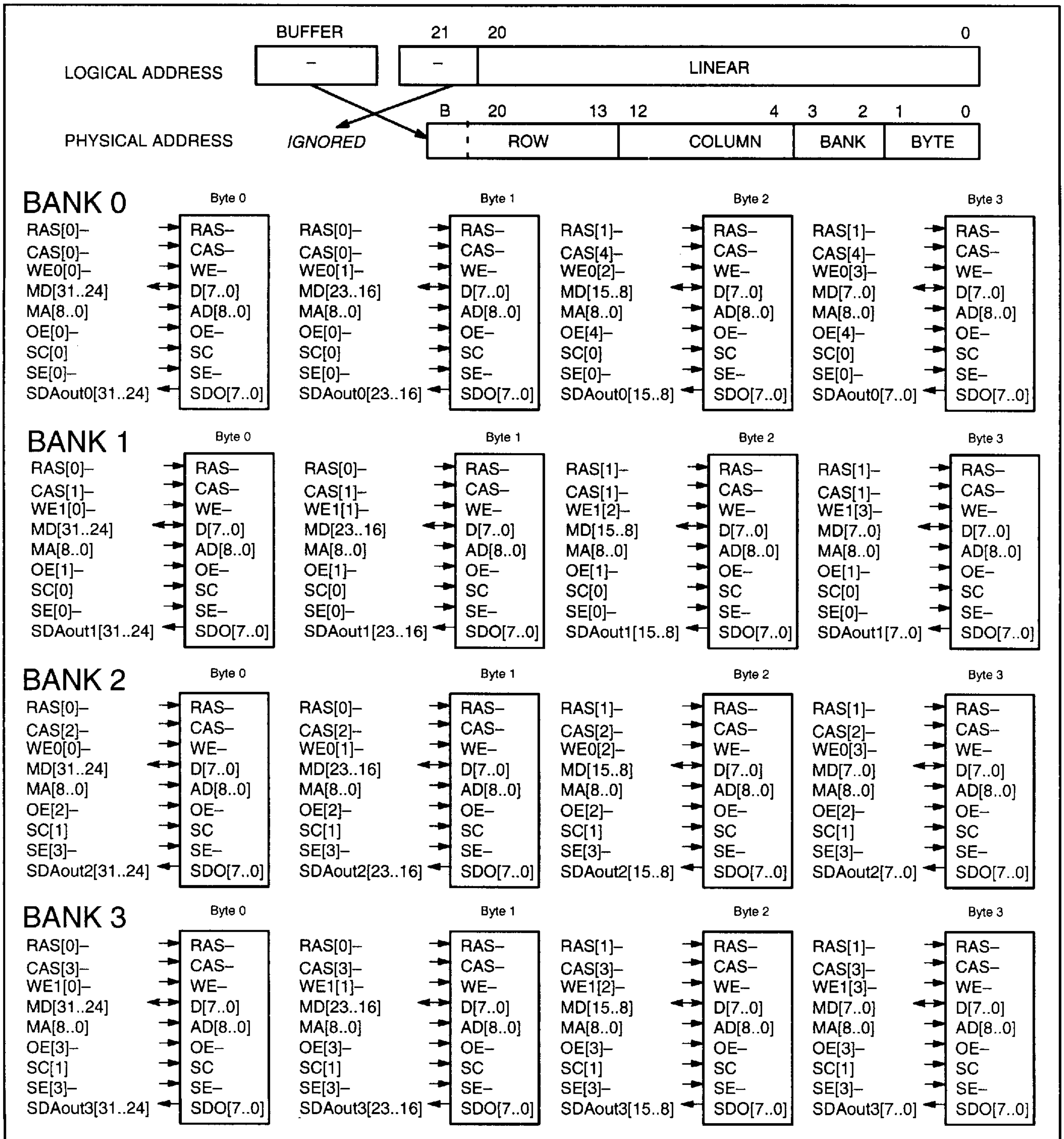


Figure 123. Config 15 (mem_config.config = 1111) 4 banks of 256K VRAMS, 2 buffers of 2MB

7.2. VRAM Access

All VRAM accesses can be described as sequences of the VRAM timing templates shown in figures 124 through 132. The most common adjustments are:

vram_miss_adj = 1

vram_read_adj = 1

vram_read_sample = 1

vram_write_sample = 1

See section 4.7.1, *Memory Configuration Register.*

7.2.1. ROW MISS

Normally the Power 9100 uses the fast page mode method to access the VRAMS. Whenever a new access is to a different page of VRAM than the previous access, the row miss timing sequence is inserted to switch the rams (both banks) to the new bank. Figures 124 and 125 shows this sequence. Note that a write mask is always loaded into the VRAMS. If the programmer has disabled the plane mask feature the Power 9100 automatically generates the all-planes-enabled mask.

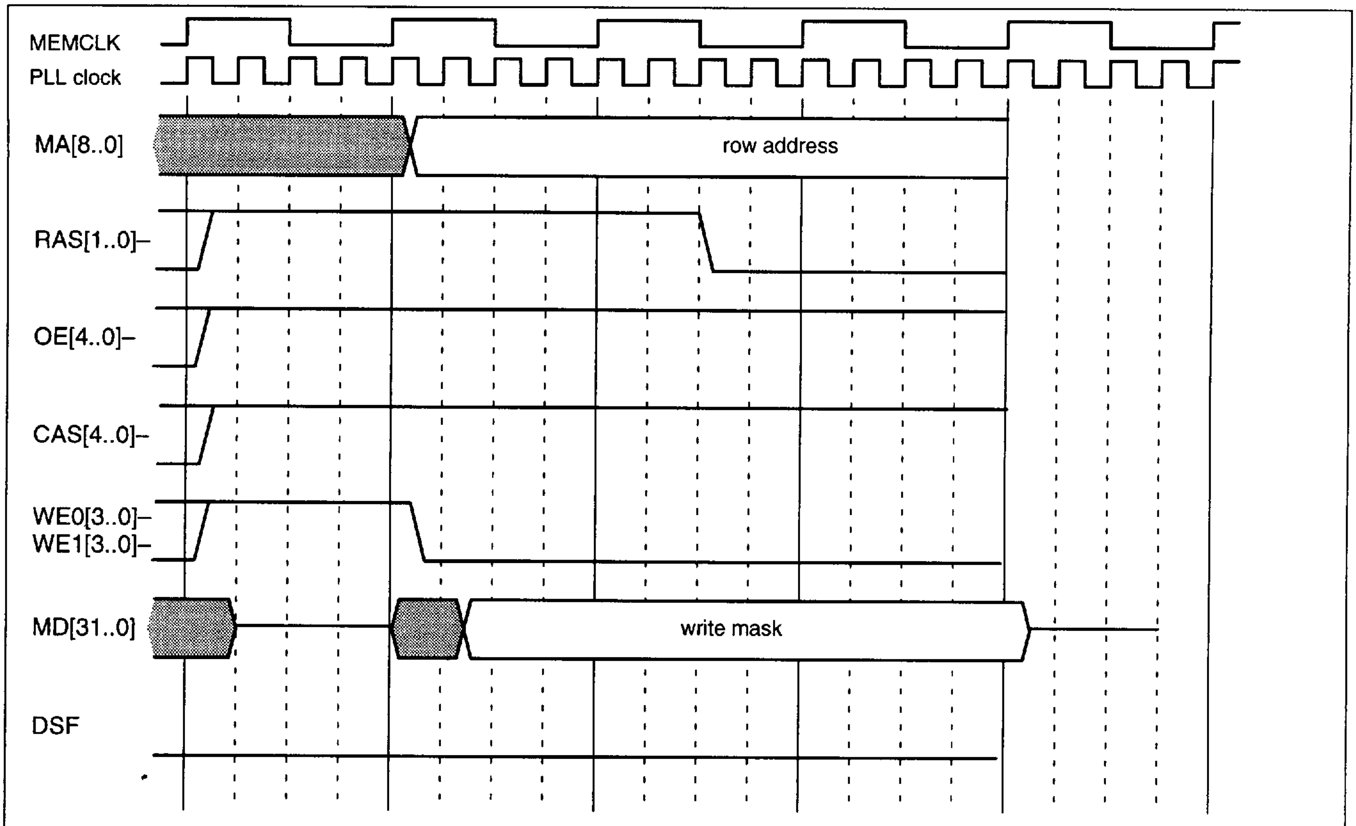


Figure 124. Row miss cycle (mem_config.vram_miss_adj = 0)

7.2. VRAM Access, continued

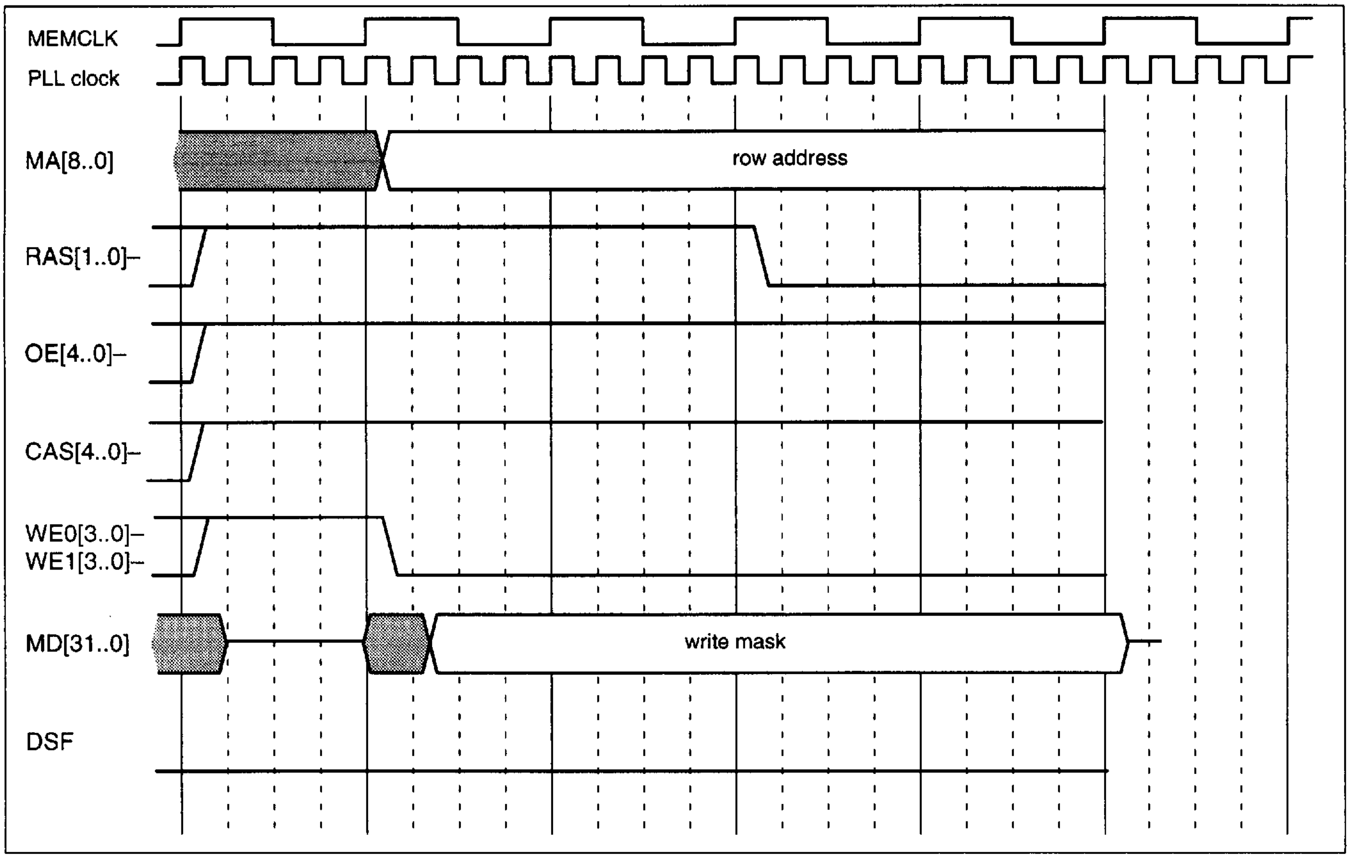


Figure 125. Row miss cycle (mem_config.vram_miss_adj = 1)

7.2. VRAM Access, continued

7.2.2. READ

Read operations take two cycles, which allows for the slow turn off time of VRAMS. Figure 126 shows a read operation from bank 0.

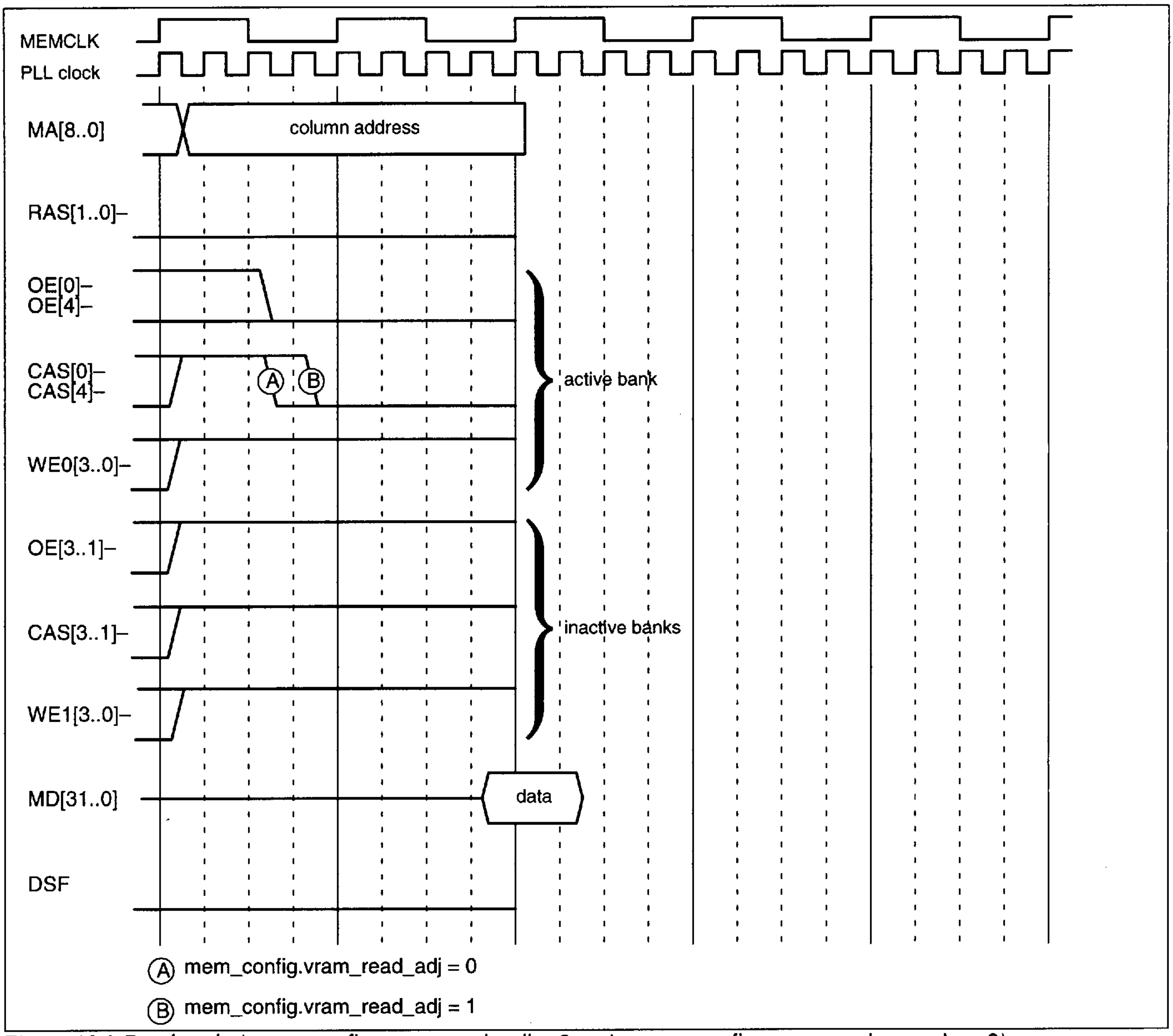


Figure 126. Read cycle (mem_config.vram_read_adj = 0 or 1, mem_config.vram_read_sample = 0)

7.2. VRAM Access, continued

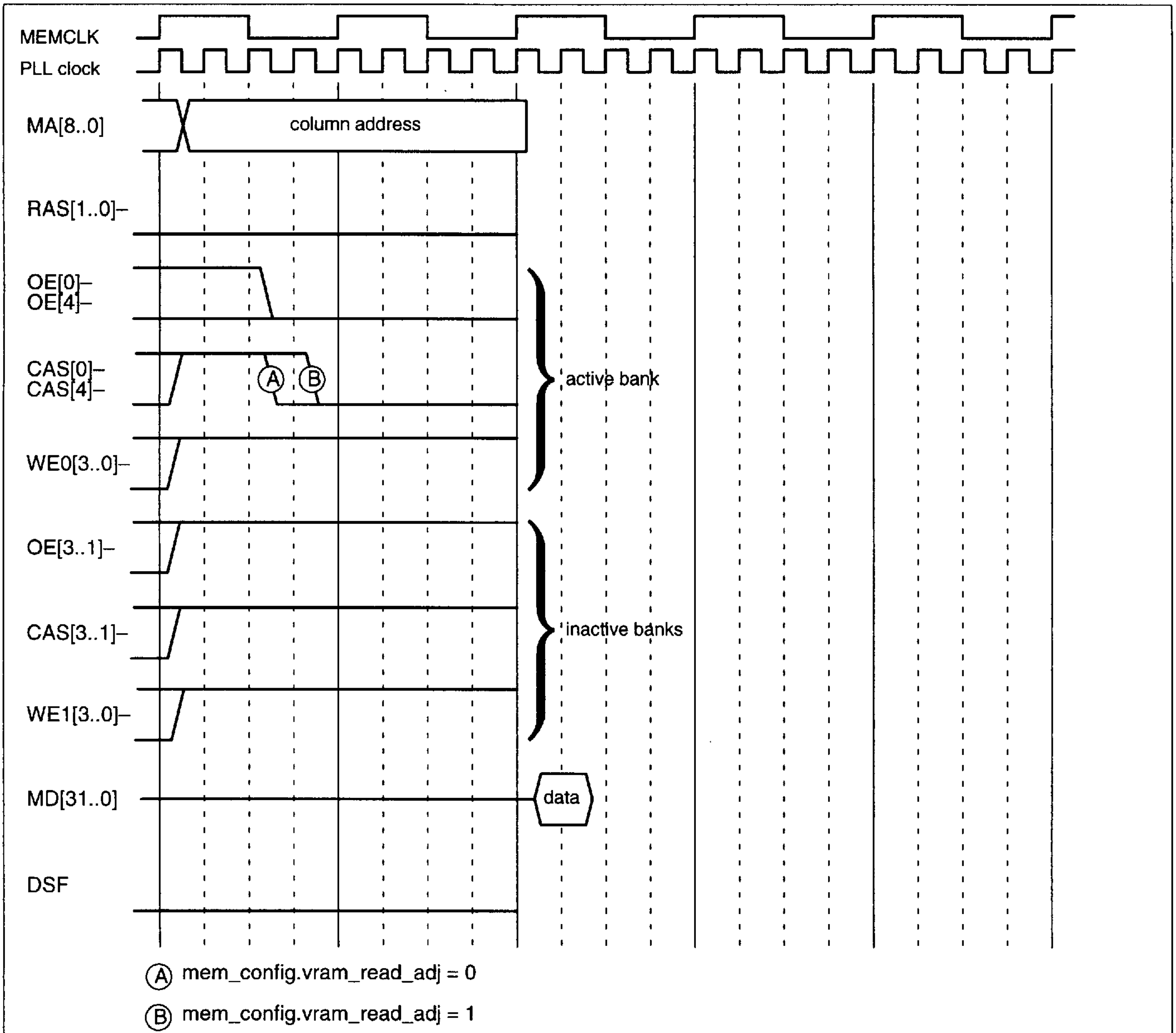


Figure 127. Read cycle (mem_config.vram_read_adj = 0 or 1, mem_config.vram_read_sample = 1)

7.2. VRAM Access, continued

7.2.3. WRITE

A write operation requires either one or two cycles. There are two basic write templates, shown in figure 128. All write operations start with the single cycle that is marked "write" in the template. If the next cycle is *not* a write to the other bank of VRAMs, then the write operation is extended to two cycles by inserting the extra cycle labelled "finish write".

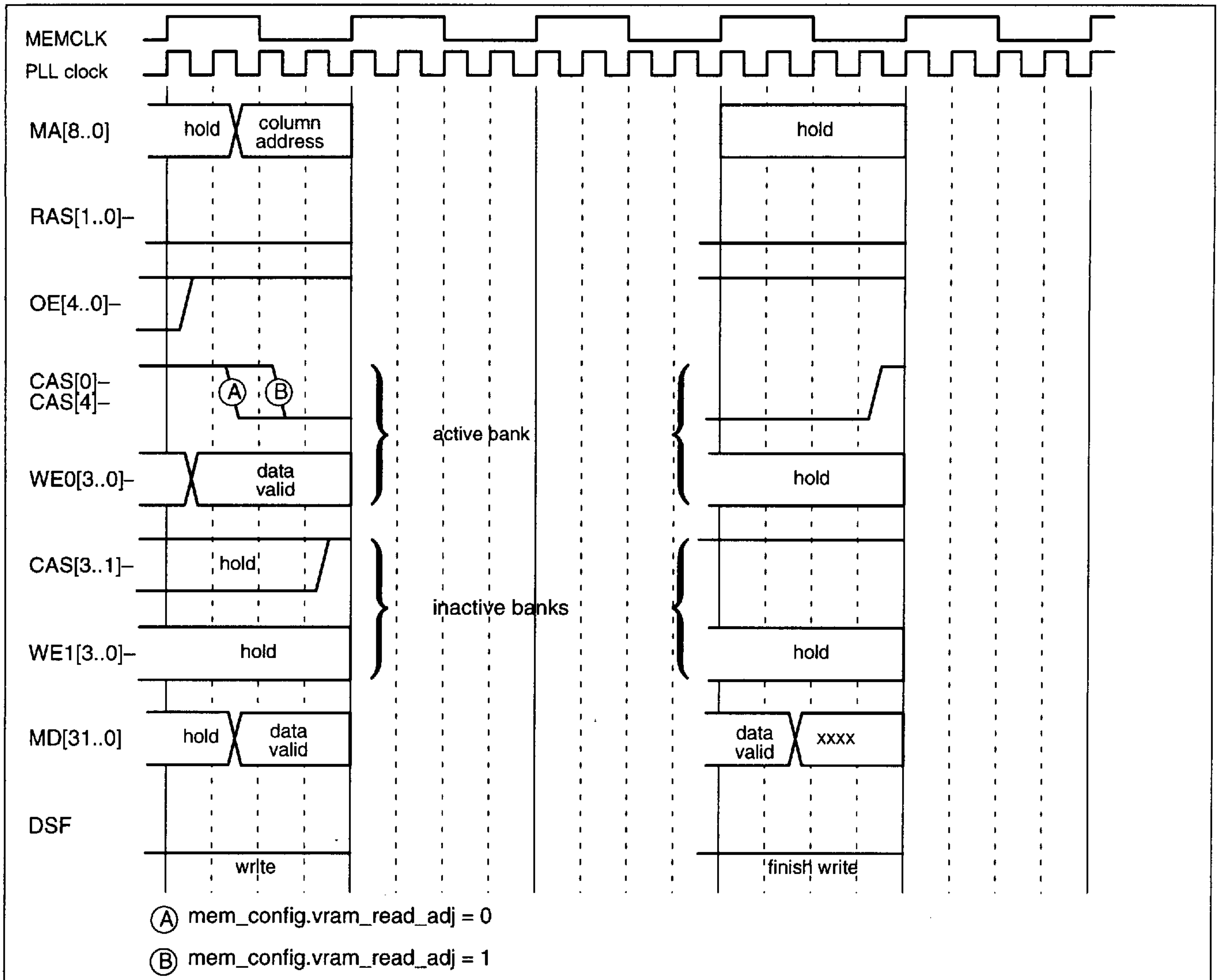


Figure 128. Write cycle (mem_config.vram_write_adj = 0 or 1)

7.2. VRAM Access, continued

7.2.4. READ TRANSFER

Figure 129 shows the template used to reload the internal VRAM video shift register. During a vertical retrace the Power 9100 will reload the entire shift register inside of the VRAMs. Thereafter the Power 9100 will generate a split shift register reload whenever the VRAMs are more than half empty (Based upon the internally generated QSF signal). Because the Power 9100 does not intervene between scan lines (i.e., during horizontal retrace) this method requires that each scan line of the image be located in sequential linear memory and that the VRAM shift clock be inhibited during horizontal blanking intervals. Therefore the physical scan line width of the monitor must match exactly the logical scan line width loaded into the drawing engine. This means that there are no extra pixels between scan lines in memory.

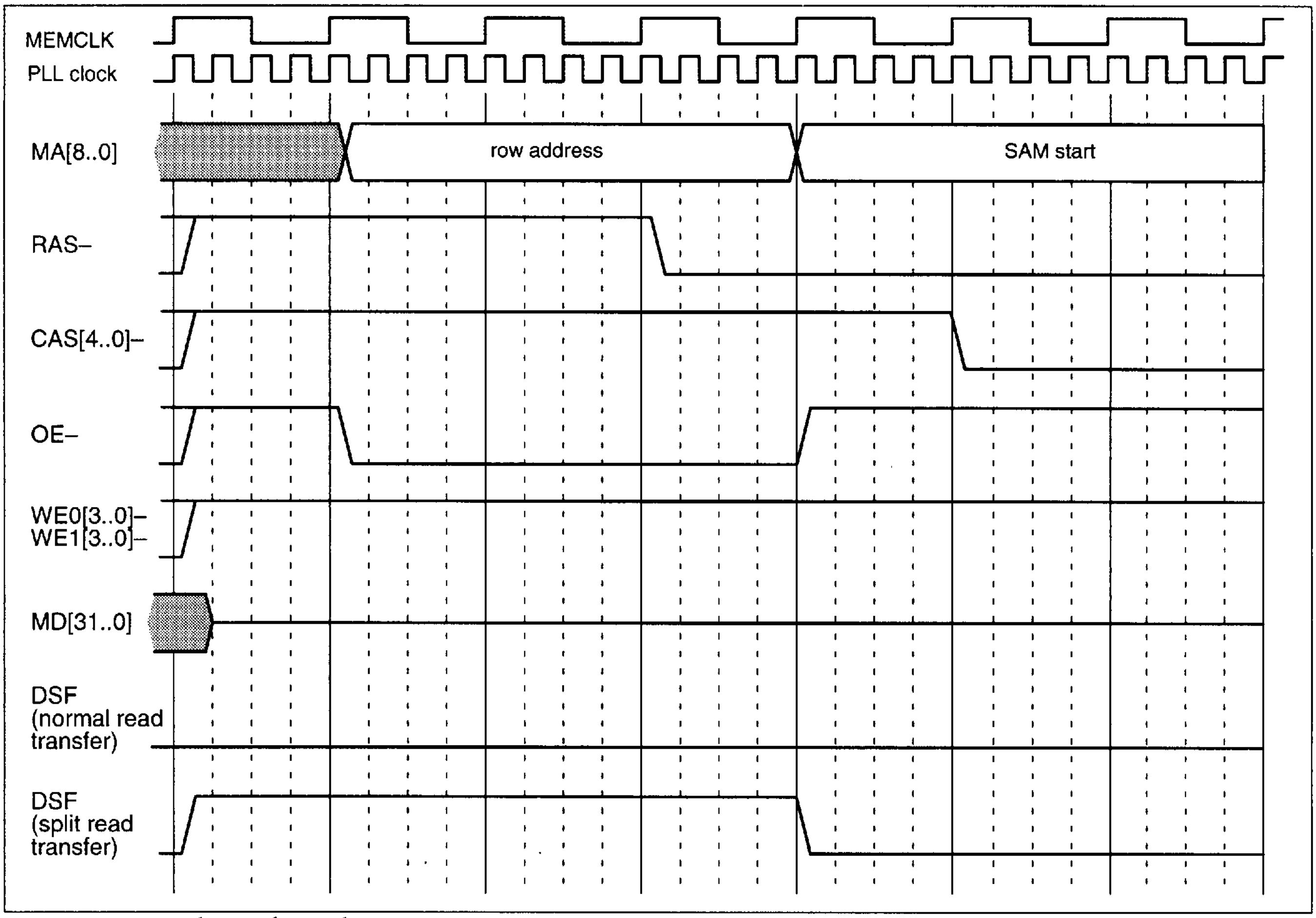


Figure 129. Read transfer cycle

7.2. VRAM Access, continued

7.2.5. REFRESH

Figure 130 shows the template used to refresh the VRAMs. The memory controller has refresh request hold over function which the frame buffer controller may queue up to 4 memory refresh requests in order to reduce the interruption to the video processor.

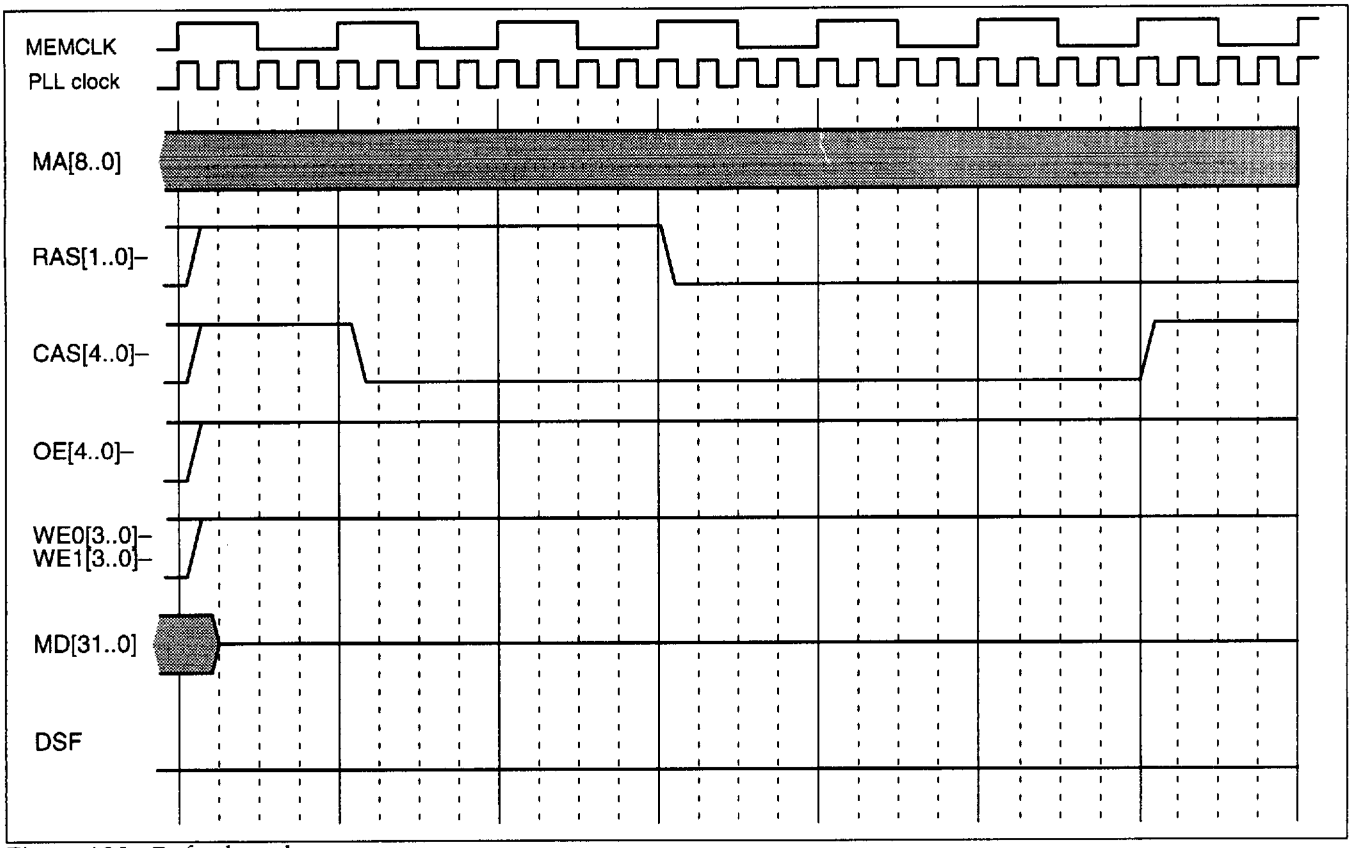


Figure 130. Refresh cycle

7.2. VRAM Access, continued

7.2.6. IDLE

Figure 131 shows the state of the VRAM controls when no RAM activity is required.

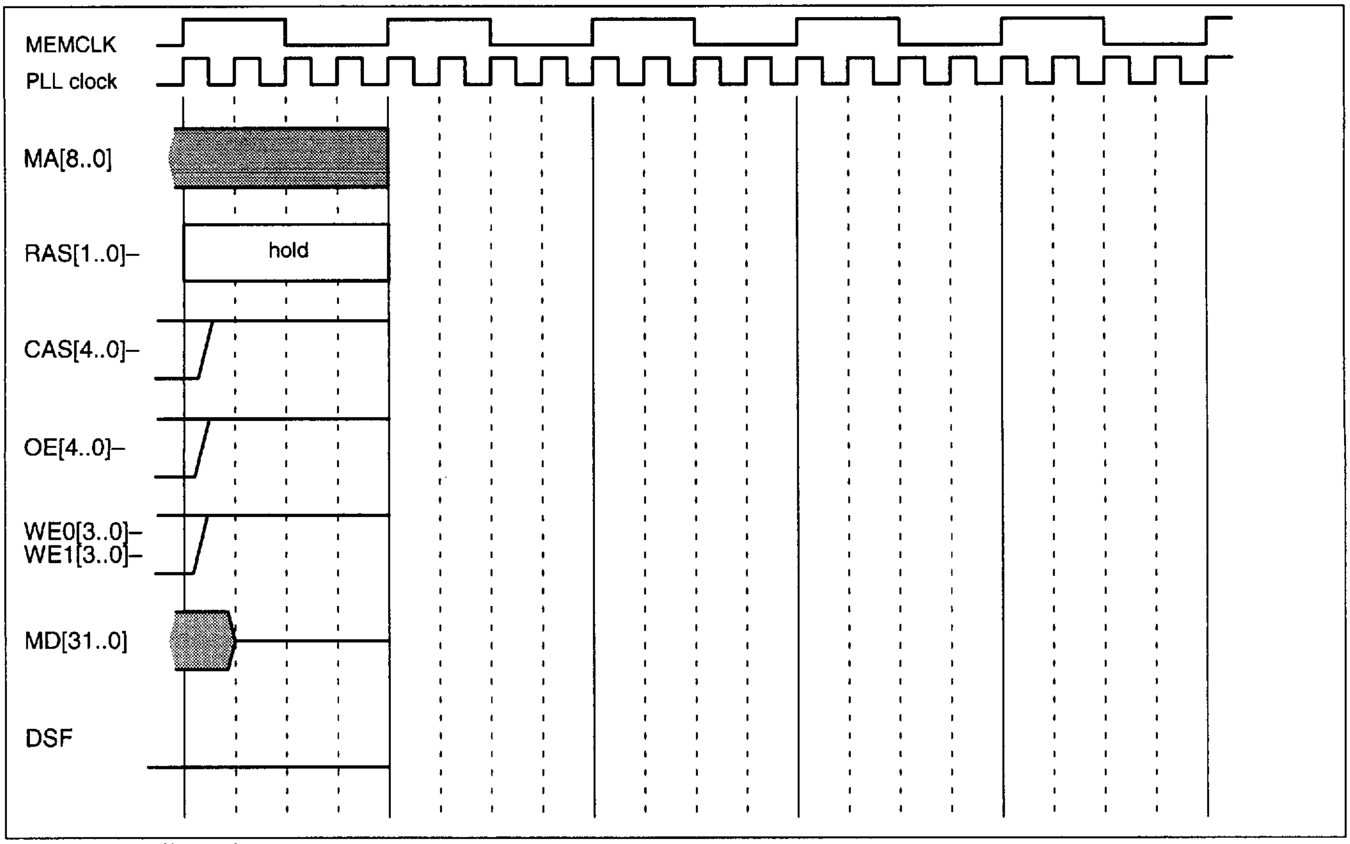


Figure 131. Idle cycle

7.2. VRAM Access, continued

7.2.7. RESET STATE

Figure 132 shows the reset state of the VRAM controls during and after power on reset, synchronous reset, and asynchronous reset.

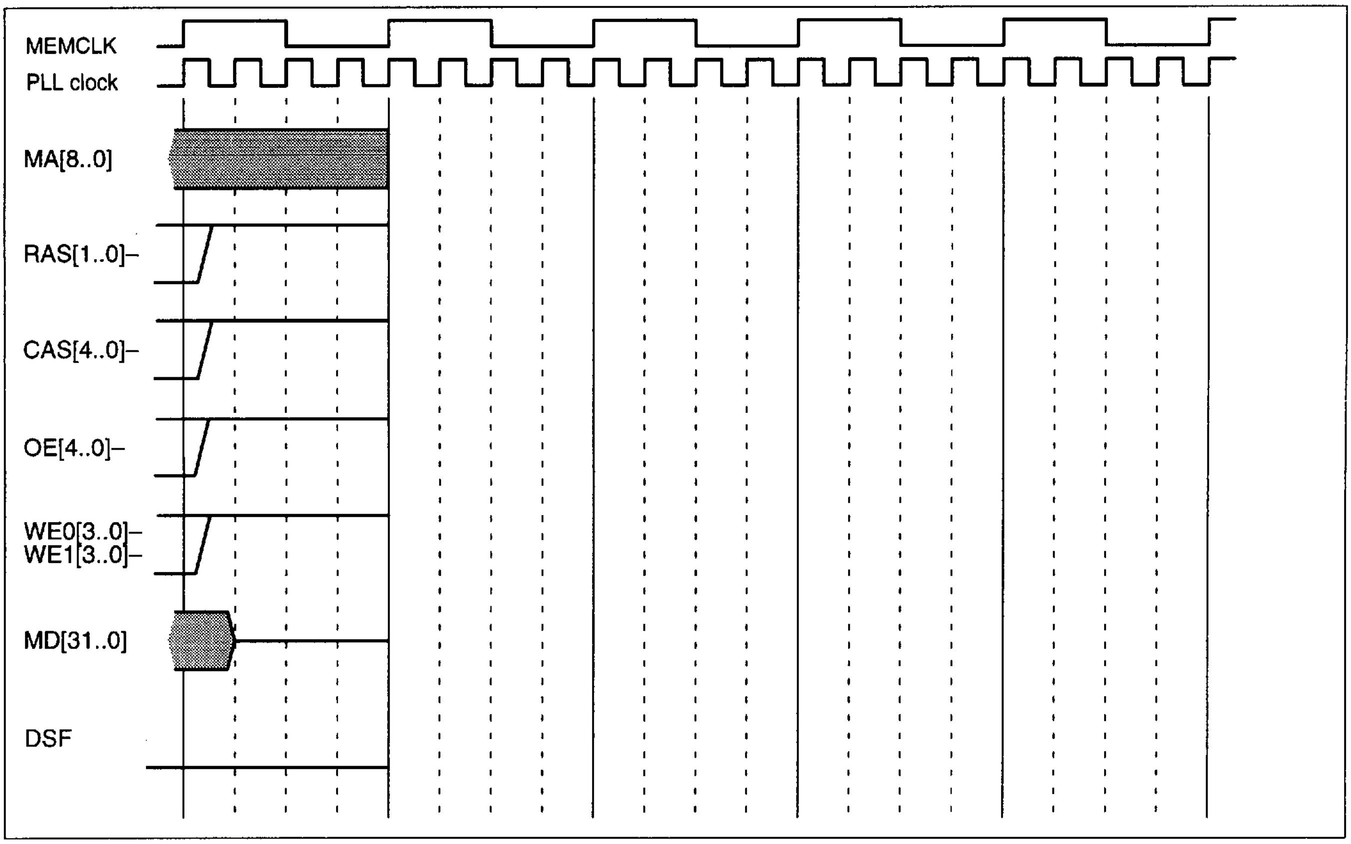


Figure 132. Reset state

Chapter 8. Video and RAMDAC Interface

8.1. Video Control

This section contains information about video control issues. The topics covered are:

- Video Clock generation
- Video Shift Clock generation
- Video Serial Enable generation
- Video Address generation
- Video signals
- Video control registers
- Video timing
- External synchronization issues
- Screen repaint control issues

8.1.1. VIDEO CLOCK GENERATION

Figure 133 shows the different clock generation circuits for the video section. Three clocks are generated: CRTC_CLK, QSF_CLK, and SHIFT_CLK. These clocks control different sections of the video display logic.

See also section 4.6.

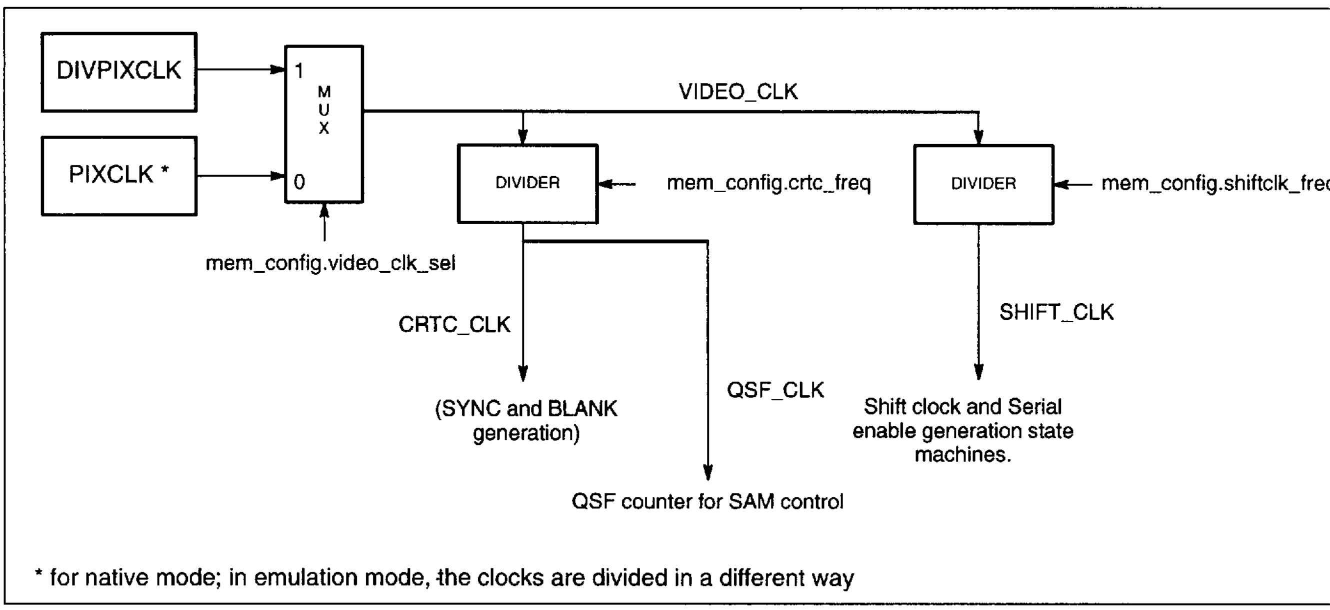


Figure 133. Video clock generation for native mode

8.1. Video Control, continued

8.1.2. VIDEO SHIFT CLOCK GENERATION

Figure 134 shows the different clock generation patterns for different values of mem_config.shiftclk_mode and mem_config.vad_sht.

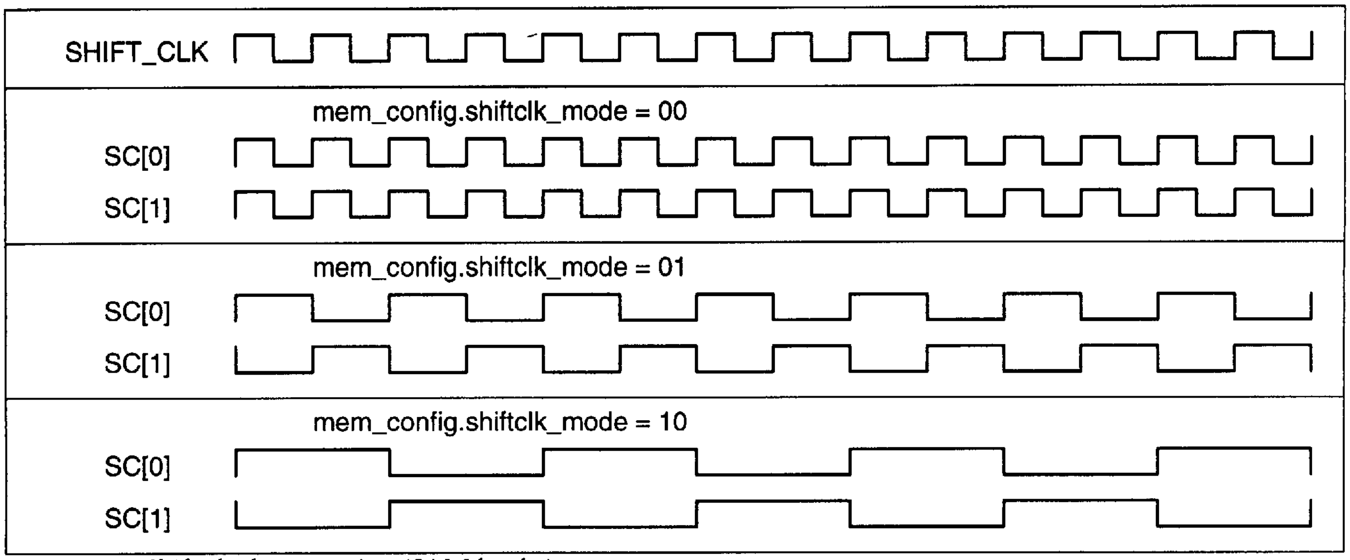
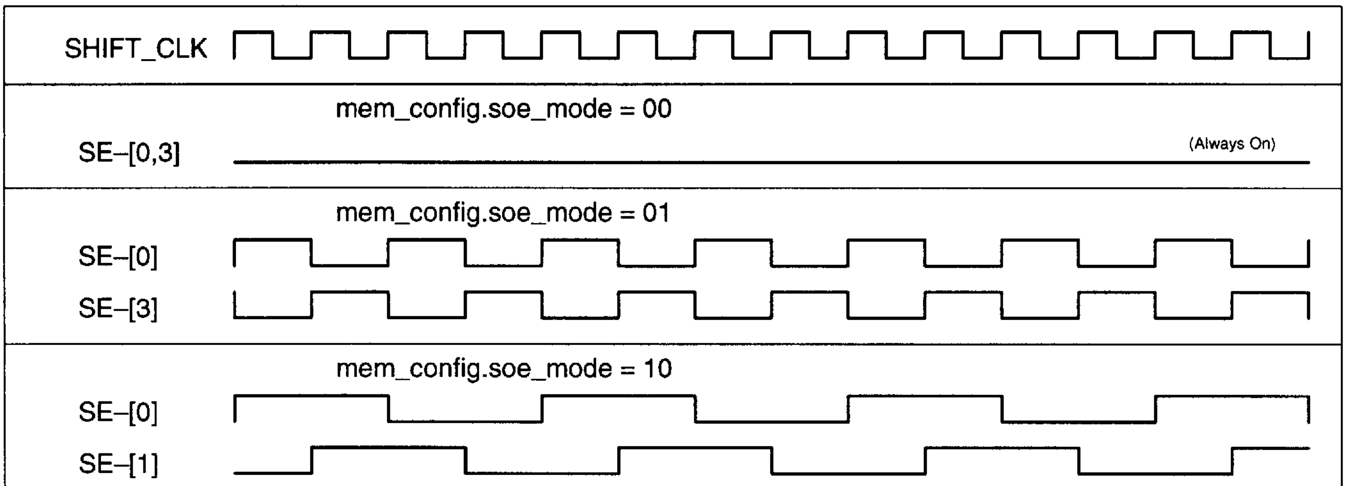


Figure 134. Shift clock generation (SAM banks)

8.1. Video Control, continued

8.1.3. VIDEO SERIAL ENABLE GENERATION

Figure 135 shows the different serial enable generation patterns for different values of mem_config.soe_mode and mem_config.vad_sht.



Use this mode in conjunction with external logic to generate serial enables (SE1 and SE2) for a four-bank system with a 32-bit wide RAMDAC. Refer to the *Power 9100/Video Power Board Application Note* for more information.

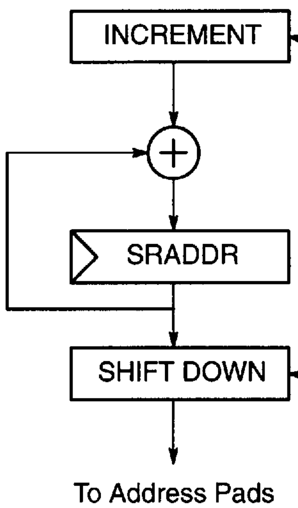
Figure 135. Serial enable generation (SAM banks)

8.1. Video Control, continued

8.1.4. VIDEO ADDRESS GENERATION

Figure 136 shows how the VRAM shift register reload address is controlled. Since the shiftdown at the bottom of the data path is controlled in part by the memory configuration, the amount programmed into the srtctl.src_incs field must be computed to correct for this shiftdown.

Figure 137 shows the most commonly used settings. Refer also to the mem_config register (figure 91), the srtctl register (figure 87) and to section 7.1, *Frame Buffer Design Notes.*



srtctl.src_incs	Transfer Type	Increment Amount
00	SPLIT	256
00	FULL	512
01	SPLIT	512
01	FULL	1024
10	SPLIT	1024
10	FULL	2048

mem_config.config	mem_config.vad_sht	Shift Down (bits)
–00	0	0
–00	1	1
–01,–10	0	1
–01,–10	1	2
–11	0	2
–11	1	–

Figure 136. Video address generation

Number of Banks	Depth of Memory	SAM Size	mem_config.config[1..0]	srtctl.src_incs	mem_config.vad_sht
1	256K	full	00	00	0
2	256K	full	10 or 10	01	0
4	256K	full	11	10	0
1	256K	half	00	00	1
2	256K	half	01 or 10	00	0
4	256K	half	11	01	0

Figure 137. Video address generation settings

8.1. Video Control, continued

8.1.5. VIDEO SIGNALS AND REGISTERS

A complete video control section resides on the Power 9100 chip. This section provides the synchronization, blanking, and timing signals for the graphics subsystem, as well as the control sections for screen refresh timing and VRAM control.

The Power 9100 provides separate HSYNC, VSYNC, and BLANK– video signals in native mode.

The video control section of the Power 9100 is fully programmable and is defined by the host system. For example: The video control section can be defined to generate an internal sync signal or accept an external signal to control screen refresh, thereby simplifying the task of merging Power 9100-created graphics with other video images.

The external pins are controlled by srtctl2.vsync_plt and srtctl2.hsync_plt. These controls allow the selection of external signal polarity and the support of VESA monitor power down modes. See figure 88.

For easy reference, figure 138 repeats the video control register summary presented in chapter 4.

8.1. Video Control, continued

Register	Description/Function
HORIZONTAL TIMING	
hrzc	Horizontal counter. Read only. Specifies the current pixel position along a horizontal sweep; the Power 9100 increments this counter upon each occurrence of CRTC_CLK. It resets to zero at the assertion of HSYNC. The value occupies the lower 12 bits of the register, which is set by the Power 9100.
hrzt	Horizontal length. Read/write. Specifies the length of a horizontal scan line. The value occupies the lower 12 bits of the register, which is set by the host. The Power 9100 compares the current hrzc value (the current pixel position) to this value to determine when to wrap around. The hrzt register must be a multiple of 8 for all modes except 8-bit-per-pixel with a 64-bit wide RAMDAC, when it must be a multiple of 16. For simplicity, hrzt may be a multiple of 16 for all modes.
hrzsr	Horizontal sync assertion. Read/write. Specifies the the amount of time in CRC_CLKS, the horizontal sync is active. The value occupies the lower 12 bits of the register, which is set by the host.
hrzbr	Horizontal blank deassertion. Read/write. Specifies the location along a horizontal sweep which defines the deassertion of the horizontal blank. The value occupies the lower 12 bits of the register, which is set by the host.
hrzbf	Horizontal blank assertion. Read/write. Specifies the location along a horizontal sweep which defines the assertion of the horizontal blank. The value occupies the lower 12 bits of the register, which is set by the host.
prehrzc	Horizontal counter preload value. Read/write. Specifies the value with which to preload hrzc upon receipt of an internal or external HSYNC or an external VSYNC; allows synchronization of the Power 9100 with external video sources, whatever the combination of internal or external delays. The value occupies the lower 12 bits of the register, which is set by the host. Set this register to zero when using only internal syncs.
VERTICAL TIMING	
vrtc	Vertical counter. Read only. Specifies the current line position along a vertical sweep; the Power 9100 increments this counter upon each occurrence of HSYNC. The value occupies the lower 12 bits of the register, which is set by the Power 9100. In the vertical timing sequence, vrtt defines the number of horizontal lines from the assertion of VSYNC to the deassertion of the next VSYNC (the number of horizontal lines in a complete vertical scan cycle). It resets to zero at the assertion of VSYNC.
vrtt	Vertical length. Read/write. Specifies the number of lines in a vertical sweep. The value occupies the lower 12 bits of the register, which is set by the host. The vrtc register counts from zero to vrtt–1 and then repeats.
vrtsr	Vertical sync assertion. Read/write. Specifies the amount of time, in horizontal scan lines, the vertical sync is active. The value occupies the lower 12 bits of the register, which is set by the host.
vrtbr	Vertical blank deassertion. Read/write. Specifies the location along a vertical sweep which defines the vertical blank deassertion. The value occupies the lower 12 bits of the register, which is set by the host.
vrtbf	Vertical blank assertion. Read/write. Specifies the location along a vertical sweep which defines the vertical blank assertion. The value occupies the lower 12 bits of the register, which is set by the host.
prevrtc	Vertical counter preload value. Read/write. Specifies the value with which to preload vrtc upon receipt of an internal or external VSYNC; allows synchronization of the Power 9100 with external video sources, whatever the combination of internal or external delays. The value occupies the lower 12 bits of the register, which is set by the host. Set this register to zero when using only internal syncs.

Figure 138. Video control registers

8.1. Video Control, continued

8.1.6. VIDEO TIMING

The video signals and registers work together to drive the signals that control the monitor. Remember, all horizontal counts are in units of CRTC_CLK (see figure 133) and vertical counts are in units of scan lines.

HORIZONTAL VIDEO TIMING

Figure 139 presents the timing sequence for horizontal video control . All of the horizontal timing registers are loaded with counts derived from CRTC_CLK which may represent multiple pixels.

Figure 138 describes the functions of the horizontal timing registers.

The programmer must satisfy the following condition:

hrzsr < hrzbr < hrzbf < hrzt

VERTICAL VIDEO TIMING

Figure 140 presents the timing diagram for vertical video timing control. All of the vertical timing registers are loaded with counts that represent horizontal scan lines. Also, each number loaded into a vertical timing register represents the total count in the count sequence.

Figure 138 describes the functions of the vertical timing registers.

The programmer must satisfy the following condition:

vrtsr < vrtbr < vrtbf < vrtt

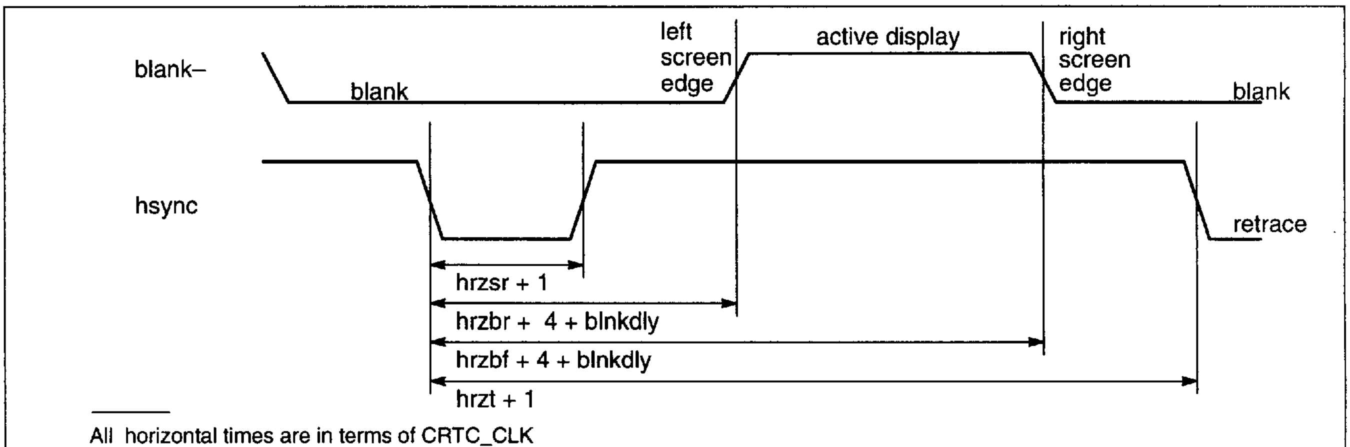


Figure 139. Power 9100 horizontal video timing parameters

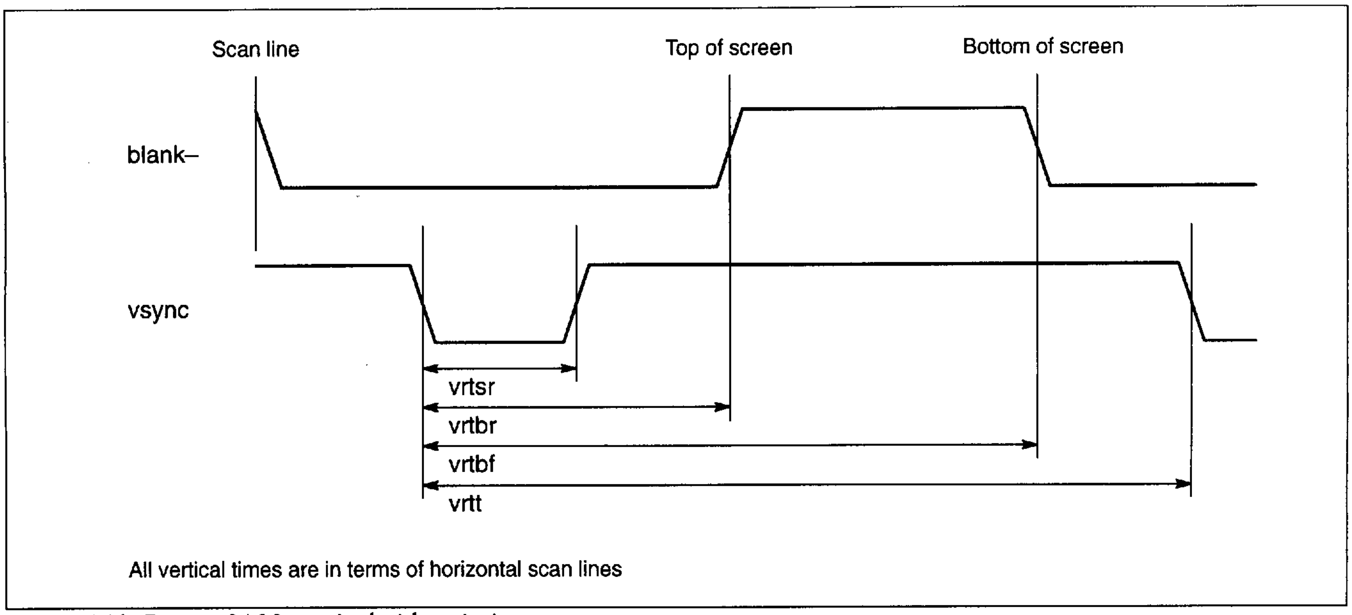


Figure 140. Power 9100 vertical video timing parameters

8.1.7. EXTERNAL SYNCHRONIZATION

The VSYNC and HSYNC pins can be defined as inputs to provide synchronization from an external source. This allows merging of Power 9100 native mode video with video generated by an external source. With values programmed into prehrzc and prevrtc, the Power 9100 can synchronize exactly with the external source, regardless of internal and external delays in the system.

There are two standard modes for implementation of external sync timing: external VSYNC only and external VSYNC and HSYNC.

With external VSYNC only, the Power 9100 preloads both the horizontal and vertical counters (hrzc and vrtc) on the falling edge of an external VSYNC pulse. This synchronizes both HSYNC and VSYNC to the external source.

With external VSYNC and HSYNC, the Power 9100 preloads the hrzc counter on the falling edge of an external HSYNC and preloads the vrtc counter on the falling edge of an external VSYNC.

HSYNC and VSYNC input pulses must be at least one DIVPIXCLK wide, but must also be shorter than the horizontal sync rising edge defined in the hrzsr register. This applies to native mode only.

8.1.8. SCREEN REPAINT CONTROL

Screen repaint is a generic descriptor for the functions used by the frame buffer memory to provide pixel information to the "back-end" of the video subsystem to display stored information on the monitor. The Power 9100 provides two basic methods to ensure that the shift registers of the VRAMs always contain the correct data to be shifted onto the screen: normal or split shift mode and restricted or HBLNK– reload mode. These two methods are controlled by the setting of the hblnk_reload bit in the srtctl register (see figure 87).

NORMAL (SPLIT SHIFT MODE)

For normal operation, the hblnk_reload bit in the srtctl register is zero. At the end of the display, the Power 9100 reloads the shift register inside the VRAMs. It performs a split shift register reload when the VRAM output shift registers become half empty, based on the internally generated QSF signal. The QSF signal counters are software programmable.

Normal split shift mode requires that each scan line of the image be located in sequential linear memory. The shift clock is inhibited during BLANK intervals.

The physical scan line width of the monitor must exactly match the logical scan line width loaded into the drawing engine. This requires that there be no extra pixels between scan lines in memory. After VBLINK, the Power 9100 performs a full reload of VRAM addresses; subsequent reloads are split reloads.

8.1. Video Control, continued

RESTRICTED (HBLNK- RELOAD MODE)

For restricted operation, the hblnk_reload bit in the srtctl register is one. The entire VRAM shift register is reloaded for every scan line (HBLNK asserted). In this mode, every scan line in the VRAM must be entirely contained within a single shift register. This constrains scan line length to be less than one whole, one half, or one quarter of the entire shift register length. When using this mode, the available screen resolutions are restricted, as defined in figure 141.

VIRTUAL SCREEN SIZES 8bpp Values (Restricted Mode) (Divide by 2 for 15/16bpp, divide by 4 for 32bpp)						
men_config.config	Maximum Screen Resolution		Scan Increment	Double Buffering	Memory	Banks
	Horizontal	Vertical				
0001, 0010, 0100	1024	1024	1/2 row	No	1M	1 bank, 256K 2 banks, 128K
0011, 0101, 0110	2048	1024	1/2 row	No	2M	4 banks, 128K 2 banks, 256K
1011, 1110, 1101	1024	1024	1/4 row	Yes	2M	2 banks, 256K 4 banks, 128K
1111	2048	1024	1/4 row	Yes	4M	4 banks, 256K
0111	2048	2048	1/4 row	No	4M	4 banks, 256K

Figure 141. Screen resolutions in 8 bpp values, restricted mode (divide by 2 for 15/16bpp, divide by 4 for 32bpp)

8.2. RAMDAC

Figure 142 shows the RAMDAC read cycle. Figure 144 shows the RAMDAC write cycle. Power 9100 designs are expected to utilize a triple 8-bit color with VGA pseudo-color look-up table and pixel unpack RAMDAC (for example, the Brooktree 485). The host has 8-bit data access to the RAMDAC through the shared frame buffer pins. Only the RAMDAC status read register is shadowed in Power 9100.

For VGA mode, the Power 9100 uses standard DAC I/O locations 3CFh, 3C7h, 3C8h, and 3C9h. In this case WE1[3..2]– an d RS[3..2] are always low, and WE1[1..0]– and RS[1..0] are determined by the host bus I/O address.

Setting dac_access_adj = 1 provides a three-dotclk delay between cycles to allow for RAMDAC recovery.

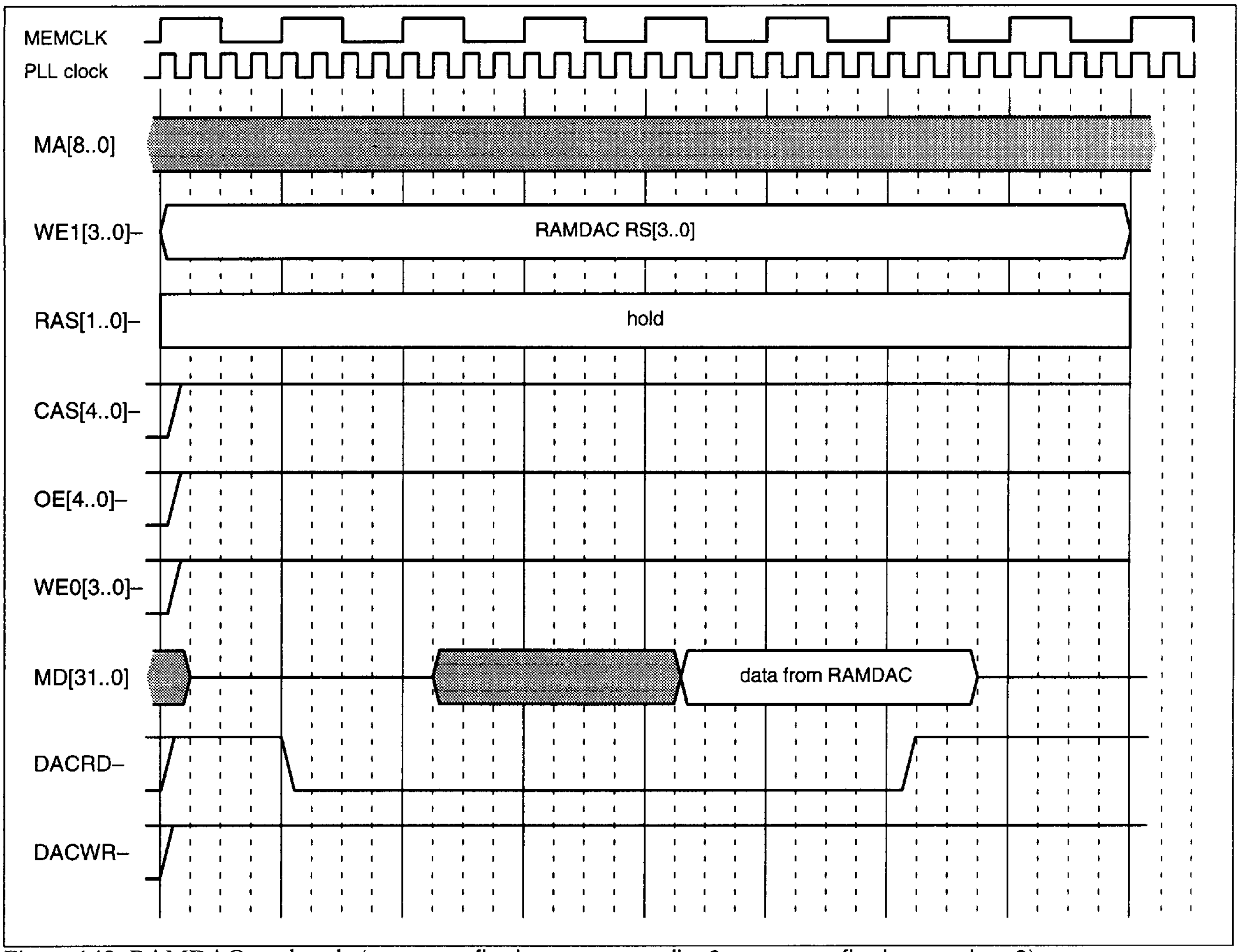


Figure 142. RAMDAC read cycle (mem_config.dac_access_adj = 0, mem_config.dac_mode = 0)

8.2. RAMDAC, continued

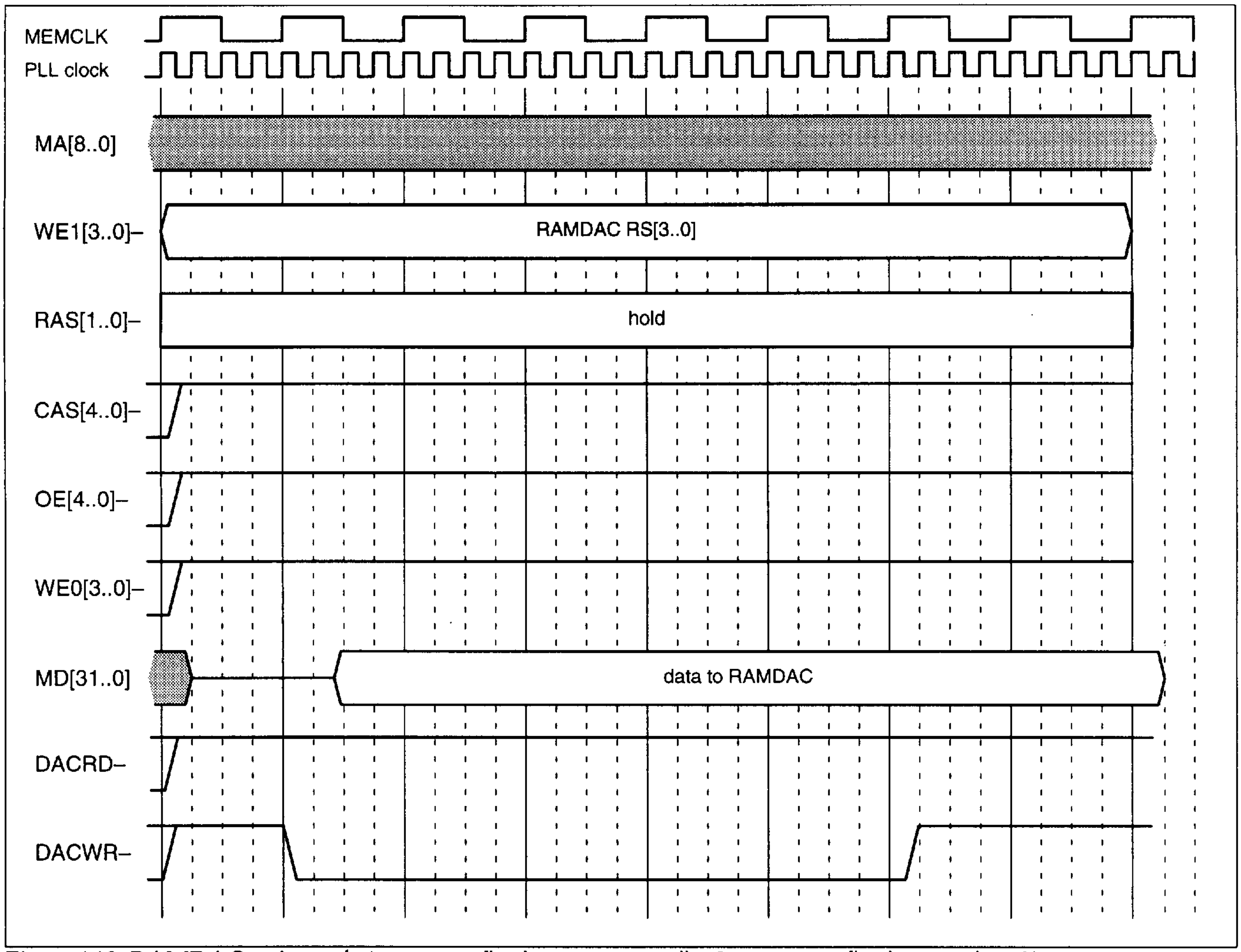


Figure 143. RAMDAC write cycle (mem_config.dac_access_adj = 0, mem_config.dac_mode = 0)

8.2. RAMDAC, continued

Figures 144 and 145 describe a less commonly used RAMDAC protocol where DACRD– is the strobe and DACWR– indicates a write cycle (low, figure 144) or a read cycle (high, figure 145).

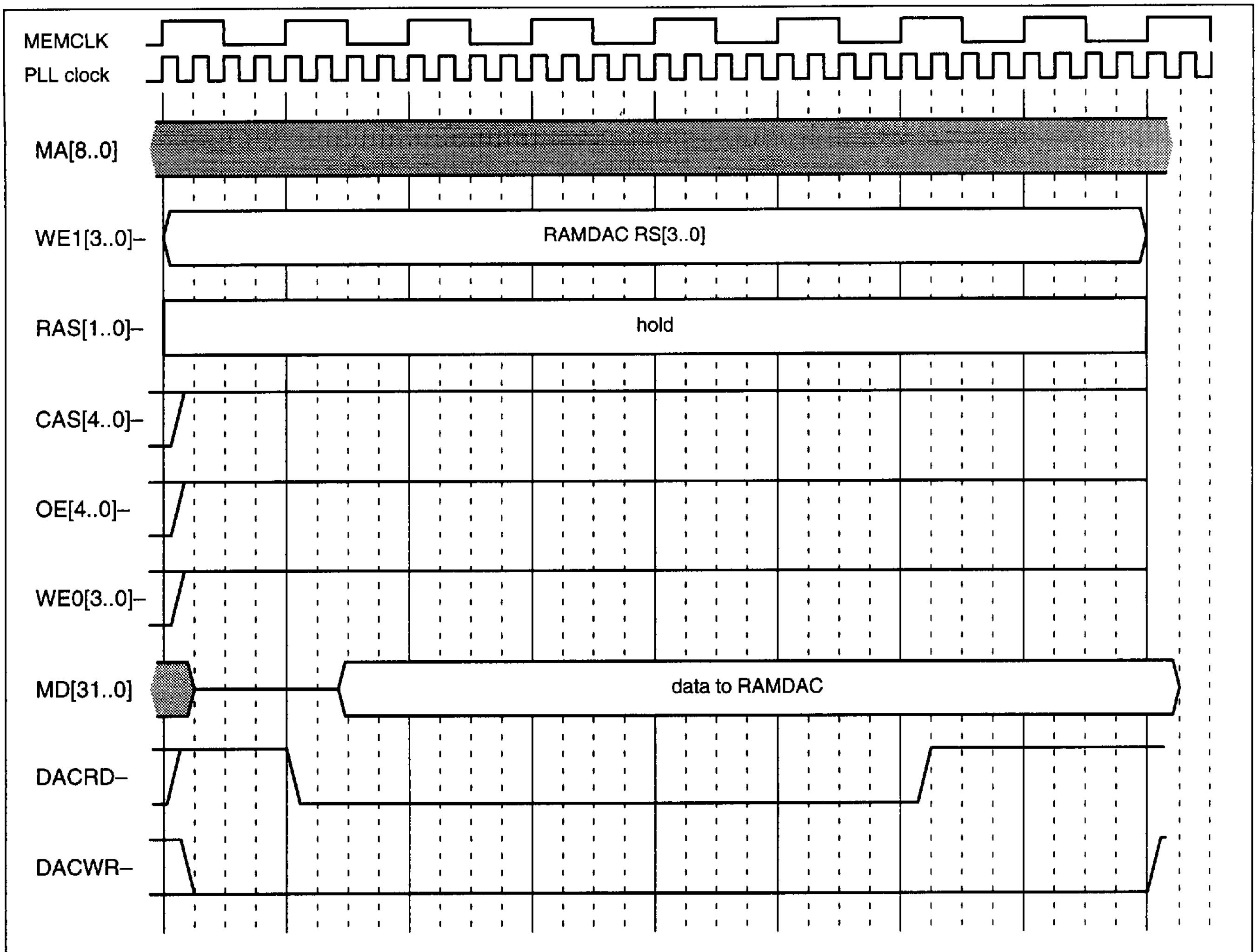


Figure 144. RAMDAC write cycle (mem_config.dac_access_adj = 0, mem_config.dac_mode = 1)

8.2. RAMDAC, continued

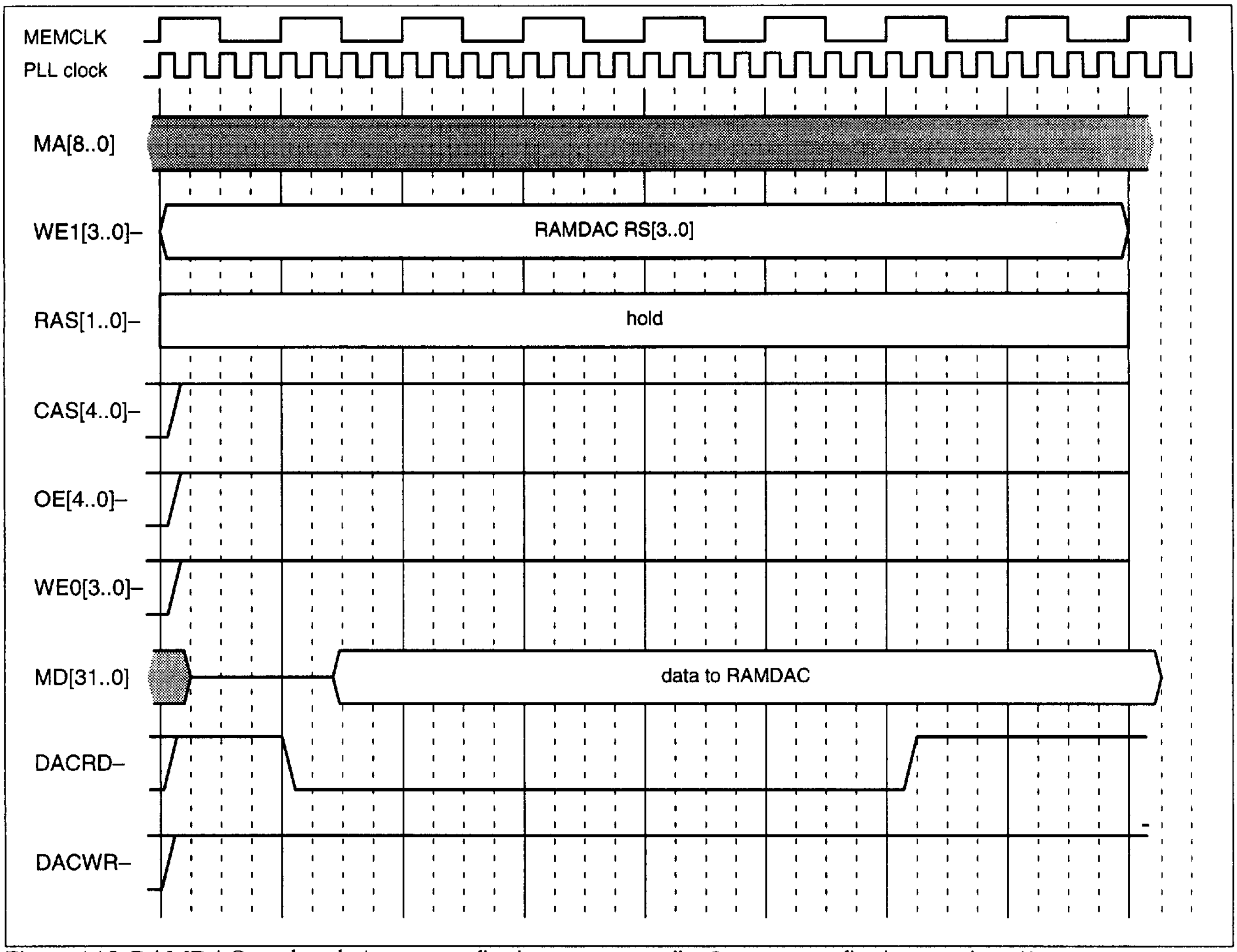


Figure 145. RAMDAC read cycle (mem_config.dac_access_adj = 0, mem_config.dac_mode = 1)

Chapter 9. Coprocessor Interface

The video coprocessor interface allows a separate coprocessor to share the Power 9100 host interface and frame buffer. The first set of timing diagrams show how the host is allowed to read and write control registers of the video coprocessor. The second set of diagrams specifies how the protocol for sharing the frame buffer operates. The coprocessor uses the VCEN– signal to qualify the shared VCGRNT–, VCIOR–, and VCIOW– signals.

9.1. Video Coprocessor I/O Read

This template is invoked to read a 32-bit register from the video coprocessor.

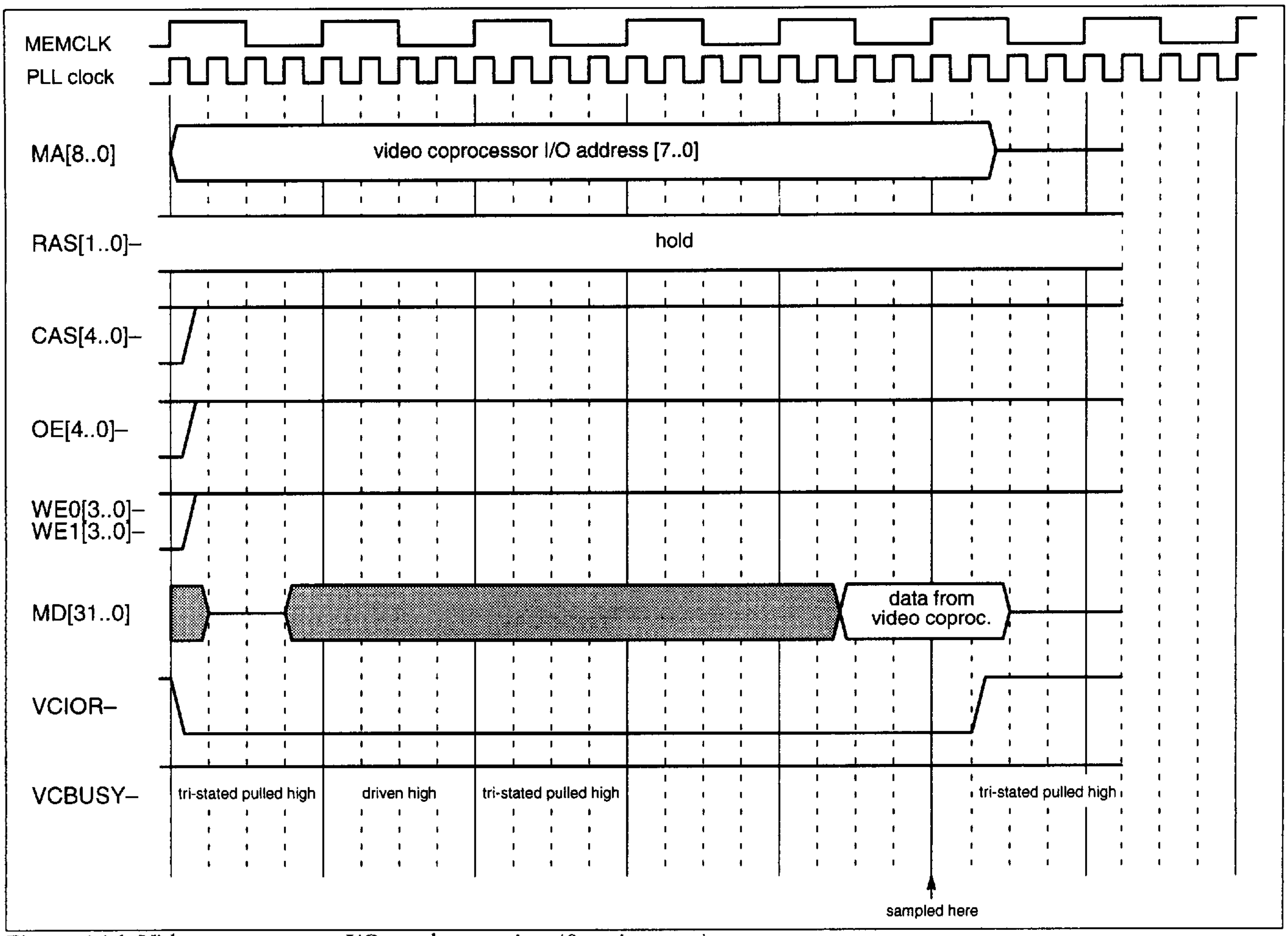


Figure 146. Video coprocessor I/O read operation (0 wait states)

9.1. Video Coprocessor I/O Read, continued

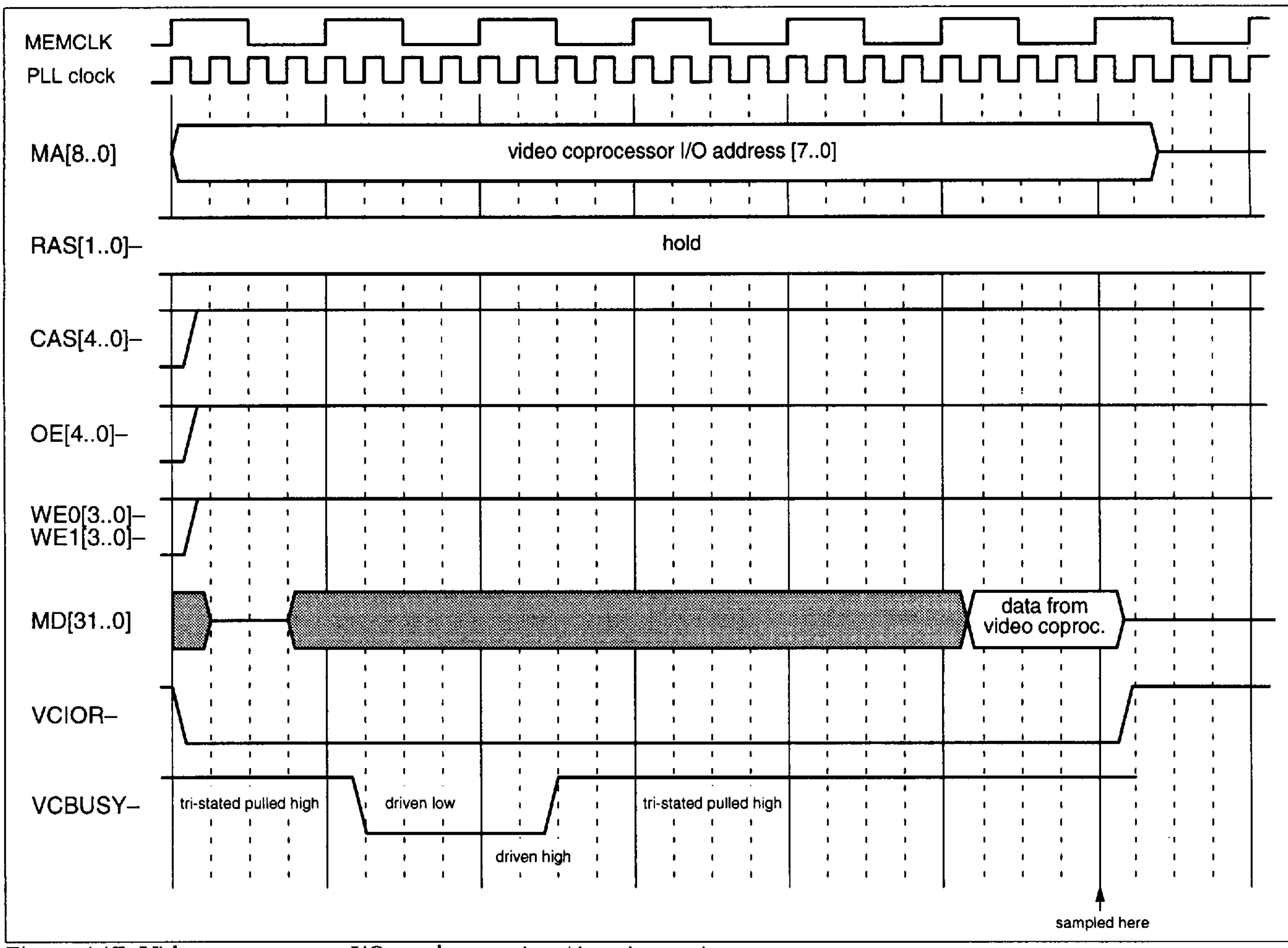


Figure 147. Video coprocessor I/O read operation (1 wait state)

9.2. Video Coprocessor I/O Write

This template is invoked to write a 32-bit value to a register in the video coprocessor.

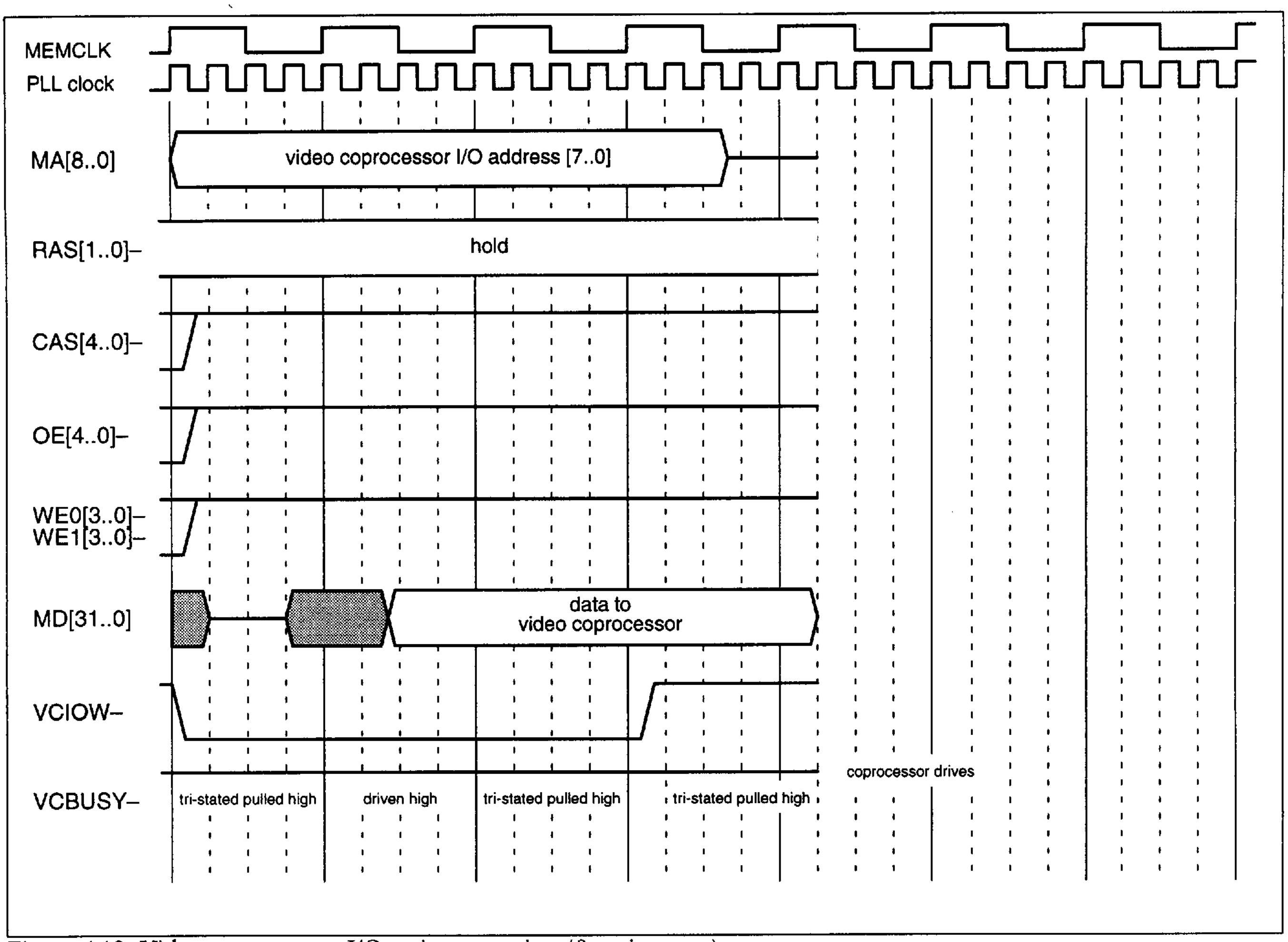


Figure 148. Video coprocessor I/O write operation (0 wait states)

9.2. Video Coprocessor I/O Write, continued

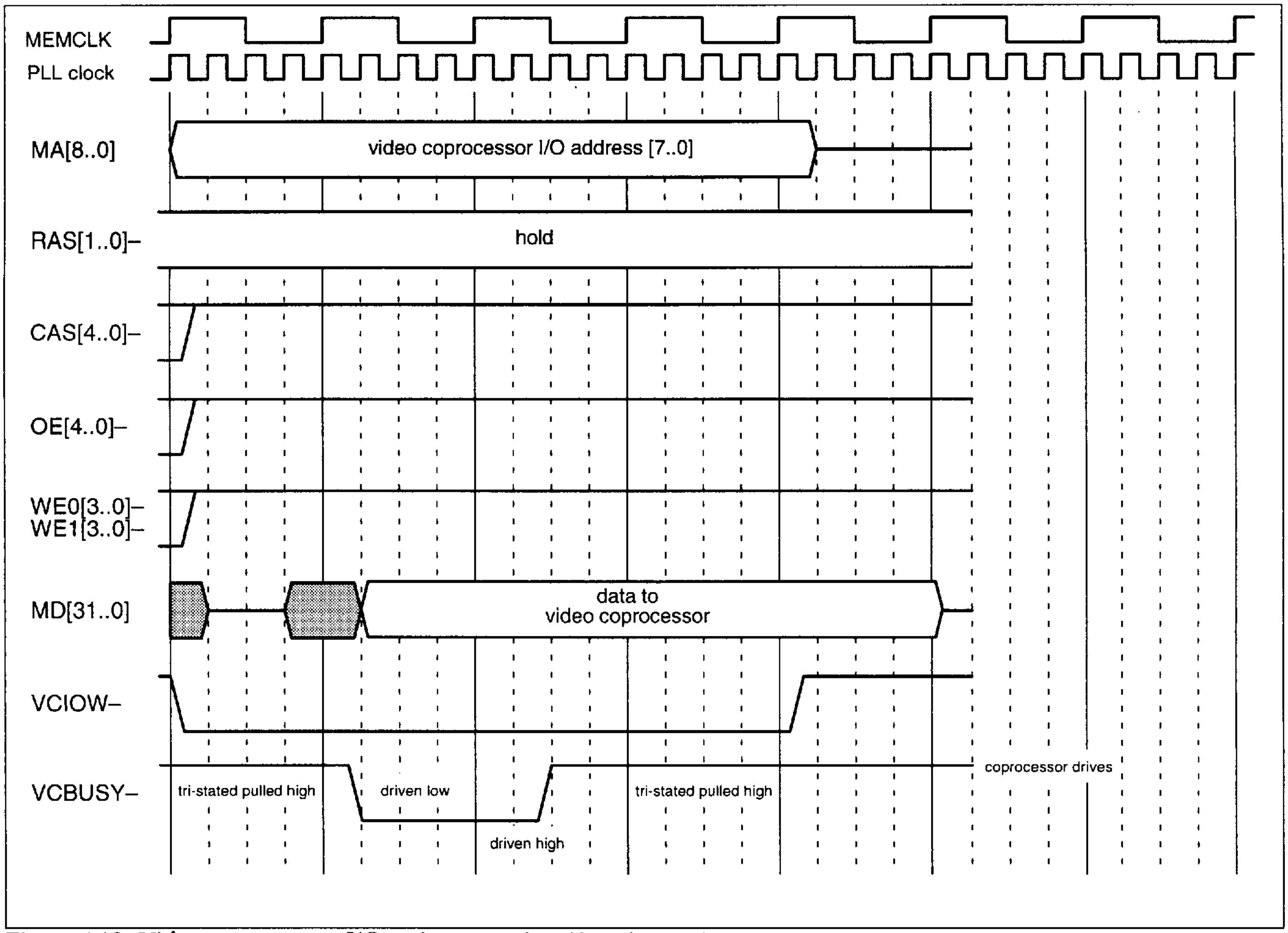


Figure 149. Video coprocessor I/O write operation (1 wait state)

9.3. Video Coprocessor Grant

The video coprocessor requests direct access to the frame buffer by asserting the VCREQ– signal. After some number of cycles, the Power 9100 will grant access to the frame buffer. Figure 150 shows the Power 9100 grant sequence.

Note that once VCREQ– is asserted, it must remain asserted until a complete grant/release sequence has completed.

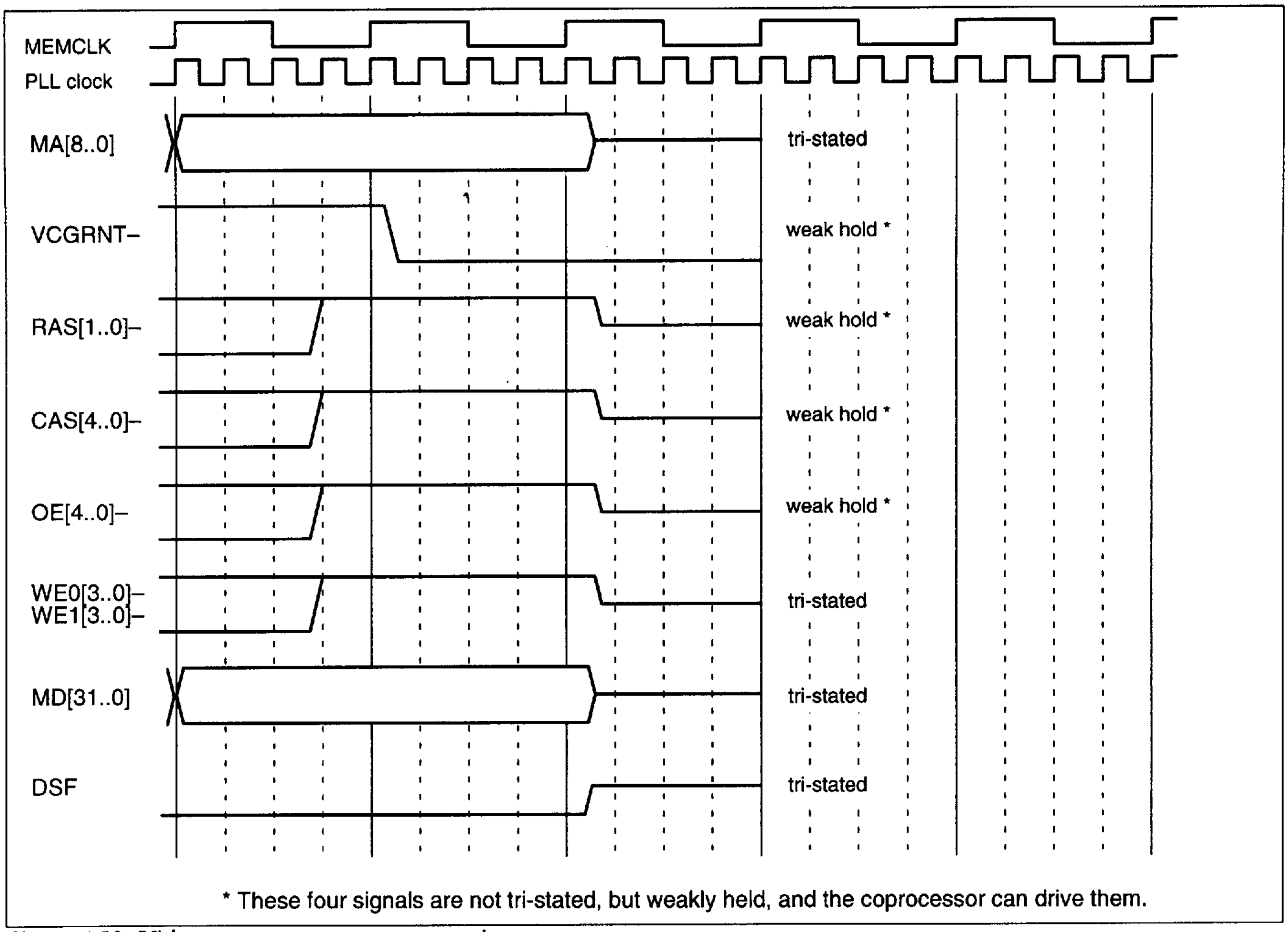
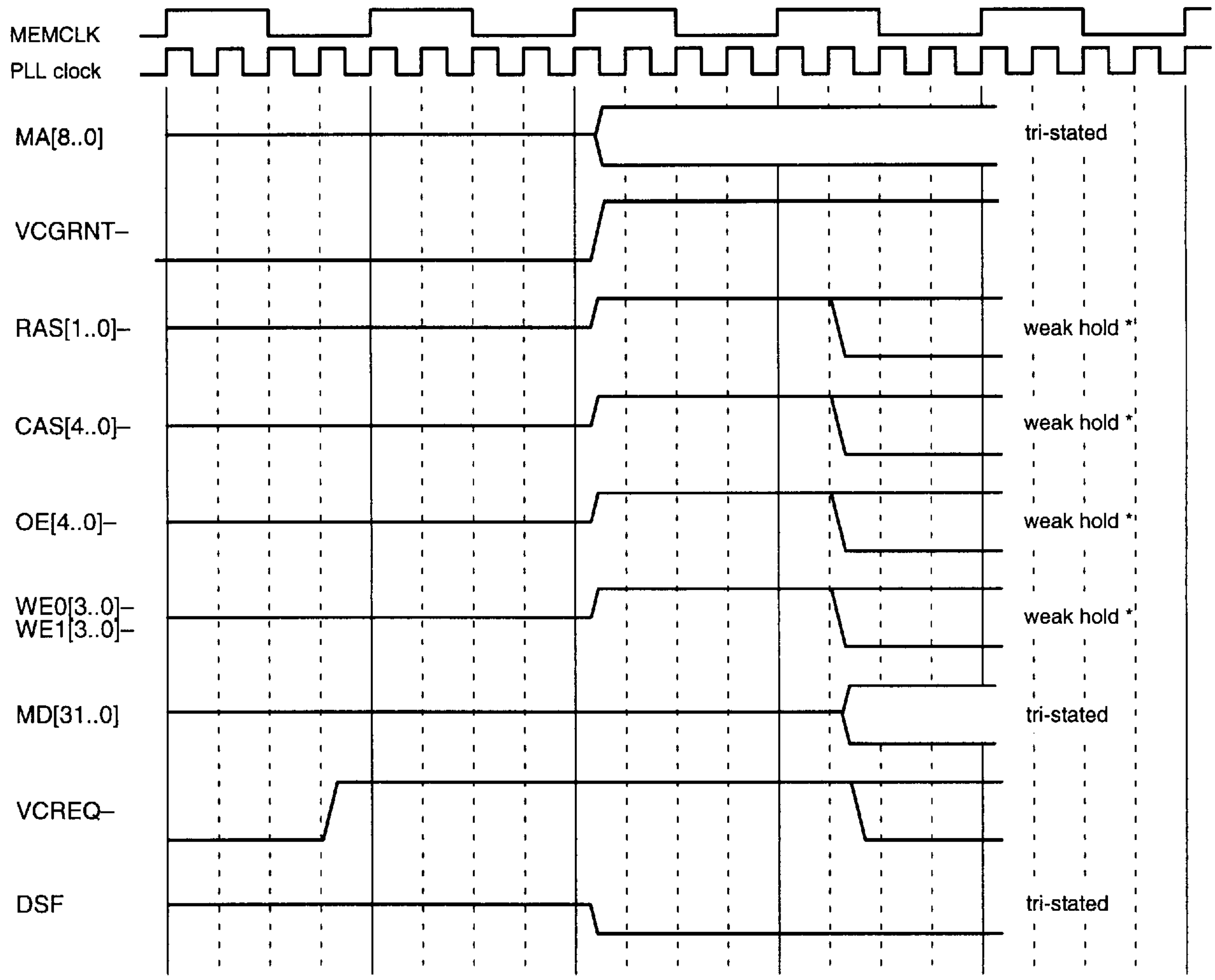


Figure 150. Video coprocessor grant operation

9.4. Video Coprocessor Release

When the video coprocessor wishes to release the frame buffer back to the Power 9100 is deasserts VCREQ–. The Power 9100 responds by deasserting VCGRNT– two cycles later. It also again drives all of the frame buffer control signals starting in the same cycle. The video coprocessor must retain VCREQ– deasserted for at least 2 cycles.



* These four signals are not tri-stated, but weakly held, and the coprocessor can drive them.

Figure 151. Video coprocessor release operation

9.5. Video Coprocessor Preempt

For higher priority operations (shift register reload and memory refresh) the Power 9100 will preempt the video coprocessor. This is done by deasserting the VCGRNT– signal. This informs the video coprocessor that a higher priority operation is pending. It must complete the current operation and release the frame buffer by deasserting the VCREQ– just like in the release operation.

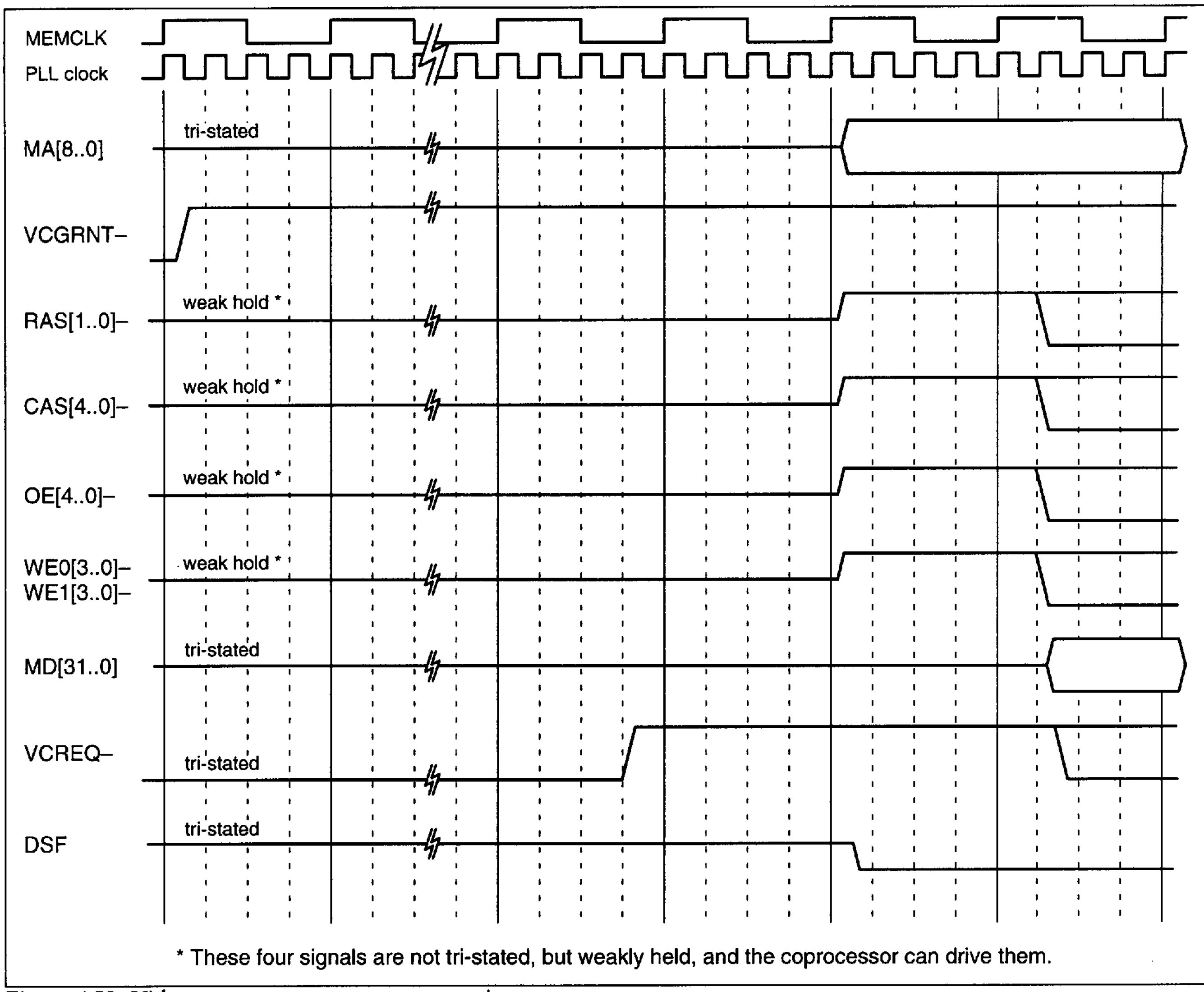


Figure 152. Video coprocessor preempt operation

Chapter 10. Auxiliary Chip Control

10.1. EEPROM Control

The optional EEPROM is connected via two shared pins: CKSEL[2] and VCEN–. The Power 9100 uses the I^2C slave protocol. The software sees three bits: CONFIG[66].VCEN, CONFIG[66].CKSEL[2], and CONFIG[64].EEDAIN. The circuit schematic for connecting them is shown in figure 153.

The Power 9100 supports the EEPROM through the two-wire I^2C interface. Both pins used for EEPROM control are potentially shared with other devices. The EEPROM clock output is shared with VCEN–, the video coprocessor enable signal. The EEPROM serial data signal is shared with the CKSEL[2] signal. Both these signals can be set through the CONFIG[66] register. The data signal can also be read, via the CONFIG[64] register.

When using the EEPROM, the CKSEL[2] pin must be dedicated to it; it cannot also be used to control a clock generator or other hardware. EEPROM and Video Power can coexist, in spite of the shared pin.

To make sure the EEPROM does not drive the bus, thus allowing the use of Video Power, the initialization code should drive VCEN– high, by writing a zero to CONFIG[66].vcen, waiting for a period determined by the stop condition setup time of the EEPROM, then setting CKSEL[2] high by writing a one to CONFIG[66].cksel2. This satisfies the "stop condition" of the EEPROM, effectively turning it off. CONFIG[66].cksel2 should then be left high indefinitely.

This allows Video Power to be accessed at will, without interference from the EEPROM.

To access the EEPROM, you must follow the I^2C protocol, as described in the EEPROM data sheets. Each time you clock the EEPROM, you toggle VCEN–. This resets Video Power, causing its register contents to be lost. Video Power resets in a tri-state mode, however, so the EEPROM access is not affected by the presence of Video Power.

Obviously, the EEPROM should not be accessed while Video Power is processing a video frame. As the EEPROM is intended to be used only for initialization, this should not provide any hardship.

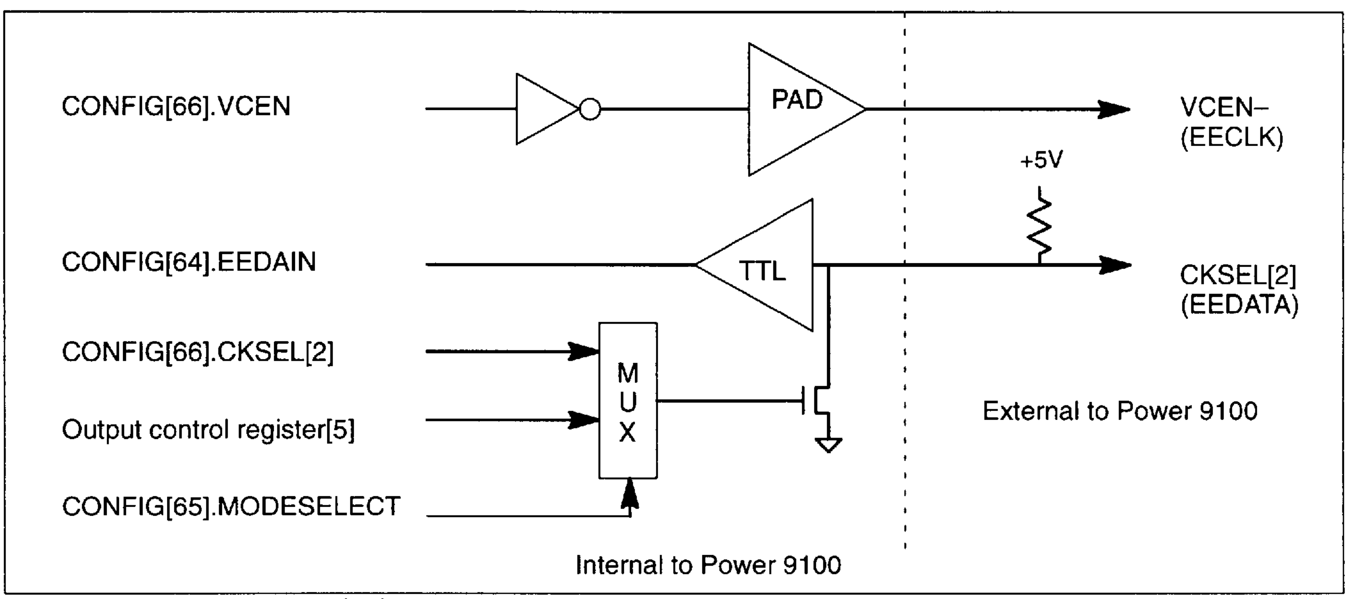


Figure 153. EEPROM control schematic

10.2. Clock Synthesizer Control

The Power 9100 directly supports programmable clock synthesizers such as the ICD2016A from IC Designs. This requires three output pins to drive CKSEL[2..0]. In emulation mode, the pins are driven directly by the output control register[5] (CKSEL[2]) and miscellaneous output register[3..2] (CKSEL[1..0]), I/O port 3C2h write/3CCh read. See sections 12.4.3 and 12.5.12. In native mode, the pins are driven by CONFIG[66].CKSEL.

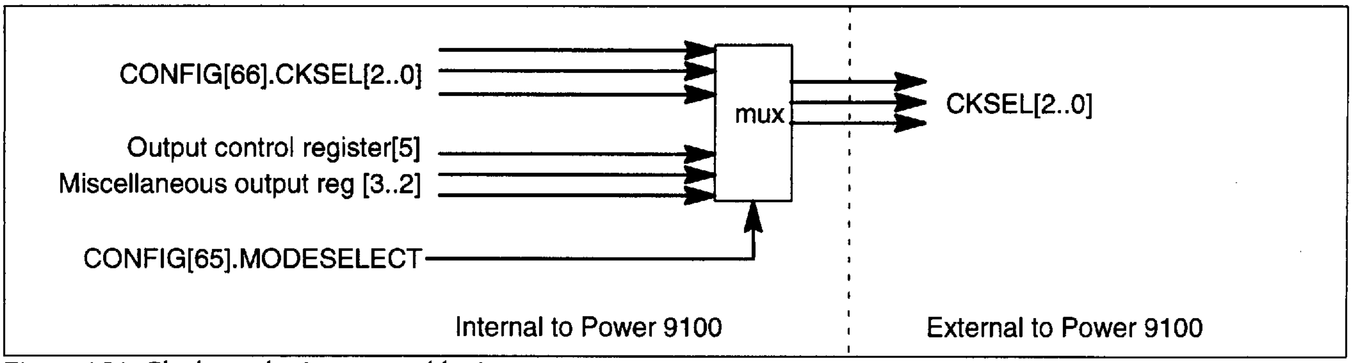


Figure 154. Clock synthesizer control logic

10.3. BIOS Access

Figure 155 shows the BIOS read cycle during the period when native mode is activated. It takes a total of eight memclk cycles to do one access. The last two cycles should provide enough data turn off time, which is typical for regular ROMs. At 50 MHz, the Power 9100 works with a 120ns ROM.

When the ROM is enabled, it is connected through the frame buffer bus. For silicon revision N4E–A4 or higher, the BIOS ROM can be disabled on VL systems through pu_config[25] (see figure 27, pu_config and figure 34, CONFIG[10]).

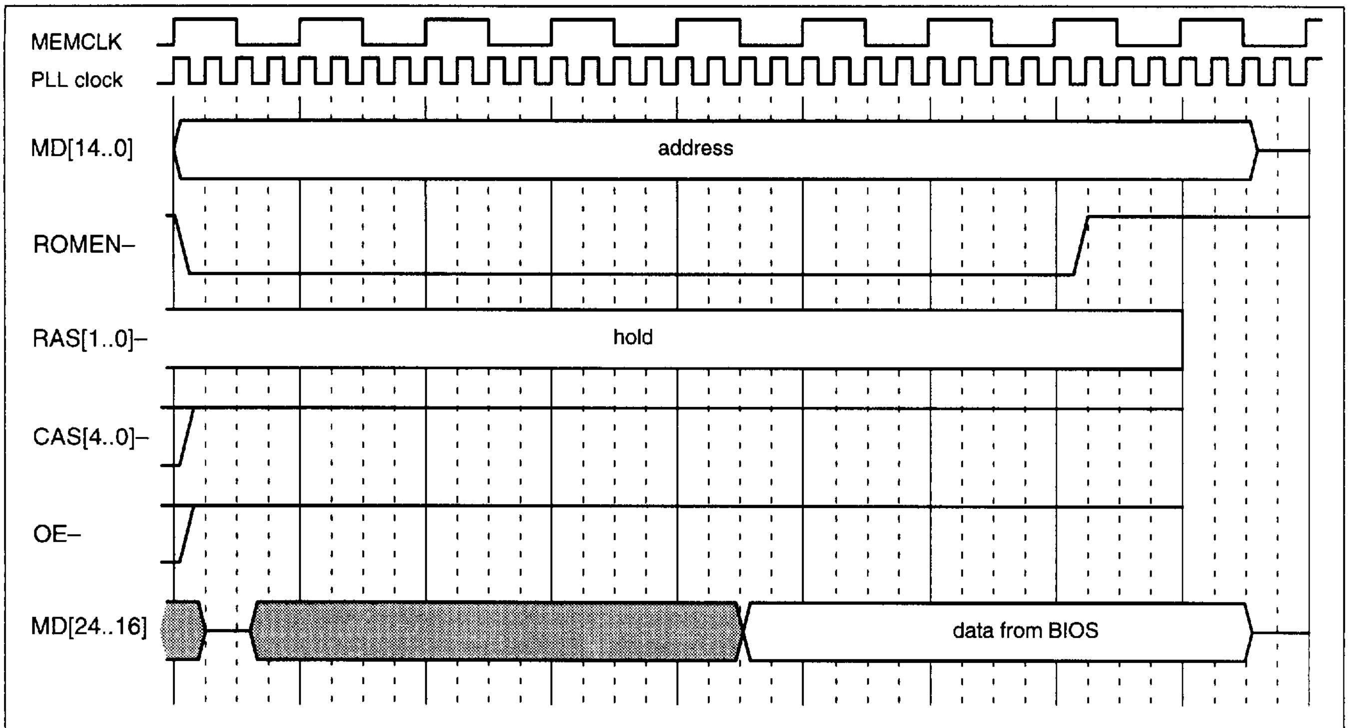


Figure 155. BIOS ROM read cycle

Chapter 11. SVGA Overview

11.1. SVGA Compatible Text/Graphics Engine

Resetting the Power 9100 sets it to VGA-emulation mode or native mode as determined by CONFIG[67].modeselect (figure 44) and PU_CONFIG.init_modeselect (figure 27). Note that upon setting native mode, the SVGA is reset. The Power 9100 is 100% VGA-compatible in VGA-emulation mode.

When used as an SVGA, the Power 9100's performance is comparable to other high-performance VGA controllers.

The Power 9100 supports all VGA modes, as shown in figure 156. (None of these modes is a Hercules-compatible graphics mode.)

VGA Name	Resolution Cols x Rows	Character Cell	Display Format	Number of Text Pages	Display Mode	Resolution (in pixels)
0	40 x 25	8 x 8	16/256K bw	8	Alpha	320 x 200
0*	40 x 25	8 x 14	16/256K bw	8	Alpha	320 x 350
0+	40 x 25	9 x 16	16/256K bw	8	Alpha	360 x 400
1	40 x 25	8 x 8	16/256K	8	Alpha	320 x 200
1*	40 x 25	8 x 14	16/256K	8	Alpha	320 x 350
1+	40 x 25	9 x 16	16/256K	8	Alpha	360 x 400
2	80 x 25	8 x 8	16/256K bw	8	Alpha	640 x 200
2*	80 x 25	8 x 14	16/256K bw	8	Alpha	720 x 400
2+	80 x 25	9 x 16	16/256K bw	8	Alpha	720 x 200
3	80 x 25	8 x 8	16/256K	8	Alpha	640 x 350
3*	80 x 25	8 x 14	16/256K	8	Alpha	720 x 400
3+	80 x 25	9 x 16	16/256K	8	Alpha	320 x 200
4	40 x 25	8 x 8	4/256K	1	Graph	320 x 200
5	40 x 25	8 x 8	4/256K bw	1	Graph	320 x 200
6	80 x 25	8 x 8	4/256K bw	1	Graph	640 x 200
7	80 x 25	9 x 14	bw	8	Alpha	720 x 400
7+	80 x 25	9 x 16	bw	8	Alpha	320 x 200
D	40 x 25	8 x 8	16/256K	8	Graph	320 x 200
E	80 x 25	8 x 8	16/256K	4	Graph	640 x 200
F	80 x 25	8 x 14	bw	2	Graph	640 x 350
10	80 x 25	8 x 14	16/256K	2	Graph	640 x 350
11	80 x 30	8 x 16	2/256K	1	Graph	640 x 480
12	80 x 30	8 x 16	16/256K	1	Graph	640 x 480
13	40 x 25	8 x 8	256/256K	1	Graph	320 x 200

Figure 156. Standard VGA display modes. All of these modes are supported by the Power 9100 SVGA mode

11.2. SVGA Enhanced Display Modes

Screen Resolution	Vertical Refresh Rates Hz	Number of Display Colors *	Frame Buffer Size**
640x480	up to 72	16, 256	512K
640x480	up to 72	32K, 16M	1MB
800x600	up to 72	16, 256	512K
800x600	up to60	32K	512K
1024x768	70 Hz max	16	512K
1024x768	70 Hz max	256	1MB
1024x768	43.5 interlaced	16	512K
1024x768	43.5 interlaced	256	1MB
1280x1024	43.5 interlaced	16	1MB

* The number of displayable colors is dependent on the DAC implementation. Most DACs with a separate 8-bit part for SVGA modes only support up to 256 colors on that port.

** In most cases, the frame buffer will be 1 MB for SVGA modes. When using 128Kx8 VRAMS only 512KB of SVGA memory is available.

Figure 157. Power 9100 enhanced modes and memory requirements in emulation mode

11.3. SVGA Signal Description

Signal	Type	Description
BLANK–	Output	Blanking period in progress
CAS0–	Output	Column-address strobe for planes 0 and 1, bank 0
CAS4–	Output	Column-address strobe for planes 2 and 3, bank 0
DACRD–	Output	DAC read
DACWR–	Output	DAC write
HSYNC	Output	Horizontal sync signal
MA[8..0]	Output	Multiplexed frame buffer address bus
MD[31..0]	I/O	Frame buffer data bus
MEMCLK	Input	Memory clock
WE0[3..0]–	Output	Memory write strobes for planes 0 through 3
PIXCLK	Input	Video pixel clock input
RAS0–	Output	Row-address strobe for lower bank planes 0 and 1
RAS1–	Output	Row-address strobe for lower bank planes 2 and 3
ROMEN–	Output	ROM enable
SENSE	Input	SENSE signal from the RAMDAC
VIDOUT[7..0]	Output	Video data output
VIDOUTCLK	Output	Video pixel clock output
VSYNC	Output	Vertical sync signal

Figure 158. SVGA signal description

Chapter 12. SVGA Registers

All documentation in the chapter applies only when the Power 9100 is functioning in SVGA emulation mode.

12.1. WEITEK-Specific Registers

The extra functionality of the Power 9100 is controlled by the Power 9100 extended registers, the Power 9100 extended bits, and a Power 9100 additional I/O port. The extended bit definitions are unused or reserved bits in existing VGA registers. Also, one EGA bit is redefined for Power 9100 use.

12.1.1. POWER 9100 EXTENDED REGISTERS

Figure 159 lists the extended Power 9100 registers.

Name	Port	Index	Lock Protection*	Section
Power 9100 Additional Sequencer Registers				
Power 9100 Control Register 0	03C5	05	Write	12.5.7
Power 9100 Control Register 1	03C5	06	Write	12.5.8
Power 9100 Revision	03C5	07	—	12.5.9
Power 9100 ID	03C5 (read)	10	Read	12.5.10
Power 9100 Miscellaneous	03C5	11	Read/Write	12.5.11
Power 9100 Output Control	03C5	12	Read/Write	12.5.12
Power 9100 Additional Controller Registers				
Power 9100 Interlace	03x5	19	—	12.6.27
Power 9100 Serial Start Address High	03x5	1A	—	12.6.28
Power 9100 Serial Start Address Low	03x5	1B	—	12.6.29
Power 9100 Serial Offset	03x5	1C	—	12.6.30
Power 9100 Total Characters per Line	03x5	1D	—	12.6.31
Power 9100 Additional Attribute Registers				
Power 9100 Overscan Color High	03C0/03C1	15	—	12.8.8
Power 9100 Additional CRT Registers				
Power 9100 Attributes states	03D5	24	—	12.6.32
* = Lock protection provided by control register lock bit (Power 9100 miscellaneous register, bit 5) *x* = B hex in monochrome mode and D hex in color mode.				

Figure 159. WEITEK-specific extended registers

12.1. WEITEK-Specific Registers, continued

LOCKING AND UNLOCKING REGISTERS

Locking and Unlocking. Except for the Power 9100 revision register (which is always available), the Power 9100 extended registers must be unlocked before they can be accessed. Bit 5 of the Power 9100 miscellaneous register (see section 12.5.11) must be reset to zero to unlock the Power 9100 extended registers using the following procedure:

1. Disable interrupts
2. Write I/O 3C4 with 11 hex
3. Write I/O 3C5
4. Write I/O 3C5
5. Read I/O 3C5 in AL
6. Logical AND AL, DF hex
7. Write to 3C5 with AL
8. Enable interrupts

12.1.2. POWER 9100 ADDITIONAL I/O PORT

The bank select register is an additional register that is accessed at the port location shown in figure 160.

Name	Port (Hex)	Section
Power 9100 Bank Select	03CD/03CD	12.4.10

Figure 160. WEITEK-specific register at additional I/O port

12.1.3. POWER 9100 EXTENDED BIT DEFINITIONS

The extended bit definitions are unused or reserved bits in existing VGA registers and are always accessible. Figure 161 lists the extended bits, the registers that contain the extended bits, and the register index for those registers.

VGA Register	Bits	Register Index	Section
Sequencer index	3,4	—	12.5.1
Clocking mode	6,7	Sequencer 01	12.5.3
CRT underline location	7	CRT controller 14	12.6.22
CRT cursor start position	6,7	CRT controller 0A	12.6.12
CRT preset row scan	7	CRT controller 08	12.6.10
Attribute mode control	1,2	Attribute 10	12.8.3
Attribute color plane enable	6,7	Attribute 12	12.8.5

Figure 161. WEITEK extended bit definitions

12.2. Register Groups

12.2.1. VGA AND WEITEK REGISTER GROUPS

The VGA and WEITEK register groups are presented in figure 162.

12.2.2. REGISTERS ORGANIZED BY GROUP

The VGA and WEITEK registers organized by group are presented in the five-part figure 163.

Group	Port	Function	Use	Mode	Fields	Section
General	94	Enable	Color, monochrome	Read, write	VGA (Motherboard)	12.4
	102	Enable	Color, monochrome	Read, write	VGA (Adaptor)	12.4
	3BA	Data	Monochrome only	Read, write	VGA	12.4
	3C2	Data	Color, monochrome	Read, write	VGA	12.4
	3C3	Enable	Color, monochrome	Read, write	VGA (Motherboard)	12.4
	3C7	Data	Color, monochrome	Read, write	VGA	12.4
	3CA	Data	Color only	Read only	VGA	12.4
	3CC	Data	Color, monochrome	Read only	VGA	12.4
	3CD	Data	Color, monochrome	Read, write	Power 9100	12.4
	3DA	Data	Color only	Read, write	VGA	12.4
	46E8	Enable	Color, monochrome	Read, write	VGA (Adaptor)	12.4
Sequencer	3C4	Index	Color, monochrome	Read, write	VGA, Power 9100	12.5
	3C5	Data	Color, monochrome	Read, write	VGA, Power 9100	12.5
CRT controller	3B4	Index	Monochrome only	Read, write	VGA	12.6
	3B5	Data	Monochrome only	Read, write	VGA, Power 9100	12.6
	3D4	Index	Color only	Read, write	VGA	12.6
	3D5	Data	Color only	Read, write	VGA, Power 9100	12.6
Graphics controller	3CE	Index	Color, monochrome	Read, write	VGA, Power 9100	12.7
	3CF	Data	Color, monochrome	Read, write	VGA, Power 9100	12.7
Attribute controller	3C0	Index and data	Color, monochrome	Write only	VGA, Power 9100	12.8
	3C1	Index and data	Color, monochrome	Read only	VGA, Power 9100	12.8

Figure 162. VGA/WEITEK register groups

12.2. Register Groups, continued

Register Group	Port	Index	Access	Use	Register	Fields	Section
General	94	—	R/W	Both	VGA enable register (Motherboard)	VGA	12.4.1
	102	—	R/W	Both	VGA enable register (Adaptor)	VGA	12.4.2
	3BA	—	Read	Mono	Input status 1 register	VGA	12.4.5
		—	Write	Mono	Feature control register	VGA	12.4.6
	3C2	—	Read	Both	Input status 0 register	VGA	12.4.4
		—	Write	Both	Miscellaneous output register	VGA	12.4.3
	3C3	—	R/W	Both	VGA enable register (Motherboard)	VGA	12.4.7
	3C7	—	R/W	Both	DAC status register	VGA	12.4.8
	3CA	—	Read	Color	Feature control register	VGA	12.4.6
	3CC	—	Read	Both	Miscellaneous output register	VGA	12.4.3
	3CD	—	R/W	Both	Power 9100 bank select register	Power 9100	12.4.10
	3DA	—	Read	Color	Input status 1 register	VGA	12.4.5
		—	Write	Color	Feature control register	VGA	12.4.6
	46E8	—	R/W	Both	VGA enable register (AT)	VGA	12.4.9
Sequencer	3C4	—	R/W	Both	Sequencer index register	Both	12.5.1
	3C5	00	R/W	Both	Reset register	VGA	12.5.2
		01	R/W	Both	Clocking mode register	Both	12.5.3
		02	R/W	Both	Map mask register	VGA	12.5.4
		03	R/W	Both	Character map select register	VGA	12.5.5
		04	R/W	Both	Memory mode register	VGA	12.5.6
		05	R/W	Both	Power 9100 control register 0	Power 9100	12.5.7
		06	R/W	Both	Power 9100 control register 1	Power 9100	12.5.8
		07	R/W	Both	Power 9100 revision register	Power 9100	12.5.9
		10	Read	Both	Power 9100 ID	Power 9100	12.5.10
		11	R/W	Both	Power 9100 miscellaneous register	Power 9100	12.5.11
		12	R/W	Both	Power 9100 output control register	Power 9100	12.5.12

Figure 163. VGA/WEITEK registers organized by group (part 1 of 6)

12.2. Register Groups, continued

Register Group	Port	Index	Access	Use	Register	Fields	Section
CRT controller	3B4	—	R/W	Mono	CRT controller index register	VGA	12.6.1
	3B5	00	R/W	Mono	Horizontal total register	VGA	12.6.2
		01	R/W	Mono	Horizontal display enable end register	VGA	12.6.3
		02	R/W	Mono	Start horizontal blanking register	VGA	12.6.4
		03	R/W	Mono	End horizontal blanking register	VGA	12.6.5
		04	R/W	Mono	Start horizontal retrace pulse register	VGA	12.6.6
		05	R/W	Mono	End horizontal retrace register	VGA	12.6.7
		06	R/W	Mono	Vertical total register	VGA	12.6.8
		07	R/W	Mono	Overflow register	VGA	12.6.9
		08	R/W	Mono	Preset row scan register	Both	12.6.10
		09	R/W	Mono	Maximum scan line register	VGA	12.6.11
		0A	R/W	Mono	Cursor start register	Both	12.6.12
		0B	R/W	Mono	Cursor end register	VGA	12.6.13
		0C	R/W	Mono	Start address high register	VGA	12.6.14
		0D	R/W	Mono	Start address low register	VGA	12.6.15
		0E	R/W	Mono	Cursor location high register	VGA	12.6.16
		0F	R/W	Mono	Cursor location low register	VGA	12.6.17
		10	R/W	Mono	Vertical retrace start register	VGA	12.6.18
		11	R/W	Mono	Vertical retrace end register	VGA	12.6.19
		12	R/W	Mono	Vertical display enable end register	VGA	12.6.20
		13	R/W	Mono	Offset register	VGA	12.6.21
		14	R/W	Mono	Underline location register	Both	12.6.22
		15	R/W	Mono	Start vertical blank register	VGA	12.6.23
		16	R/W	Mono	End vertical blank register	VGA	12.6.24
		17	R/W	Mono	CRT mode control register	VGA	12.6.25
		18	R/W	Mono	Line compare register	VGA	12.6.26
		19	R/W	Mono	Power 9100 interlace register	Power 9100	12.6.27
		1A	R/W	Mono	Power 9100 serial start address high reg.	Power 9100	12.6.28
		1B	R/W	Mono	Power 9100 serial start address low reg.	Power 9100	12.6.29
		1C	R/W	Mono	Power 9100 serial offset register	Power 9100	12.6.30
		1D	R/W	Mono	Power 9100 total characters per line reg.	Power 9100	12.6.31

Figure 163. VGA/WEITEK registers organized by group (part 2 of 6)

12.2. Register Groups, continued

Register Group	Port	Index	Access	Use	Register	Fields	Section
CRT controller	3D4	—	R/W	Color	CRT controller index register	VGA	12.6.1
(continued)	3D5	00	R/W	Color	Horizontal total register	VGA	12.6.2
		01	R/W	Color	Horizontal display enable end register	VGA	12.6.3
		02	R/W	Color	Start horizontal blanking register	VGA	12.6.4
		03	R/W	Color	End horizontal blanking register	VGA	12.6.5
		04	R/W	Color	Start horizontal retrace pulse register	VGA	12.6.6
		05	R/W	Color	End horizontal retrace register	VGA	12.6.7
		06	R/W	Color	Vertical total register	VGA	12.6.8
		07	R/W	Color	Overflow register	VGA	12.6.9
		08	R/W	Color	Preset row scan register	Both	12.6.10
		09	R/W	Color	Maximum scan line register	VGA	12.6.11
		0A	R/W	Color	Cursor start register	Both	12.6.12
		0B	R/W	Color	Cursor end register	VGA	12.6.13
		0C	R/W	Color	Start address high register	VGA	12.6.14
		0D	R/W	Color	Start address low register	VGA	12.6.15
		0E	R/W	Color	Cursor location high register	VGA	12.6.16
		0F	R/W	Color	Cursor location low register	VGA	12.6.17
		10	R/W	Color	Vertical retrace start register	VGA	12.6.18
		11	R/W	Color	Vertical retrace end register	VGA	12.6.19
		12	R/W	Color	Vertical display enable end register	VGA	12.6.20
		13	R/W	Color	Offset register	VGA	12.6.21
		14	R/W	Color	Underline location register	Both	12.6.22
		15	R/W	Color	Start vertical blank register	VGA	12.6.23
		16	R/W	Color	End vertical blank register	VGA	12.6.24
		17	R/W	Color	CRT mode control register	VGA	12.6.25
		18	R/W	Color	Line compare register	VGA	12.6.26
		19	R/W	Color	Power 9100 interlace register	Power 9100	12.6.27
		1A	R/W	Color	Power 9100 serial start address high reg.	Power 9100	12.6.28
		1B	R/W	Color	Power 9100 serial start address low reg.	Power 9100	12.6.29
		1C	R/W	Color	Power 9100 serial offset register	Power 9100	12.6.30
		1D	R/W	Color	Power 9100 total characters per line reg.	Power 9100	12.6.31
		24	R		Power 9100 attributes state register	Power 9100	12.6.32

Figure 163. VGA/WEITEK registers organized by group (part 3 of 6)

12.2. Register Groups, continued

Register Group	Port	Index	Access	Use	Register	Fields	Section
Graphics controller	3CE	—	R/W	Both	Graphics index register	Both	12.7.1
	3CF	00	R/W	Both	Set/reset register	VGA	12.7.2
		01	R/W	Both	Enable set/reset register	VGA	12.7.3
		02	R/W	Both	Color compare register	VGA	12.7.4
		03	R/W	Both	Data rotate register	Both	12.7.5
		04	R/W	Both	Read map select register	VGA	12.7.6
		05	R/W	Both	Graphics mode register	Both	12.7.7
		06	R/W	Both	Miscellaneous register	VGA	12.7.8
		07	R/W	Both	Color don't care register	VGA	12.7.9
		08	R/W	Both	Bit mask register	VGA	12.7.10
		09			Reserved		
		0A			Reserved		
		0B			Reserved		
		0C			Reserved		
		0D			Reserved		
		0E			Reserved		
		0F			Reserved		

Figure 163. VGA/WEITEK registers organized by group (part 4 of 6)

12.2. Register Groups, continued

Register Group	Port	Index	Access	Use	Register	Fields	Section
Attribute controller	3C0	—	Write	Both	Attribute index register	VGA	12.8.1
		00	Write	Both	Palette register 00	VGA	12.8.2
		01	Write	Both	Palette register 01	VGA	12.8.2
		02	Write	Both	Palette register 02	VGA	12.8.2
		03	Write	Both	Palette register 03	VGA	12.8.2
		04	Write	Both	Palette register 04	VGA	12.8.2
		05	Write	Both	Palette register 05	VGA	12.8.2
		06	Write	Both	Palette register 06	VGA	12.8.2
		07	Write	Both	Palette register 07	VGA	12.8.2
		08	Write	Both	Palette register 08	VGA	12.8.2
		09	Write	Both	Palette register 09	VGA	12.8.2
		0A	Write	Both	Palette register 0A	VGA	12.8.2
		0B	Write	Both	Palette register 0B	VGA	12.8.2
		0C	Write	Both	Palette register 0C	VGA	12.8.2
		0D	Write	Both	Palette register 0D	VGA	12.8.2
		0E	Write	Both	Palette register 0E	VGA	12.8.2
		0F	Write	Both	Palette register 0F	VGA	12.8.2
		10	Write	Both	Attribute mode control register	Both	12.8.3
		11	Write	Both	Overscan control register	VGA	12.8.4
		12	Write	Both	Color plane enable register	Both	12.8.5
		13	Write	Both	Horizontal pixel panning register	VGA	12.8.6
		14	Write	Both	Color select register	VGA	12.8.7
		15	Write	Both	Power 9100 overscan color high register	Power 9100	12.8.8

Figure 163. VGA/WEITEK registers organized by group (part 5 of 6)

12.2. Register Groups, continued

Register Group	Port	Index	Access	Use	Register	Fields	Section
Attribute controller (continued)	3C1	—	Read	Both	Attribute index register	VGA	12.8.1
		00	Read	Both	Palette register 00	VGA	12.8.2
		01	Read	Both	Palette register 01	VGA	12.8.2
		02	Read	Both	Palette register 02	VGA	12.8.2
		03	Read	Both	Palette register 03	VGA	12.8.2
		04	Read	Both	Palette register 04	VGA	12.8.2
		05	Read	Both	Palette register 05	VGA	12.8.2
		06	Read	Both	Palette register 06	VGA	12.8.2
		07	Read	Both	Palette register 07	VGA	12.8.2
		08	Read	Both	Palette register 08	VGA	12.8.2
		09	Read	Both	Palette register 09	VGA	12.8.2
		0A	Read	Both	Palette register 0A	VGA	12.8.2
		0B	Read	Both	Palette register 0B	VGA	12.8.2
		0C	Read	Both	Palette register 0C	VGA	12.8.2
		0D	Read	Both	Palette register 0D	VGA	12.8.2
		0E	Read	Both	Palette register 0E	VGA	12.8.2
		0F	Read	Both	Palette register 0F	VGA	12.8.2
		10	Read	Both	Attribute mode control register	Both	12.8.3
		11	Read	Both	Overscan control register	VGA	12.8.4
		12	Read	Both	Color plane enable register	Both	12.8.5
		13	Read	Both	Horizontal pixel panning register	VGA	12.8.6
		14	Read	Both	Color select register	VGA	12.8.7
		15	Read	Both	Power 9100 overscan color high register	Power 9100	12.8.8

Figure 163. VGA/WEITEK registers organized by group (part 6 of 6)

12.3. Memory Map

Port	Index	Group	Access	Use	Register	Fields	Section
94	—	General	R/W	Both	VGA enable register (Motherboard)	VGA	12.4.1
102	—	General	R/W	Both	VGA enable register (Adaptor)	VGA	12.4.2
3B4	—	CRT controller	R/W	Mono	CRT controller index register	VGA	12.6.1
3B5	00	CRT controller	R/W	Mono	Horizontal total register	VGA	12.6.2
	01	CRT controller	R/W	Mono	Horizontal display enable end register	VGA	12.6.3
	02	CRT controller	R/W	Mono	Start horizontal blanking register	VGA	12.6.4
	03	CRT controller	R/W	Mono	End horizontal blanking register	VGA	12.6.5
	04	CRT controller	R/W	Mono	Start horizontal retrace pulse register	VGA	12.6.6
	05	CRT controller	R/W	Mono	End horizontal retrace register	VGA	12.6.7
	06	CRT controller	R/W	Mono	Vertical total register	VGA	12.6.8
	07	CRT controller	R/W	Mono	Overflow register	VGA	12.6.9
	08	CRT controller	R/W	Mono	Preset row scan register	Both	12.6.10
	09	CRT controller	R/W	Mono	Maximum scan line register	VGA	12.6.11
	0A	CRT controller	R/W	Mono	Cursor start register	Both	12.6.12
	0B	CRT controller	R/W	Mono	Cursor end register	VGA	12.6.13
	0C	CRT controller	R/W	Mono	Start address high register	VGA	12.6.14
	0D	CRT controller	R/W	Mono	Start address low register	VGA	12.6.15
	0E	CRT controller	R/W	Mono	Cursor location high register	VGA	12.6.16
	0F	CRT controller	R/W	Mono	Cursor location low register	VGA	12.6.17
	10	CRT controller	R/W	Mono	Vertical retrace start register	VGA	12.6.18
	11	CRT controller	R/W	Mono	Vertical retrace end register	VGA	12.6.19
	12	CRT controller	R/W	Mono	Vertical display enable end register	VGA	12.6.20
	13	CRT controller	R/W	Mono	Offset register	VGA	12.6.21
	14	CRT controller	R/W	Mono	Underline location register	Both	12.6.22
	15	CRT controller	R/W	Mono	Start vertical blank register	VGA	12.6.23
	16	CRT controller	R/W	Mono	End vertical blank register	VGA	12.6.24
	17	CRT controller	R/W	Mono	CRT mode control register	VGA	12.6.25
	18	CRT controller	R/W	Mono	Line compare register	VGA	12.6.26
	19	CRT controller	R/W	Mono	Power 9100 interlace register	Power 9100	12.6.27
	1A	CRT controller	R/W	Mono	Power 9100 serial start address high reg.	Power 9100	12.6.28
	1B	CRT controller	R/W	Mono	Power 9100 serial start address low reg.	Power 9100	12.6.29
	1C	CRT controller	R/W	Mono	Power 9100 serial offset register	Power 9100	12.6.30
	1D	CRT controller	R/W	Mono	Power 9100 total characters per line reg.	Power 9100	12.6.31

Figure 164. VGA/WEITEK registers organized by I/O address (part 1 of 5)

12.3. Memory Map, continued

Port	Index	Group	Access	Use	Register	Fields	Section
3BA	—	General	Read	Mono	Input status 1 register	VGA	12.4.5
	—	General	Write	Mono	Feature control register	VGA	12.4.6
3C0	—	Attribute controller	Write	Both	Attribute index register	VGA	12.8.1
	00	Attribute controller	Write	Both	Palette register 00	VGA	12.8.2
	01	Attribute controller	Write	Both	Palette register 01	VGA	12.8.2
	02	Attribute controller	Write	Both	Palette register 02	VGA	12.8.2
	03	Attribute controller	Write	Both	Palette register 03	VGA	12.8.2
	04	Attribute controller	Write	Both	Palette register 04	VGA	12.8.2
	05	Attribute controller	Write	Both	Palette register 05	VGA	12.8.2
	06	Attribute controller	Write	Both	Palette register 06	VGA	12.8.2
	07	Attribute controller	Write	Both	Palette register 07	VGA	12.8.2
	08	Attribute controller	Write	Both	Palette register 08	VGA	12.8.2
	09	Attribute controller	Write	Both	Palette register 09	VGA	12.8.2
	0A	Attribute controller	Write	Both	Palette register 0A	VGA	12.8.2
	0B	Attribute controller	Write	Both	Palette register 0B	VGA	12.8.2
	0C	Attribute controller	Write	Both	Palette register 0C	VGA	12.8.2
	0D	Attribute controller	Write	Both	Palette register 0D	VGA	12.8.2
	0E	Attribute controller	Write	Both	Palette register 0E	VGA	12.8.2
	0F	Attribute controller	Write	Both	Palette register 0F	VGA	12.8.2
	10	Attribute controller	Write	Both	Attribute mode control register	Both	12.8.3
	11	Attribute controller	Write	Both	Overscan control register	VGA	12.8.4
	12	Attribute controller	Write	Both	Color plane enable register	Both	12.8.5
	13	Attribute controller	Write	Both	Horizontal pixel panning register	VGA	12.8.6
	14	Attribute controller	Write	Both	Color select register	VGA	12.8.7
	15	Attribute controller	Write	Both	Power 9100 overscan color high register	Power 9100	12.8.8

Figure 164. VGA/WEITEK registers organized by I/O address (part 2 of 5)

12.3. Memory Map, continued

Port	Index	Group	Access	Use	Register	Fields	Section
3C1	—	Attribute controller	Read	Both	Attribute index register	VGA	12.8.1
	00	Attribute controller	Read	Both	Palette register 00	VGA	12.8.2
	01	Attribute controller	Read	Both	Palette register 01	VGA	12.8.2
	02	Attribute controller	Read	Both	Palette register 02	VGA	12.8.2
	03	Attribute controller	Read	Both	Palette register 03	VGA	12.8.2
	04	Attribute controller	Read	Both	Palette register 04	VGA	12.8.2
	05	Attribute controller	Read	Both	Palette register 05	VGA	12.8.2
	06	Attribute controller	Read	Both	Palette register 06	VGA	12.8.2
	07	Attribute controller	Read	Both	Palette register 07	VGA	12.8.2
	08	Attribute controller	Read	Both	Palette register 08	VGA	12.8.2
	09	Attribute controller	Read	Both	Palette register 09	VGA	12.8.2
	0A	Attribute controller	Read	Both	Palette register 0A	VGA	12.8.2
	0B	Attribute controller	Read	Both	Palette register 0B	VGA	12.8.2
	0C	Attribute controller	Read	Both	Palette register 0C	VGA	12.8.2
	0D	Attribute controller	Read	Both	Palette register 0D	VGA	12.8.2
	0E	Attribute controller	Read	Both	Palette register 0E	VGA	12.8.2
	0F	Attribute controller	Read	Both	Palette register 0F	VGA	12.8.2
	10	Attribute controller	Read	Both	Attribute mode control register	Both	12.8.3
	11	Attribute controller	Read	Both	Overscan control register	VGA	12.8.4
	12	Attribute controller	Read	Both	Color plane enable register	Both	12.8.5
	13	Attribute controller	Read	Both	Horizontal pixel panning register	VGA	12.8.6
	14	Attribute controller	Read	Both	Color select register	VGA	12.8.7
	15	Attribute controller	Read	Both	Power 9100 overscan color high register	Power 9100	12.8.8
3C2	—	General	Read	Both	Input status 0 register	VGA	12.4.4
	—	General	Write	Both	Miscellaneous output register	VGA	12.4.3
3C3	—	General	R/W	Both	VGA enable register (Motherboard)	VGA	12.4.7

Figure 164. VGA/WEITEK registers organized by I/O address (part 3 of 5)

12.3. Memory Map, continued

Port	Index	Group	Access	Use	Register	Fields	Section
3C4	—	Sequencer	R/W	Both	Sequencer index register	Both	12.5.1
3C5	00	Sequencer	R/W	Both	Reset register	VGA	12.5.2
	01	Sequencer	R/W	Both	Clocking mode register	Both	12.5.3
	02	Sequencer	R/W	Both	Map mask register	VGA	12.5.4
	03	Sequencer	R/W	Both	Character map select register	VGA	12.5.5
	04	Sequencer	R/W	Both	Memory mode register	VGA	12.5.6
	05	Sequencer	R/W	Both	Power 9100 control register 0	Power 9100	12.5.7
	06	Sequencer	R/W	Both	Power 9100 control register 1	Power 9100	12.5.8
	07	Sequencer	R/W	Both	Power 9100 revision register	Power 9100	12.5.9
	10	Sequencer	Read	Both	Power 9100 ID	Power 9100	12.5.10
	11	Sequencer	R/W	Both	Power 9100 miscellaneous register	Power 9100	12.5.11
	12	Sequencer	R/W	Both	Power 9100 output control register	Power 9100	12.5.12
3C7	—	General	R/W	Both	DAC status register	VGA	12.4.8
3CA	—	General	Read	Color	Feature control register	VGA	12.4.6
3CC	—	General	Read	Both	Miscellaneous output register	VGA	12.4.3
3CD	—	General	R/W	Both	Power 9100 bank select register	Power 9100	12.4.10
3CE	—	Graphics controller	R/W	Both	Graphics index register	Both	12.7.1
3CF	00	Graphics controller	R/W	Both	Set/reset register	VGA	12.7.2
	01	Graphics controller	R/W	Both	Enable set/reset register	VGA	12.7.3
	02	Graphics controller	R/W	Both	Color compare register	VGA	12.7.4
	03	Graphics controller	R/W	Both	Data rotate register	Both	12.7.5
	04	Graphics controller	R/W	Both	Read map select register	VGA	12.7.6
	05	Graphics controller	R/W	Both	Graphics mode register	Both	12.7.7
	06	Graphics controller	R/W	Both	Miscellaneous register	VGA	12.7.8
	07	Graphics controller	R/W	Both	Color don't care register	VGA	12.7.9
	08	Graphics controller	R/W	Both	Bit mask register	VGA	12.7.10
	09	(Reserved)			Reserved		
	0A	(Reserved)			Reserved		
	0B	(Reserved)			Reserved		
	0C	(Reserved)			Reserved		
	0D	(Reserved)			Reserved		
	0E	(Reserved)			Reserved		
	0F	(Reserved)			Reserved		

Figure 164. VGA/WEITEK registers organized by I/O address (part 4 of 5)

Port	Index	Group	Access	Use	Register	Fields	Section
3D4	—	CRT controller	R/W	Color	CRT controller index register	VGA	12.6.1
3D5	00	CRT controller	R/W	Color	Horizontal total register	VGA	12.6.2
	01	CRT controller	R/W	Color	Horizontal display enable end register	VGA	12.6.3
	02	CRT controller	R/W	Color	Start horizontal blanking register	VGA	12.6.4
	03	CRT controller	R/W	Color	End horizontal blanking register	VGA	12.6.5
	04	CRT controller	R/W	Color	Start horizontal retrace pulse register	VGA	12.6.6
	05	CRT controller	R/W	Color	End horizontal retrace register	VGA	12.6.7
	06	CRT controller	R/W	Color	Vertical total register	VGA	12.6.8
	07	CRT controller	R/W	Color	Overflow register	VGA	12.6.9
	08	CRT controller	R/W	Color	Preset row scan register	Both	12.6.10
	09	CRT controller	R/W	Color	Maximum scan line register	VGA	12.6.11
	0A	CRT controller	R/W	Color	Cursor start register	Both	12.6.12
	0B	CRT controller	R/W	Color	Cursor end register	VGA	12.6.13
	0C	CRT controller	R/W	Color	Start address high register	VGA	12.6.14
	0D	CRT controller	R/W	Color	Start address low register	VGA	12.6.15
	0E	CRT controller	R/W	Color	Cursor location high register	VGA	12.6.16
	0F	CRT controller	R/W	Color	Cursor location low register	VGA	12.6.17
	10	CRT controller	R/W	Color	Vertical retrace start register	VGA	12.6.18
	11	CRT controller	R/W	Color	Vertical retrace end register	VGA	12.6.19
	12	CRT controller	R/W	Color	Vertical display enable end register	VGA	12.6.20
	13	CRT controller	R/W	Color	Offset register	VGA	12.6.21
	14	CRT controller	R/W	Color	Underline location register	Both	12.6.22
	15	CRT controller	R/W	Color	Start vertical blank register	VGA	12.6.23
	16	CRT controller	R/W	Color	End vertical blank register	VGA	12.6.24
	17	CRT controller	R/W	Color	CRT mode control register	VGA	12.6.25
	18	CRT controller	R/W	Color	Line compare register	VGA	12.6.26
	19	CRT controller	R/W	Color	Power 9100 interlace register	Power 9100	12.6.27
	1A	CRT controller	R/W	Color	Power 9100 serial start address high reg.	Power 9100	12.6.28
	1B	CRT controller	R/W	Color	Power 9100 serial start address low reg.	Power 9100	12.6.29
	1C	CRT controller	R/W	Color	Power 9100 serial offset register	Power 9100	12.6.30
	1D	CRT controller	R/W	Color	Power 9100 total characters per line reg.	Power 9100	12.6.31
	24	CRT controller	Read		Power 9100 attributes states	Power 9100	12.6.32
3DA	—	General	Read	Color	Input status 1 register	VGA	12.4.5
	—	General	Write	Color	Feature control register	VGA	12.4.6
46E8	—	General	R/W	Both	VGA enable register (Adaptor)	VGA	12.4.9

Figure 164. VGA/WEITEK registers organized by I/O address (part 5 of 5)

12.4. General Registers

The general register group controls the timing interface between the VGA hardware and the CPU, system memory, palette DAC, and monitor.

Register Type	General Register	Port	Section
Standard VGA registers	VGA enable register (Motherboard)	94	12.4.1
	VGA enable register (Adaptor)	102	12.4.2
	Miscellaneous output register	3CC (read)	12.4.3
		3C2 (write)	12.4.3
	Input status 0 register	3C2 (read)	12.4.4
	Input status 1 register	3BA (monochrome, read)	12.4.5
		3DA (color, read)	12.4.5
	Feature control register	3CA (color, read)	12.4.6
		3BA (monochrome, write)	12.4.6
		3DA (color, write)	12.4.6
	VGA enable register (Motherboard)	3C3	12.4.7
	DAC status register	3C7 (read)	12.4.8
	VGA enable register (Adaptor)	46E8	12.4.9

Figure 165. General registers

12.4.1. VGA ENABLE REGISTER

The *VGA enable register* enables and disables the video I/O and memory address decoding. The register at port 94 is used only in motherboard systems.

REGISTER FORMAT

VGA Enable (microChannel) Port 94

7	6	5	4	3	2	1	0
Reserved		VSE5	R	VSE3	Reserved		

Bit No.	Meaning
7..6	Reserved
5	Video Subsystem Enable (VSE5)
4	Reserved
3	Video Subsystem Enable (VSE3)
2..0	Reserved

Figure 166. VGA enable register format

FIELD DEFINITION

Bit 5	Bit 3	Meaning
0	0	Disable video I/O and address decoding
0	1	Disable video I/O and address decoding
1	0	Disable video I/O and address decoding
1	1	Enable video I/O and address decoding

Figure 167. VGA enable register fields

REGISTER DESCRIPTION

Port 102 is enabled only when bit 4 of address 46E8 is 1 (for adaptor systems) and when bit 5 of address 94 is 0 (for motherboard systems). Otherwise, port 102 remains locked.

The Power 9100 enable logic for adaptor systems is presented in figure 183 and the enable logic for motherboard systems is presented in figure 184.

For motherboard configurations, the Power 9100 responds to ports 3C3 and 94 only. For adapter configurations, the Power 9100 responds to address 46E8 and disregards any attempt to write to ports 3C3 or 94.

12.4.2. VGA ENABLE REGISTER

The *VGA enable register* enables and disables the video I/O and memory address decoding. The register at port 102 is used in both adaptor and motherboard systems.

REGISTER FORMAT

VGA Enable Port 102

7	6	5	4	3	2	1	0
Reserved							VSE

Bit No.	Meaning
7..1	Reserved
0	Video Subsystem Enable (VSE)

Figure 168. VGA enable register format

FIELD DEFINITION

Bit	Value	Meaning
0	0	Disable video I/O and address decoding
	1	Enable video I/O and address decoding

Figure 169. VGA enable register fields

REGISTER DESCRIPTION

Port 102 is enabled only when bit 4 of address 46E8 is 1 (for adaptor systems) and when bit 5 of address 94 is 0 (for motherboard systems). Otherwise, port 102 remains locked.

The Power 9100 enable logic for adaptor systems is presented in figure 183 and the enable logic for motherboard systems is presented in figure 184.

12.4. General Registers, continued

12.4.3. MISCELLANEOUS OUTPUT REGISTER

The *miscellaneous output register* controls sync pulse polarity, clock frequency, CPU access, and emulation.

REGISTER FORMAT

Miscellaneous Output Port 3CC (Read), 3C2 (Write)

7	6	5	4	3 2	1	0
VSP	HSP	PB	R	CS	ER	IOA

Bit No.	Meaning
7	Vertical Sync Polarity (VSP)
6	Horizontal Sync Polarity (HSP)
5	Page Bit for Even/Odd (PB)
4	Reserved
3..2	Clock Select (CS)
1	Enable RAM (ER)
0	Input/Output Address (IOA)

Figure 170. Miscellaneous output register format

REGISTER DESCRIPTION

All bits of this register are reset to 0 with a hardware reset.

Bits 7 and 6 specify sync polarity to determine the scan rate in multifrequency monitors (see figure 171).

Bits 3 and 2 control the pixel clock generator. In emulation mode, bit 3 is CKSEL[1] and bit 2 is CKSEL[0]. Reset state is 00b. For CKSEL[3], see section 12.5.12 and section 10.2. To control CKSEL[2..0] in native mode, see section 10.2.

Bit 0 determines the address location of the CRT controller index register of the CRT controller registers group (see figure 215).

Bit 7	Bit 6	Vertical Size	Active Lines
0	0	Reserved	Reserved
0	1	400 lines	414 lines
1	0	350 lines	362 lines
1	1	480 lines	496 lines

Figure 171. Vertical size and sync polarity

FIELD DEFINITION

Bits	Value	Meaning
7	0	Positive vertical retrace sync pulse
	1	Negative vertical retrace sync pulse
6	0	Positive horizontal retrace pulse
	1	Negative horizontal retrace pulse
5	0	Low 64K memory page, diagnostic use in Odd/Even modes (0–5)
	1	High 64K memory page, diagnostic use in Odd/Even modes (0–5)
3..2	00	25.175 MHz clock (640 horizontal pixels)
	01	28.322 MHz clock (720 horizontal pixels)
	10	External clock from auxiliary video connector (clock between 14.3 and 28.4 MHz)
	11	Reserved
1	0	Disable video memory access from CPU
	1	Enable video memory access from CPU
0	0	Monochrome emulation (CRTC addresses set to 3B*x*, input status 1 register address set to 3BA)
	1	Color emulation (CRTC addresses set to 3D*x*, input status 1 register address set to 3DA)

Figure 172. Miscellaneous output register fields

12.4. General Registers, continued

12.4.4. INPUT STATUS 0 REGISTER

The *input status 0 register* monitors the status of the vertical retrace interrupt and senses the setting of the selected configuration switch on the hardware board.

REGISTER FORMAT

Input Status 0 — Port 3C2 (Read)

7	6	5	4	3	2	1	0
CI	Reserved		SS	Reserved			

Bit No.	Meaning
7	CRT Interrupt (CI)
6..5	Reserved
4	Switch Sense (SS)
3..0	Reserved

Figure 173. Input status 0 register format

FIELD DEFINITION

Bit	Value	Meaning
7	0	Vertical retrace interrupt is pending
	1	Vertical retrace interrupt is cleared
4	0	Selected sense switch = OFF
	1	Selected sense switch = ON

Figure 174. Input status 0 register fields

REGISTER DESCRIPTION

Bit 4 reports the status of the sense pin (167). This information is used by the software at power-on self-test to determine the type of display hardware (color or monochrome) being used.

12.4. General Registers, continued

12.4.5. INPUT STATUS 1 REGISTER

The *input status 1 register* permits the software to examine the color outputs of the attribute controller and to synchronize updates with display retrace periods.

REGISTER FORMAT

Input Status 1 Port 3DA/3BA (Read)

7..6	5..4	3	2..1	0
Reserved	DU	VR	Reserved	DE

Bit No.	Meaning
7..6	Reserved
5..4	Diagnostic use (DU)
3	Vertical Retrace (VR)
2..1	Reserved
0	Display Enable NOT (DE)

Figure 175. Input status 1 register format

FIELD DEFINITION

Bits	Value	Meaning
5..4	*xx*	Shows color outputs as specified by VSM field of color plane enable register (see figures 177 and 289)
3	0	Video information is being displayed
	1	Vertical retrace interval
0	0	Display is in display mode
	1	Horizontal or vertical retrace interval (real-time status of inverted display enable signal)

Figure 176. Input status 1 register fields

REGISTER DESCRIPTION

The video status mux (VSM) field of the color plane enable register in the attribute controller group controls the color output contents of bits 5 and 4 (see figures 177 and 289).

Bits 3 and 0 are used together to synchronize software updates of the display with the horizontal and vertical retrace periods. Bit 3 can be programmed to interrupt the CPU on IRQ2 using the enable vertical interrupt (EVI) and clear vertical interrupt (CVI) fields of the vertical retrace end register in the CRT controllers group (see figure 244).

VSM	DU	Meaning
00	Bit 5	Color output P2
	Bit 4	Color output P0
01	Bit 5	Color output P5
	Bit 4	Color output P4
10	Bit 5	Color output P3
	Bit 4	Color output P1
11	Bit 5	Color output P7
	Bit 4	Color output P6

Figure 177. Color output contents of diagnostic bits

Bit 3	Bit 0	Display Status
0	0	Display mode
0	1	Horizontal retrace period
1	0	Reserved
1	1	Vertical retrace period

Figure 178. Retrace periods

12.4. General Registers, continued

12.4.6. FEATURE CONTROL REGISTER

The *feature control register* selects the vertical sync output signal sent to the monitor.

REGISTER FORMAT

Feature Control Port 3CA (Read), 3DA/3BA (Write)

7	6	5	4	3	2	1	0
Reserved				VSS	Reserved		

Bit No.	Meaning
7..4	Reserved
3	Vertical Sync Select (VSS)
2..0	Reserved

Figure 179. Feature control register format

FIELD DEFINITION

Bit	Value	Meaning
3	0	Normal vertical sync output
	1	Not allowed

Figure 180. Feature control register fields

REGISTER DESCRIPTION

Bit 3 must be set to 0 (see figure 180).

12.4.7. VGA ENABLE REGISTER

The *VGA enable register* enables and disables the video I/O and memory address decoding. The register at port 3C3 is used only in motherboard systems.

REGISTER FORMAT

VGA Enable (motherboard) Port 3C3

7	6	5	4	3	2	1	0
Reserved							VSE

Bit No.	Meaning
7..1	Reserved
0	Video Subsystem Enable (VSE)

Figure 181. VGA enable register format

FIELD DEFINITION

Bit	Value	Meaning
0	0	Disable video I/O and address decoding
	1	Enable video I/O and address decoding

Figure 182. VGA enable register fields

REGISTER DESCRIPTION

The Power 9100 enable logic for adaptor systems is presented in figure 183 and the enable logic for motherboard systems is presented in figure 184.

46E8 (Hex) bit 4	46E8 (Hex) bit 3	102 (Hex) bit 0	Power 9100
0 *	1 *	1 *	Enable *
* Disable = any other combination			

Figure 183. Power 9100 enable logic for adaptor systems

3C3 (Hex) bit 0	102 (Hex) bit 0	94 (Hex) bit 5	94 (Hex) bit 3	Power 9100
1 *	1 *	1 *	(ignored)	Enable *
* Disable = any other combination				

Figure 184. Power 9100 enable logic for motherboard systems

12.4. General Registers, continued

12.4.8. DAC STATUS REGISTER

The *DAC status register* reflects whether the palette DAC is being read or written.

REGISTER FORMAT

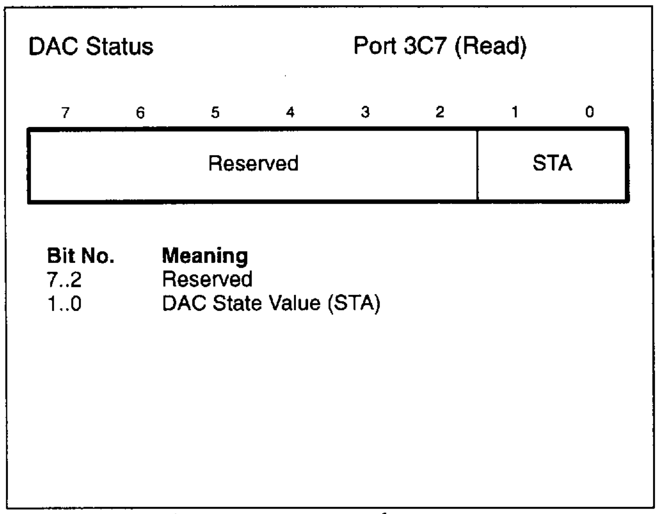


Figure 185. DAC status register format

FIELD DEFINITION

Bits	Value	Operation
1..0	00	Read, PEL address read register accessed last
	01	Not used
	10	Not used
	11	Write, PEL address write register accessed last

Figure 186. DAC status register fields

REGISTER DESCRIPTION

Bits 1 and 0 are located in the palette DAC.

12.4.9. VGA ENABLE REGISTER

The *VGA enable register* enables and disables the video I/O and memory address decoding. The register at port 46E8 is used only in adaptor systems.

REGISTER FORMAT

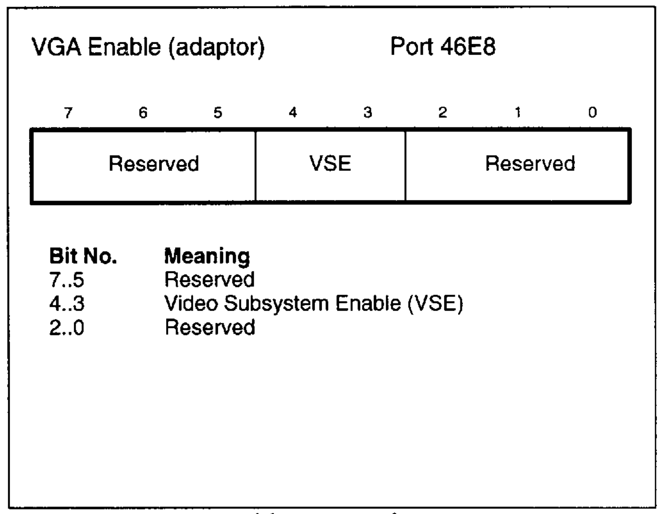


Figure 187. VGA enable register format

FIELD DEFINITION

Bits	Value	Meaning
4..3	00	Disable video I/O and address decoding
	01	Enable video I/O and address decoding
	10	Disable video I/O and address decoding
	11	Disable video I/O and address decoding

Figure 188. VGA enable register fields

REGISTER DESCRIPTION

Port 102 is enabled only when bit 4 of address 46E8 is 1 (for adaptor systems) and when bit 5 of address 94 is 0 (for motherboard systems). Otherwise, port 102 remains locked.

The Power 9100 enable logic for adaptor systems is presented in figure 183 and the enable logic for motherboard systems is presented in figure 184.

12.4. General Registers, continued

12.4.10. POWER 9100 BANK SELECT REGISTER

The *Power 9100 bank select register* provides the most significant four bits of the display memory address.

REGISTER FORMAT

Power 9100 Bank Select   Port 03CD

7	6	5	4	3	2	1	0
Read Bank/Bank 0 [3..0]				Write Bank/Bank 1 [3..0]			

Bit No.	Meaning
7..4	Read Bank/Bank 0 [3..0] (RB0)
3..0	Write Bank/Bank 1 [3..0] (WB1)

Figure 189. Power 9100 bank select register format

FIELD DEFINITION

Bits [7..4] are the *read bank/bank 0 [3..0]* bits. When the host address maps only the "A" segment to the display memory, if bank switching is enabled, these four bits provide the most significant four bits of the 20-bit display memory address during read operations. When both the "A" and "B" segments are mapped, read or write accesses to the "A" segment will use this register to provide the most significant four bits of the display memory address.

Bits [3..0] are the *write bank/bank 1 [3..0]* bits. When the host address maps only the "A" segment to the display memory, if bank switching is enabled, these four bits provide the most significant four bits of the 20-bit display memory address during write operations. When both the "A" and "B" segments are mapped, read or write accesses to the "B" segment will use this register to provide the most significant four bits of the display memory address.

REGISTER DESCRIPTION

Refer to section 12.5.11, the Power 9100 miscellaneous register, for information about the bank switching enable bit. Note that this register is specific to the Power 9100 and is not a standard VGA register.

12.5. Sequencer Registers

The sequencer register group controls the update sequence of display functions. These registers are accessed via the sequence index register port at hex address 3C4 and the sequencer data registers port at hex address 3C5.

12.5.1. SEQUENCER INDEX REGISTER

The *sequencer index register* provides the address index for the sequencer data registers and enables Power 9100 additional registers in the general registers group.

REGISTER FORMAT

Sequencer Index Port 3C4

7	6	5	4	3	2	1	0
Reserved			IHB	RSVD	IB		

Bit No.	Meaning
7..5	Reserved
4	Power 9100 Index High Bit (IHB)
3	Reserved
2..0	Standard VGA Index Bits (IB)

Figure 190. Sequencer index register format

REGISTER DESCRIPTION

The index register is a pointer register which is located at address 03C4 hex. The value loaded in this register determines which sequencer register is accessed when I/O operations are performed by the host to address 03C5 hex. This value is referred to as the index. In addition, the sequencer index register contains a bit which enables additional Power 9100 registers.

Bits [7..5] are reserved.

Bit 3 is reserved.

Bits 4 and 2..0 together address the VGA and Power 9100 extended registers in the sequencer registers group (see figure 191). Bit 4 is the *index high* bit and is reserved in the standard VGA. Bits [2..0] are defined as in the standard VGA.

FIELD DEFINITION

Bit 4	Bits 2..0	Register	Port	Section
0	000	Reset register	3C5	12.5.2
0	001	Clocking mode register	3C5	12.5.3
0	010	Map mask register	3C5	12.5.4
0	011	Character map select register	3C5	12.5.5
0	100	Memory mode register	3C5	12.5.6
0	101	Power 9100 control register 0	3C5	12.5.7
0	110	Power 9100 control register 1	3C5	12.5.8
0	111	Power 9100 revision register	3C5	12.5.9
1	000	Power 9100 ID (read)	3C5	12.5.10
1	001	Power 9100 miscellaneous register	3C5	12.5.11
1	010	Reserved		
1	011	Power 9100 output control register	3C5	12.5.12
1	1*xx*	Reserved		

Figure 191. Index field definition

12.5. Sequencer Registers, continued

12.5.2. RESET REGISTER

The *reset register* resets the VGA controller by causing a clear-and-halt condition to occur.

REGISTER FORMAT

Reset Port 3C5, Seq Index 00

7	6	5	4	3	2	1	0
Reserved						SR	AR

Bit No.	Meaning
7..2	Reserved
1	Synchronous Reset (SR)
0	Asynchronous Reset (AR)

Figure 192. Reset register format

FIELD DEFINITION

Bit 1	Bit 0	Meaning
0	0	Generate and hold the system in reset
0	1	
1	0	
1	1	Release the reset

Figure 193. Reset fields

REGISTER DESCRIPTION

Bit 1 must be 0 before changing the dot clock (DC) or 8/9 dot clock (8/9) fields of the clocking mode register in the sequencer registers group (see figure 195), or the clock select (CS) field of the miscellaneous output register in the general registers group (see figure 172).

Using bit 0 to reset the VGA controller can cause a loss of memory contents.

12.5. Sequencer Registers, continued

12.5.3. CLOCKING MODE REGISTER

The *clocking mode register* configures the sequencer timing circuits.

REGISTER FORMAT

Clocking Mode Port 3C5, Seq Index 01

7	6	5	4	3	2	1	0
DSP13	MPMD	SO	S4	DC	SL	R	8/9

Bit No.	Meaning
7	Power 9100 Disable Serial Planes 1 & 3 (DSP13)
6	Power 9100 Memory Page Mode Disable (MPMD)
5	Screen Off (SO)
4	Shift Four (S4)
3	Dot Clock (DC)
2	Shift Load (SL)
1	Reserved
0	8/9 Dot Clocks (8/9)

Figure 194. Clocking mode register format

FIELD DEFINITION

Bit	Value	Meaning
7	0	Power 9100 input to planes 1 & 3 enabled
	1	Power 9100 input to planes 1 & 3 disabled
6	0	Power 9100 page mode access enabled
	1	Power 9100 page mode access disabled
5	0	Normal screen operation
	1	Video screen off, maximum memory bandwidth assigned to CPU
4	0	Load serializers every character clock
	1	Load serializers every fourth clock
3	0	Set dot clock to master clock frequency
	1	Divide master clock by 2 for dot clock for use with 320 and 360 horizontal pixel modes
2	0	Load serializers every character clock
	1	Load serializers every other clock
0	0	Character clocks are 8 dots wide
	1	Character clocks are 9 dots wide

Figure 195. Clocking mode register fields

REGISTER DESCRIPTION

This memory and video control read/write register is accessed through location 03C5 hex when the index field of the sequencer index register is set to 01. In the standard VGA, bits [6..7] are reserved. In the Power 9100, these bits (**DSP13** and **MPMD**) have specific meanings as follows:

1. The **DSP13** field should be set in overlap modes when graphics planes are chained together to create two long graphics planes. Bit 7 is the *disable serial planes 1 and 3* bit. When this bit is set to 1, the serial input of planes 1 and 3 is disabled.
2. Bit 6 is the *memory page mode disable* bit. When this bit is set to 1, page mode accesses to the frame buffer DRAM/VRAM are disabled. This bit should always be set to zero in the Power 9100.

Bits [5..0] are defined as in the standard VGA.

Bits 4 and 2 together control the loading of the VGA video serializers (see figure 196).

Bit 3 affects all other timings because they are derived from the dot clock.

The horizontal total register in the CRT controller registers group uses bit 0 to specify the character width in pixels for graphics modes (see figure 218).

The reset register synchronous reset (SR) field of the sequencer registers group (see figure 193) must be 0 before changing bits 3 or 0.

Bit 4	Bit 2	Serializer Load	Resolution
0	0	Every character clock	720 dots/line
0	1	Every other clock	360 dots/line
1	0	Every fourth clock	180 dots/line
1	1	Every fourth clock	180 dots/line

Figure 196. Serializer fields

12.5. Sequencer Registers, continued

12.5.4. MAP MASK REGISTER

The *map mask register* enables CPU writing to the four display memory planes.

REGISTER FORMAT

Map Mask — Port 3C5, Seq Index 02

7	6	5	4	3	2	1	0
Reserved				EM3	EM2	EM1	EM0

Bit No.	Meaning
7..4	Reserved
3	Mask Memory Plane 3 (EM3)
2	Mask Memory Plane 2 (EM2)
1	Mask Memory Plane 1 (EM1)
0	Mask Memory Plane 0 (EM0)

Figure 197. Map mask register format

FIELD DEFINITION

Bit	Value	CPU Write Operations
3	0	Memory plane 3 disabled
	1	Memory plane 3 enabled
2	0	Memory plane 2 disabled
	1	Memory plane 2 enabled
1	0	Memory plane 1 disabled
	1	Memory plane 1 enabled
0	0	Memory plane 0 disabled
	1	Memory plane 0 enabled

Figure 198. Map mask register fields

REGISTER DESCRIPTION

The chain four (C4) field of the memory mode register (see figure 204) controls display memory bit plane access. In write modes, the display plane is selected normally by bits 3..0. In read modes, the active bit plane is selected normally by the read map select (RMS) field of the read map select register in the graphics controller registers group (see figure 271).

When bits 3..0 are set to a value of all 1's, the CPU can perform a 32-bit write operation in one memory cycle to enhance scrolling operations. In odd/even modes, bits 1 and 0 should have the same map mask value and bits 3 and 2 should have the same map mask value. All maps should be enabled when chain mode 4 is selected.

12.5. Sequencer Registers, continued

12.5.5. CHARACTER MAP SELECT REGISTER

The *character map select register* permits up to 512 characters to be displayed simultaneously.

REGISTER FORMAT

Character Map Select Port 3C5, Seq Index 03

7 6	5	4	3 2	1 0
Reserved	SAH	SBH	SA	SB

Bit No.	Meaning
7..6	Reserved
5	Select Char Generator A, High Order (SAH)
4	Select Char Generator B, High Order (SBH)
3..2	Select Character Generator A (SA)
1..0	Select Character Generator B (SB)

Figure 199. Character map select register format

FIELD DEFINITION

Bit 5	Bits 3..2	Character Table	Address Offset
0	00	1	0K
0	01	2	16K
0	10	3	32K
0	11	4	48K
1	00	5	8K
1	01	6	24K
1	10	7	40K
1	11	8	56K

Figure 200. SAH and SA fields

Bit 4	Bits 1..0	Character Table	Address Offset
0	00	1	0K
0	01	2	16K
0	10	3	32K
0	11	4	48K
1	00	5	8K
1	01	6	24K
1	10	7	40K
1	11	8	56K

Figure 201. SBH and SB fields

REGISTER DESCRIPTION

Bits 5..0 and attribute byte together determine the appearance of the displayed text (see figure 202). When bits 5 and 3..2 and bits 4 and 1..0 superfields have identical contents, the system uses bit 3 of the attribute byte to select the foreground colors. When these two superfields are different, the system uses bit 3 of the attribute byte to select a character generator.

Condition	Attribute Byte	Selection
Bits 5 and 3..2 = bits 4 and 1..0	Bit 3 = 0	Standard foreground colors
	Bit 3 = 1	Intensified foreground colors
Bits 5 and 3..2 ≠ bits 4 and 1..0	Bit 3 = 0	Character gen B
	Bit 3 = 1	Character gen A

Figure 202. Color and character generator selection

12.5. Sequencer Registers, continued

12.5.6. MEMORY MODE REGISTER

The *memory mode register* controls the way display memory functions.

REGISTER FORMAT

Memory Mode — Port 3C5, Seq Index 04

7	6	5	4	3	2	1	0
Reserved				C4	O/E	EM	R

Bit No.	Meaning
7..4	Reserved
3	Chain Four (C4)
2	Odd/Even (O/E)
1	Extended Memory (EM)
0	Reserved (VGA, Power 9100)

Figure 203. Memory mode register format

FIELD DEFINITION

Bit	Value	Meaning
3	0	Display planes selected by map mask and read map select registers (see figure 198)
	1	Display planes selected by low-order A1 and A0 address bits (see figure 205)
2	0	Odd (maps 1 and 3) and even (maps 0 and 2) addressing mode enabled
	1	Sequential addressing mode enabled
1	0	No extended memory present (display memory less than 64 KB)
	1	Extended memory present (display memory greater than 64 KB)

Figure 204. Memory mode register fields

A1	A0	Map Selected
0	0	0
0	1	1
1	0	2
1	1	3

Figure 205. Memory mode when bit 3 = 1

REGISTER DESCRIPTION

Bit 3 controls display memory bit plane access. In write modes, the display plane is selected normally by the EM3..EM0 fields of the map mask register (see figure 198). In read modes, the active bit plane is selected normally by the read map select (RMS) field of the read map select register in the graphics controller registers group (see figure 271).

Bit 2 of the memory mode register must be the complement of the O/E field of the graphics mode register in the graphics controller registers group (see figure 273).

Bit 0 of the memory mode register is reserved, as in the standard VGA.

12.5. Sequencer Registers, continued

12.5.7. POWER 9100 CONTROL REGISTER 0

The *Power 9100 control register 0* determines character width and the order of memory displays.

The Power 9100 control register 0 must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Control Register 0 — Port 03C5 — Seq Index 05

7	6	5	4	3	2	1	0
DVSAE	BO	OCWM	CPUD2	IL	SVS17	CL	

Bit No.	Meaning
7	Dual Video Start Addr/Offset Enable (DVSAE)
6	Bit Order (BO)
5	Odd Character Width Mode (OCWM)
4	CPU Divide 2 (CPUD2)
3	Interlace (IL)
2	Serial Video Start Address Bit 17 (SVS17)
0–1	Character Length (CL)

Figure 206. Power 9100 control register 0 format

FIELD DEFINITION

Bits [0..1] are the *character width high* bits. These are used in conjunction with the sequencer clocking mode register (SCMR) bit 0 to define the width of a character in pixels according to figure 207.

Bit 2 is *serial video start address* bit 16. This is the most significant bit of the 17-bit serial video start address register. The serial video start address consists of the serial video start address low register (bits [7..0]), the serial video start high register (bits [15..8]) and bit 2 of this register (bit 16). A seventeen bit start address enables the serial video frame buffer to start anywhere within the first 512 K of video memory.

Bit 3 is the *interlace* bit. When this bit is set, interlace mode is used for all modes of operation.

Bit 4 is the *CPU divide* 2 bit. When this bit is set to 1, it enables the CPU divide 2 mode. In this mode, the entire 16–bit CPU memory address is shifted down by one bit so that A[16..1] is mapped to MA[15..0]. A[0] is used to select either maps 0 and 1 or maps 2 and 3. In the Power 9100, MA[16] is provided by host A[17].

Bit 5 is the *odd character width mode* bit. When this bit is clear and the Power 9100 is programmed to generate characters of odd width, then the last pixel of the character is determined by the character code and bit 2 of the attribute mode control register (see figure 286) as in standard VGA. When this bit is set, the last pixel of the character always comes from the font data.

Bit 6 indicates *bit order.* When this bit is reset to 0, bit 7 is displayed first; this is the IBM bit order. When this bit is set to 1, bit 0 is displayed first.

Bit 7 is the *dual video start address/offset enable* bit and is not supported in the Power 9100. It must be set to zero.

Also, when bit 7 is set, the standard horizontal display enable end register displays the total number of graphics pixels in a scan line, and the Power 9100 characters per line register specifies the total number of characters per scan line.

When this bit is 0, only the standard set of CRT controller registers are used.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03C5 hex when the index field of the sequencer index register is 05. After a reset, this register is initialized to a value of 01.

Bit 1	Bit 0	SCMR B0	Character Width (pixels)
0	0	1	6
0	0	0	7
0	1	1	8
0	1	0	9
1	0	1	10
1	0	0	11
1	1	1	12
1	1	0	13

Figure 207. Character length

12.5. Sequencer Registers, continued

12.5.8. POWER 9100 CONTROL REGISTER 1

The *Power 9100 control register 1* provides the display mode and memory plane parameters.

The Power 9100 control register 1 must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Control Register 1 Port 03C5 Seq Index 06

7	6	5	4	3	2	1	0
RTVLI	Res	APME	HBSE	BSE	OPE	Rsvd	DPI

Bit No.	Meaning
7	Real-Time Video Load Inhibit (RTVL1)
6	Reserved
5	Alpha Page Mode Enable (APME)
4	Half Bit Shift Enable (HBSE)
3	Power 9100 Attributes (BSE)
2	Overlapping Planes Enable (OPE)
1	Reserved
0	DRAM Port Inhibit (DPI)

Figure 208. Power 9100 control register 1 format

FIELD DEFINITION

Bit 7 is the *real-time video load inhibit* bit. Reserved; must be set to zero.

Bit 6 is reserved. This bit should be set to 0.

Bit 5 is the *alpha page mode enable* bit. When this bit is set to 1, the memory address bits [4..0] and [12..8] are swapped during a video refresh font data fetch. This allows page-mode accesses to be used by assuring that all font data fetches occurring during a scan line access the same page of dynamic RAM. The system software must compensate for the memory address bit swapping when loading the fonts. Note that the memory controller is designed to use page mode whenever possible.

Bit 4 is the *half-bit shift enable* bit. When this bit is set to 1, bit 14 of the character font indicates half-bit shift and causes the character pixel row to be shifted half a pixel to the right.

Bit 3 is the *Power 9100 attributes* bit. When this bit is set to 1, the special Power 9100 attributes are used. The attribute byte is defined according to the figure 209.

Bit 2 is the *overlapping planes enable* bit. Reserved; must be set to zero.

Bit 1 is reserved.

Bit 0 is the *DRAM port inhibit* bit. Reserved; must be set to zero.

REGISTER DESCRIPTION

This video control read/write register is accessed though location 03C5 hex when the index field of the sequencer index register is set to 06. After a reset, this register is initialized to a value of 01. The BSE field enables the Power 9100 struck through position (STP) field of the cursor start register in the CRT controller registers group (see figure 235).

Bit	Attribute
0	Color 0 (Half Bright in monochrome)
1	Underline
2	Reverse Video
3	Blinking
4	Bold
5	Struck Through
6	Color 1
7	Color 2

Figure 209. Power 9100 Attributes

12.5. Sequencer Registers, continued

12.5.9. POWER 9100 REVISION REGISTER

The *Power 9100 revision register* provides device identification and revision level for the SVGA portion of the Power 9100.

REGISTER FORMAT

Power 9100 Revision Port 03C5 Seq Index 07

7	6	5	4	3	2	1	0
Device ID			Reserved		Revision Level		

Bit No.	Meaning
7..5	Device ID (DID)
4..3	Reserved
2..0	Revision Level (RL)

Figure 210. Power 9100 revision register format

FIELD DEFINITION

Bits [7..5] are the *device id* bits. These bits identify the device. This field is 010 for the Power 9100

Bits [4..3] are reserved for WEITEK use. This field may contain any combination of bits, as determined by WEITEK.

Bits [2..0] are the *revision level* bits. These bits identify the revision level of the device. This field is reserved for WEITEK use, and may contain any combination of bits, as determined by WEITEK.

REGISTER DESCRIPTION

This host I/O read-only register is accessed through location 03C5 hex when the index field of the sequencer index register is 07 hex. Note that this register is specific to the Power 9100 and is not a standard VGA register.

12.5.10. POWER 9100 ID REGISTER

The *Power 9100 ID register* is a read-only register.

Power 9100 ID Register Port 03C5 (Read) Seq Index 10

7	6	5	4	3	2	1	0
0	1	0	1	Reserved			

Bit No.	Meaning
7..4	ID Bits
6..0	Reserved

Figure 211. ID register

This read-only register is accessed through location 03C5 hex when the index field of the sequencer address register is 10 hex. Note that this register is specific to the Power 9100 and is not a standard VGA register.

Bits [7..4] are always 0101.

Bits [3..0] are reserved.

12.5.11. POWER 9100 MISCELLANEOUS REGISTER

The *Power 9100 miscellaneous register* controls the host I/O read/write interface.

The Power 9100 miscellaneous register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Miscellaneous — Port 03C5, Seq Index 11

7	6	5	4	3	2	1	0
1	BSE	CRL	1	1	BRE	1	0

Bit No.	Meaning
7	This field must always be set to a 1.
6	Bank Switching Enable (BSE)
5	Power 9100 Control Register Lock (CRL)
4	This field must always be set to a 1.
3	This field must always be set to a 1.
2	BIOS ROM Enable (BRE)
1	Reserved, set to a 1.
0	Reserved, set to a 0.

Figure 212. Power 9100 miscellaneous register format

FIELD DEFINITION

Bit 6 is the *bank switching enable* bit. When this bit is reset to 0 and the address mapping is "A" segment or "A and B" segments, the four most-significant bits of the 20-bit frame buffer address are provided by the bank select register, as defined in the bank select register description. When this bit is set to 1, address mapping is disabled.

Bit 5 is the Power 9100 *control register lock* bit. When set to 1, the Power 9100 extended registers (except for the Power 9100 revision register), as shown in figure 213, are read and write protected. When this bit is 0, these registers can be accessed. Upon reset, this bit is set to 1. To unlock the Power 9100 miscellaneous register, see section 12.1.1.

Bit 2 is the *BIOS ROM enable* bit. When this bit is reset to 0, the external BIOS ROM is disabled. When this bit is set to 1, the external BIOS ROM is enabled and responds to addresses xC0000 hex to xC7FFF hex. On reset, this bit is equal to 1.

REGISTER DESCRIPTION

This host I/O read/write register is accessed through location 03C5 hex when the index field of the sequencer index register is 11 hex. This register is only accessible at this location on the Power 9100. Note that this register is specific to the Power 9100 and is not a standard VGA register. It is set to FF hex by RESET.

Name	Port (Hex)	Index (Hex)
Power 9100 Control Register 0	03C5	05
Power 9100 Control Register 1	03C5	06
BIOS ROM Status	03C5 (read)	10
Power 9100 Miscellaneous	03C5	11
Power 9100 Output Control	03C5	12

Figure 213. Registers affected by control register lock bit

12.5. Sequencer Registers, continued

12.5.12. POWER 9100 OUTPUT CONTROL REGISTER

The Power 9100 output control register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100
Output Control

Port 03C5, Seq Index 12

7	6	5	4	3	2	1	0
User Defined		CKSEL [2]	User Defined				

Bit No.	Meaning
7..6	User Defined (UD)
5	CKSEL[2] in emulation mode
4..0	User defined (UD)

Figure 214. Power 9100 output control register format

FIELD DEFINITION

This register is implemented on-chip for purposes of read back only.

Bits[7..6] are user defined.

Bit 5 is CKSEL[2] in emulation mode and is used to support programmable clock synthesizers. To control CKSEL[2..0] in native mode, see section 10.2. See section 12.4.3 for CKSEL[1..0].

Bits [4..0] are user defined.

REGISTER DESCRIPTION

This host I/O read/write register is accessed through location 03C5 hex when the index field of the sequencer index register is 12 hex. Note that this register is specific to the Power 9100 and is not a standard VGA register. This register is only accessible at this location on the Power 9100.

12.6. CRT Controller Registers

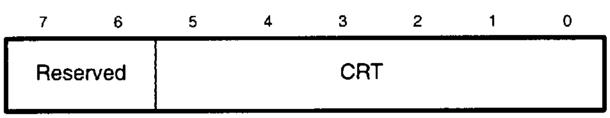
The CRT controller register group define the raster display. In color systems, these registers are accessed via the CRT controller index register port at hex address 3D4 and the CRT controller data registers port at hex address 3D5. In monochrome systems, these register ports are at hex address 3B4 and hex address 3B5, respectively.

12.6.1. CRT CONTROLLER INDEX REGISTER

The *CRT controller index register* provides the address index for the CRT controller registers.

REGISTER FORMAT

CRT Controller Index   Port 3D4/3B4



Bit No.	Meaning
7..6	Reserved
5..0	CRT Controller Register Index (CRT)

Figure 215. CRT controller index register format

REGISTER DESCRIPTION

The input/output address (IOA) field of the miscellaneous output register of the general registers group (see figure 172) determines whether the CRT controller index register is located at address 3B4 (monochrome emulation) or 3D4 (color emulation).

Note that in the VGA specification, bit 5 is specified as a chip test bit that must be set to zero. In order to access the VGA hidden register at index 24, this bit must be set to one.

FIELD DEFINITION

Field	Register	Section
00	Horizontal total register	12.6.2
01	Horizontal display enable end register	12.6.3
02	Start horizontal blanking register	12.6.4
03	End horizontal blanking register	12.6.5
04	Start horizontal retrace pulse register	12.6.6
05	End horizontal retrace register	12.6.7
06	Vertical total register	12.6.8
07	Overflow register	12.6.9
08	Preset row scan register	12.6.10
09	Maximum scan line register	12.6.11
0A	Cursor start register	12.6.12
0B	Cursor end register	12.6.13
0C	Start address high register	12.6.14
0D	Start address low register	12.6.15
0E	Cursor location high register	12.6.16
0F	Cursor location low register	12.6.17
10	Vertical retrace start register	12.6.18
11	Vertical retrace end register	12.6.19
12	Vertical display enable end register	12.6.20
13	Offset register	12.6.21
14	Underline location register	12.6.22
15	Start vertical blank register	12.6.23
16	End vertical blank register	12.6.24
17	CRT mode control register	12.6.25
18	Line compare register	12.6.26
19	Power 9100 interlace register	12.6.27
1A	Power 9100 serial start address high	12.6.28
1B	Power 9100 serial start address low register	12.6.29
1C	Power 9100 serial offset register	12.6.30
1D	Power 9100 total characters per line register	12.6.31
24	Power 9100 attributes state	

Figure 216. Bits 4..0 field definition

12.6. CRT Controller Registers, continued

12.6.2. HORIZONTAL TOTAL REGISTER

The *horizontal total register* determines the total horizontal scan time, including active display and retrace.

REGISTER FORMAT

Horizontal Total — Port 3D5/3B5, CRT Index 00

7	6	5	4	3	2	1	C
HT							

Bit No.	Meaning
7..0	Horizontal Total (HT)

Figure 217. Horizontal total register format

FIELD DEFINITION

The value loaded into the HT field is the total horizontal character count per line minus 5.

$$HT = horizontal\_character\_total - 5$$

REGISTER DESCRIPTION

The horizontal total field and the horizontal character counter internal to the CRT controller together determine the total horizontal scan time (see figure 218). All horizontal and vertical timing is based on character clock inputs to this register.

Mode	Horizontal Character Counter
Text	Incremented after each character is output to the screen
	Reset after reaching value in horizontal total register
Graphics	Incremented every 8 or 9 pixels as specified in 8/9 field of clocking mode register (see figure 195)
	Reset after reaching value in horizontal total register

Figure 218. Determining horizontal scan time

12.6.3. HORIZONTAL DISPLAY ENABLE END REGISTER

The *horizontal display enable end register* determines the number of characters on a horizontal line.

REGISTER FORMAT

Horizontal Display Enable End — Port 3D5/3B5, CRT Index 01

7	6	5	4	3	2	1	0
HDE							

Bit No.	Meaning
7..0	Horizontal Display Enable (HDE)

Figure 219. Horizontal display enable end register format

FIELD DEFINITION

The value loaded into the horizontal display enable (HDE) field is the number of displayed characters on a horizontal scan line minus one.

$$HDE = number\_of\_displayed\_characters - 1$$

REGISTER DESCRIPTION

The horizontal display enable field defines the length of the horizontal display enable signal.

12.6. CRT Controller Registers, continued

12.6.4. START HORIZONTAL BLANKING REGISTER

The *start horizontal blanking register* determines the start of the horizontal blanking period.

REGISTER FORMAT

Start Horizontal Blanking — Port 3D5/3B5, CRT Index 02

7	6	5	4	3	2	1	0
SHB							

Bit No.	Meaning
7..0	Start Horizontal Blanking (SHB)

Figure 220. Start Horizontal Blanking Register

FIELD DEFINITION

The value loaded into the start horizontal blanking field is the value of the horizontal character counter at the time the horizontal blanking period should begin.

REGISTER DESCRIPTION

The horizontal blanking signal stops the display of data during horizontal CRT refresh. The start horizontal blanking field is used by the end horizontal blanking (EHB) field of the end horizontal blanking register (see figure 222).

12.6.5. END HORIZONTAL BLANKING REGISTER

The *end horizontal blanking register* determines the end of the horizontal blanking period.

REGISTER FORMAT

End Horizontal Blanking — Port 3D5/3B5, CRT Index 03

7	6	5	4	3	2	1	0
1	DES		EHB				

Bit No.	Meaning
7	Chip Testing Bit (must be set to 1)
6..5	Display Enable Skew (DES)
4..0	End Horizontal Blanking (EHB)

Figure 221. End Horizontal Blanking Register

FIELD DEFINITION

Bits	Value	Meaning
6..5	00	Character clock skew = 0
	01	Character clock skew = 1
	10	Character clock skew = 2
	11	Character clock skew = 3
4..0	*xxxxx*	End of horizontal blanking period

Figure 222. End horizontal blanking register fields

REGISTER DESCRIPTION

Bits 6 and 5 contain the six least-significant-bit value of the horizontal character counter when the horizontal blanking interval is to become inactive according to the following formula:

$$EHB = SHB + blanking\_signal\_width$$

The start horizontal blanking (SHB) field is located in the start horizontal blanking register (see figure 220) and the blanking signal width is in character clock units. A sixth end horizontal blanking bit is located in the EHB field of the end horizontal retrace register (see figure 225).

12.6. CRT Controller Registers, continued

12.6.6. START HORIZONTAL RETRACE PULSE REGISTER

The *start horizontal retrace pulse register* determines the start of the horizontal retrace period.

REGISTER FORMAT

Start Horizontal Retrace Pulse — Port 3D5/3B5, CRT Index 04

7	6	5	4	3	2	1	0
SHR							

Bit No.	Meaning
7..0	Start Horizontal Retrace Pulse (SHR)

Figure 223. Start horizontal retrace pulse register format

FIELD DEFINITION

The value loaded into the start horizontal retrace field is the value of the horizontal character counter at the time the horizontal retrace period should begin.

REGISTER DESCRIPTION

The start horizontal retrace field is used to center the screen horizontally and is used by the end horizontal retrace (EHR) field of the end horizontal retrace register (see figure 225).

12.6.7. END HORIZONTAL RETRACE REGISTER

The *end horizontal retrace register* determines the end of the horizontal retrace period.

REGISTER FORMAT

End Horizontal Retrace register — Port 3D5/3B5, CRT Index 05

7	6	5	4	3	2	1	0
EHB	HRD		EHR				

Bit No.	Meaning
7	End Horizontal Blanking (EHB)
6..5	Horizontal Retrace Delay (HRD)
4..0	End Horizontal Retrace (EHR)

Figure 224. End horizontal retrace register format

FIELD DEFINITION

Bits	Value	Meaning
7	*x*	The sixth bit the the EHB field in the end horizontal blanking register (see figure 222)
6..5	00	Horizontal retrace delay = 0
	01	Horizontal retrace delay = 1
	10	Horizontal retrace delay = 2
	11	Horizontal retrace delay = 3
4..0	*xxxxx*	End of horizontal retrace period

Figure 225. End horizontal retrace register fields

REGISTER DESCRIPTION

Bits 4..0 contain the five least-significant-bit value of the horizontal character counter when the horizontal retrace signal is to become inactive according to the formula:

$$EHR = SHR + retrace\_signal\_width$$

The start horizontal retrace (SHR) field is located in the start horizontal retrace pulse register (see figure 223) and the retrace signal width is in character clock units. A sixth end horizontal retrace bit is located in the EHB field of the end horizontal retrace register (see figure 225).

12.6. CRT Controller Registers, continued

12.6.8. VERTICAL TOTAL REGISTER

The *vertical total register* determines the total number of vertical scan lines on the monitor.

REGISTER FORMAT

Vertical Total   Port 3D5/3B5, CRT Index 06

7	6	5	4	3	2	1	0
VT							

Bit No.	Meaning
7..0	Vertical Total (VT)

Figure 226. Vertical total register format

FIELD DEFINITION

The value loaded into the vertical total field is the total horizontal line count per screen minus two.

$$VT = vertical\_retrace + horizontal\_scan\_lines - 2$$

REGISTER DESCRIPTION

The ninth and tenth vertical total bits are located in the VT and VT1 fields of the overflow scan register (see figure 228).

12.6.9. OVERFLOW REGISTER

The *overflow register* supplies the higher-order bits for several of the CRT controller registers.

REGISTER FORMAT

Overflow   Port 3D5/3B5, CRT Index 07

7	6	5	4	3	2	1	0
VRS	VDE	VT1	LC	VBS	VRS	VDE	VT

Bit No.	Meaning
7	Vertical Retrace Start Bit 9 (VRS)
6	Vertical Display Enable End Bit 9 (VDE)
5	Vertical Total Bit 9 (VT1)
4	Line Compare Bit 8 (LC)
3	Start Vertical Blanking Bit 8 (VBS)
2	Vertical Retrace Start Bit 8 (VRS)
1	Vertical Display Enable End Bit 8 (VDE)
0	Vertical Total Bit 8 (VT)

Figure 227. Overflow register format

FIELD DEFINITION

Bit	Meaning	Section
7	The tenth VRS bit of the vertical retrace start register (the ninth VRS bit is in field VRS, bit 2)	12.6.18
6	The tenth VDE bit of the vertical display end enable register (the ninth bit is in field VDE, bit 1)	12.6.20
5	The tenth VT bit of the vertical total register (the ninth VT bit is in field VT, bit 0)	12.6.8
4	The ninth LC bit of the line compare register	12.6.26
3	The ninth VBS bit of the start vertical blanking register	12.6.23
2	The ninth VRS bit of the vertical retrace start register (the tenth VRS bit is in field VRS, bit 7)	12.6.18
1	The ninth VDE bit of the vertical display end enable register (the tenth VDE bit is in field VDE, bit 6)	12.6.20
0	The ninth VT bit of the vertical total register (the tenth VT bit is in field VT1, bit 5)	12.6.8

Figure 228. Overflow register fields

12.6. CRT Controller Registers, continued

12.6.10. PRESET ROW SCAN REGISTER

The *preset row scan register* controls scrolling and panning operations.

REGISTER FORMAT

Preset Row Scan — Port 3D5/3B5, CRT Index 08

7	6	5	4	3	2	1	0
SVAWC	BP		PRS				

Bit No.	Meaning
7	Power 9100 Serial Video Address Wrap Control (SVAWC)
6..5	Byte Panning (BP)
4..0	Preset Row Scan (PRS)

Figure 229. Preset row scan register format

FIELD DEFINITION

Bit 7 is the *serial video address wrap control* bit. When this bit is reset to 0, the serial video RAM address counter wraps to 0 after a count of 64K. This ensures compatibility with IBM 256K VGA implementation. When this bit is set to 1, the counter wraps after address 128K, enabling access to the maximum 512K memory.

Bits 6 and 5 control the number of bytes to pan, and the horizontal pixel pan (HPP) field in the horizontal pixel panning register of the attribute controller registers group determines the number of pixels to pan (see section 12.8.6). The value of bits 6..5 is normally 0 because there are currently no modes programmed for multiple-shift operation.

The value of bits 4..0 are normally 0 because the entire vertical extent of the top character row is usually displayed.

The preset row scan register and maximum scan line register (see section 12.6.11) together specify the size of the character box. The CE field of the cursor end register (see figure 237) in conjunction with the cursor end (CS) field of the cursor start register (see figure 235) specify the cursor location within this box.

The pixel panning compatibility (PPC) field of the attribute mode control register of the attribute controller registers group (see figure 286) allows line compare (see figure 253) to affect horizontal pixel panning and preset row scan register outputs (see figures 290).

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex (monochrome mode) or 03D5 hex (color mode) when the index field of the CRT controller index register is 08.

Bits	Value	Meaning
7	0	Serial video RAM address counter wraps to 0 after a count of 64K
	1	Serial video RAM address counter wraps to 0 after address 128K
6..5	*xx*	Specifies number of bytes to pan
4..0	*xxxxx*	The starting row of a character box displayed on the top character row

Figure 230. Preset row scan register fields

12.6. CRT Controller Registers, continued

12.6.11. MAXIMUM SCAN LINE REGISTER

The *maximum scan line register* specifies the number of scan lines per character.

REGISTER FORMAT

Maximum Scan Line   Port 3D5/3B5, CRT Index 09

7	6	5	4	3	2	1	0
2T4	LC	VBS	MSL				

Bit No.	Meaning
7	200–to–400–line Conversion (2T4)
6	Line Compare Bit 9 (LC)
5	Start Vertical Blanking Bit 9 (VBS)
4..0	Maximum Line Scan (MSL)

Figure 231. Maximum scan line register format

FIELD DEFINITION

Bits	Value	Meaning
7	0	The row scan counter clock is equal to the horizontal scan rate and line doubling is not enabled
	1	Allows a 200-line mode to be displayed on 400 display scan lines by dividing the row scan counter clock by 2 to duplicate each line twice
6	*x*	The tenth LC bit of the line compare register (see section 12.6.26)
5	*x*	The tenth VBS bit of the start vertical blanking register (see section 12.6.23)
4..0	*xxxxx*	The number of scan lines in a character minus 1 to specify the number of scan line characters per row

Figure 232. Maximum scan line register fields

REGISTER DESCRIPTION

The preset row scan register (see section 12.6.10) and maximum scan line register together specify the size of the character box.

12.6. CRT Controller Registers, continued

12.6.12. CURSOR START REGISTER

The *cursor start register* determines the first line of the character box that's part of the cursor.

REGISTER FORMAT

Cursor Start — Port 3D5/3B5, CRT Index 0A

7	6	5	4	3	2	1	0
STP		COO	CS				

Bit No.	Meaning
7..6	Power 9100 Struck Through Position (STP)
5	Cursor On/OFF (COO)
4..0	Row Scan Cursor Start (CS)

Figure 233. Cursor start register format

FIELD DEFINITION

Bits 7..6 designate the *struck through position.* When the Power 9100 attributes are used, these bits specify which character line is struck through according to figure 234.

Bits 5..0 are defined as in the standard VGA.

Struck through position [1..2]	Line number
0	1
1	5
2	9
3	13

Figure 234. Struck through position

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex (monochrome mode) or 03D5 hex (color mode) when the index field of the CRT controller address is 0A hex.

The preset row scan register (see section 12.6.10) and maximum scan line register (see section 12.6.11) together specify the size of the character box.

Bits 4..0 in conjunction with the cursor end (CE) field of the cursor end register (see figure 237) specify the cursor location in the character box. No cursor is generated when the value in bits 4..0 exceeds the value in the CE field.

Bits 7..6 are enabled by the Power 9100 attributes (BSE) field of the Power 9100 control register 1 of the sequencer registers group (see figure 208).

Bits	Value	Meaning
7..6	00	Line 1 struck through (Power 9100 attributes)
	01	Line 5 struck through (Power 9100 attributes)
	10	Line 9 struck through (Power 9100 attributes)
	11	Line 13 struck through (Power 9100 attributes)
5	0	Cursor turned off
	1	Cursor turned on
4..0	*xxxxx*	First scan line in character box representing start of cursor

Figure 235. Cursor start register fields

12.6. CRT Controller Registers, continued

12.6.13. CURSOR END REGISTER

The *cursor end register* determines the last scan line of the character box that's part of the cursor.

REGISTER FORMAT

Cursor End — Port 3D5/3B5, CRT Index 0B

7	6	5	4	3	2	1	0
R	CSK		CE				

Bit No.	Meaning
7	Reserved
6..5	Cursor Skew (CSK)
4..0	Cursor End (CE)

Figure 236. Cursor end register format

FIELD DEFINITION

Bits	Value	Meaning
6..5	00	Clock skew = 0 characters
	01	Clock skew = 1 character
	01	Clock skew = 2 characters
	11	Clock skew = 3 characters
C4..0	*xxxxx*	Bottom scan line of character row for cursor display

Figure 237. Cursor end register fields

REGISTER DESCRIPTION

The preset row scan register (see section 12.6.10) and maximum scan line register (see section 12.6.11) together specify the size of the character box.

Bits 6..5 specify the number of characters to delay cursor data for proper synchronization.

Bits 4..0 in conjunction with the cursor end (CS) field of the cursor start register (see figure 235) specify the cursor location in the character box. No cursor is generated when the value in the CS field exceeds the value in bits 4..0.

12.6.14. START ADDRESS HIGH REGISTER

The *start address high register*, in conjunction with the start address low register, points to the origin of the display data in the display buffer memory.

REGISTER FORMAT

Start Address High — Port 3D5/3B5, CRT Index 0C

7	6	5	4	3	2	1	0
SAH							

Bit No.	Meaning
7..0	Start Address High (SAH)

Figure 238. Start address high register format

FIELD DEFINITION

The value loaded into the start address high field is the high-order 8 bits of the start address. The low-order 8 bits are located in the start address low (SAL) field of the start address low register (see figure 239).

REGISTER DESCRIPTION

The 16-bit value obtained from the start address high and start address low (see figure 239) registers is the first address after the vertical retrace on each screen refresh.

12.6. CRT Controller Registers, continued

12.6.15. START ADDRESS LOW REGISTER

The *start address low register*, in conjunction with the start address high register, points to the origin of the display data in the display buffer memory.

REGISTER FORMAT

Start Address Low — Port 3D5/3B5, CRT Index 0D

7	6	5	4	3	2	1	0
SAL							

Bit No.	Meaning
7..0	Start Address Low (SAL)

Figure 239. Start address low register format

FIELD DEFINITION

The value loaded into the start address low field is the low-order 8 bits of the start address. The high-order 8 bits are located in the start address high (SAH) field of the start address high register (see figure 238).

REGISTER DESCRIPTION

The 16-bit value obtained from the start address high (see figure 238) and start address low registers is the first address after the vertical retrace on each screen refresh.

12.6.16. CURSOR LOCATION HIGH REGISTER

The *cursor location high register*, in conjunction with the cursor location low register, points to the cursor position in the display buffer memory.

REGISTER FORMAT

Cursor Location High — Port 3D5/3B5, CRT Index 0E

7	6	5	4	3	2	1	0
CLH							

Bit No.	Meaning
7..0	Cursor Location High (CLH)

Figure 240. Cursor location high register format

FIELD DEFINITION

The value loaded into the cursor location high field is the high-order 8 bits of the cursor location. The low-order 8 bits are located in the cursor location low (CLL) field of the cursor location low register (see figure 241).

12.6. CRT Controller Registers, continued

12.6.17. CURSOR LOCATION LOW REGISTER

The *cursor location low register*, in conjunction with the cursor location high register, points to the cursor position in the display buffer memory.

REGISTER FORMAT

Cursor Location Low   Port 3D5/3B5, CRT Index 0F

7	6	5	4	3	2	1	0
CLL							

Bit No.	**Meaning**
7..0	Cursor Location Low (CLL)

Figure 241. Cursor location low register format

FIELD DEFINITION

The value loaded into the cursor location low field is the low-order 8 bits of the cursor location. The high-order 8 bits are located in the cursor location high (CLH) field of the cursor location high register (see figure 240).

12.6.18. VERTICAL RETRACE START REGISTER

The *vertical retrace start register* determines the start of the vertical retrace period.

REGISTER FORMAT

Vertical Retrace Start   Port 3D5/3B5, CRT Index 10

7	6	5	4	3	2	1	0
VRS							

Bit No.	**Meaning**
7..0	Vertical Retrace Start (VRS)

Figure 242. Vertical retrace start register format

FIELD DEFINITION

The value loaded into the vertical retrace start field is the value of the scan line counter at the time when the vertical retrace period should begin.

REGISTER DESCRIPTION

Access to the vertical retrace start and vertical retrace end (see figure 243) registers is enabled by the end horizontal blanking register compatibility read (CR) field in the CRT controller registers group (see figure 222).

The ninth and tenth vertical retrace start bits are located in the VRS fields of the overflow register (see figure 228).

12.6. CRT Controller Registers, continued

12.6.19. VERTICAL RETRACE END REGISTER

The *vertical retrace send register* determines the end of the vertical retrace period.

REGISTER FORMAT

Vertical Retrace End   Port 3D5/3B5, CRT Index 11

7	6	5	4	3	2	1	0
PR	BW	EVI	CVI	EVR			

Bit No.	Meaning
7	Protect Registers 0–7 (PR)
6	Select 5 Refresh Cycles (BW)
5	Enable Vertical Interrupt (EVI)
4	Clear Vertical Interrupt (CVI)
3..0	Vertical Retrace End (EVR)

Figure 243. Vertical retrace end register format

FIELD DEFINITION

Bits	Value	Meaning
7	0	Protection disabled (writing enabled)
	1	Writing from host to CRTC registers disabled
6	0	Select three DRAM cycles
	1	Select five DRAM cycles for 15.75 MHz sweep-rate displays
5	0	Vertical retrace interrupt on IRQ2 enabled and latched in VR field of input status 0 register (see figure 176)
	1	Vertical retrace interrupt disabled
4	0	Vertical retrace interrupt cleared by interrupt handler, then set to 1 to avoid holding vertical retrace interrupts inactive
	1	Enable additional vertical retrace interrupts
3..0	*xxxx*	End of vertical retrace period

Figure 244. Vertical retrace end register fields

REGISTER DESCRIPTION

Access to the vertical retrace start (see figure 242) and vertical retrace end registers is enabled by the end horizontal blanking register compatibility read (CR) field in the CRT controller registers group (see figure 222).

The vertical retrace (VR) field of the the input status 0 register in the general registers group (see figure 176) can be programmed to interrupt the CPU using bit 5 and bit 4 when the IRQ2 interrupt level is shared.

The value programmed into bits 3..0 is based on the vertical retrace start (VRS) field of the vertical retrace start register (see figure 242) and the width of the vertical retrace signal in horizontal scan units as follows:

$$EVR = VRS + vertical\_retrace\_signal\_width$$

12.6.20. VERTICAL DISPLAY ENABLE END REGISTER

The *vertical display enable end register* specifies the last horizontal scan line displayed at the bottom of the screen.

REGISTER FORMAT

Vertical Display Enable End — Port 3D5/3B5, CRT Index 12

7	6	5	4	3	2	1	0
VDE							

Bit No.	Meaning
7..0	Vertical Display End Enable (VDE)

Figure 245. Vertical display enable end register format

FIELD DEFINITION

The value loaded into the vertical display end field is the number of the last horizontal scan line at the bottom of the screen.

REGISTER DESCRIPTION

Vertical blanking normally occurs before the value in the vertical display enable register is reached.

The ninth and tenth vertical display end bits are located in the VDE fields of the overflow register (see figure 228).

The border area includes the horizontal lines occurring between the value in the start vertical blanking register (see section 12.6.23) and the value in the vertical display enable register. The overscan color register in the attribute controller registers group determines border color (see section 12.8.4).

12.6.21. OFFSET REGISTER

The *offset register* specifies the width of the display.

REGISTER FORMAT

Offset — Port 3D5/3B5, CRT Index 13

7	6	5	4	3	2	1	0
OFF							

Bit No.	Meaning
7..0	Logical Line Width of Screen (OFF)

Figure 246. Offset register format

FIELD DEFINITION

The value loaded into the logical line width field is the difference in bytes or words between the addresses of vertically adjacent scan lines.

$$next\_row\_address = current\_byte\_start\_addr + off * K$$

where:

K = 2, byte addressing

K = 4, word addressing

REGISTER DESCRIPTION

The count by two (CBT) field of the CRTC mode control register specifies VGA addressing in either byte mode or word mode (see figure 252).

12.6. CRT Controller Registers, continued

12.6.22. UNDERLINE LOCATION REGISTER

The *underline location register* determines the horizontal line used for underlining characters.

REGISTER FORMAT

Underline Location   Port 3D5/3B5, CRT Index 14

7	6	5	4	3	2	1	0
SVLD	DW	CB4	UL				

Bit No.	Meaning
7	Power 9100 Serial Video Line Doubling (SVLD)
6	Double Word Mode (DW)
5	Count by Four (CB4)
4..0	Underline Location (UL)

Figure 247. Underline location register format

FIELD DEFINITION

Bit 7 is the *serial video line doubling* bit. Reserved; must be set to zero.

Bits [6..0] are defined as in the standard VGA.

Bits	Value	Meaning
7	0	Single-scan lines accessed through VRAM serial port (not implemented)
	1	Double-scan lines accessed through VRAM serial port (not implemented)
6	0	Normal word addressing
	1	Double word addressing
5	0	Normal clocking
	1	Divide character clock to memory address counter by 4
4..0	*xxxxx*	Underline at character box horizontal scan line minus 1

Figure 248. Underline location register fields

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex (monochrome mode) or 03D5 hex (color mode) when the index field of the CRT controller index register is 14 hex.

The number of scan lines per character is specified by the maximum scan line register (see section 12.6.11).

When bit 6 field is 0, the word/byte (W/B) field of the CRTC mode control register (see figure 252) controls the addressing; and when bit 6 is 1, the addressing is shifted by two bits.

12.6.23. START VERTICAL BLANKING REGISTER

The *start vertical blanking register* determines the start of the vertical blanking period.

REGISTER FORMAT

Start Vertical Blanking	Port 3D5/3B5, CRT Index 15

7	6	5	4	3	2	1	0
VBS							

Bit No.	Meaning
7..0	Vertical Blank Start (VBS)

Figure 249. Start vertical blanking register format

FIELD DEFINITION

The value loaded into the vertical blanking start field is the value of the scan line counter at the time when the vertical blanking period should begin.

REGISTER DESCRIPTION

A ninth vertical blanking start bit is located in the VBS field of the overflow register (see figure 228), and a tenth vertical blanking start bit is located in the VBS field of the maximum scan line register (see figure 232).

The border area includes the horizontal lines occurring between the value in the start vertical blanking register and the value in the vertical display enable register (see section 12.6.20). The overscan color register in the attribute controller registers group determines border color (see section 12.8.4).

12.6.24. END VERTICAL BLANKING REGISTER

The *end vertical blanking register* determines the end of the vertical blanking period.

REGISTER FORMAT

End Vertical Blanking	Port 3D5/3B5, CRT Index 16

7	6	5	4	3	2	1	0
VBE							

Bit No.	Meaning
7..0	End Vertical Blanking (VBE)

Figure 250. End vertical blanking register format

FIELD DEFINITION

The value loaded into the end vertical blanking field is the horizontal scan count value when the vertical output signal becomes inactive, and is determined from the value in the SVB field of the start vertical blanking field (see figure 249) and the width of the vertical blank signal in horizontal scan units as follows:

$$VBE = (SVB - 1) + vertical\_blanking\_signal$$

12.6. CRT Controller Registers, continued

12.6.25. CRTC MODE CONTROL REGISTER

The *CRTC mode control register* assists in display control.

REGISTER FORMAT

CRTC Mode Control   Port 3D5/3B5, CRT Index 17

7	6	5	4	3	2	1	0
HR	W/B	AW	R	CBT	HRS	SRS	CMS

Bit No.	Meaning
7	Hardware Reset (HR)
6	Word/Byte Mode (W/B)
5	Address Wrap (AW)
4	Reserved
3	Count by Two (CBT)
2	Horizontal Retrace Select (HRS)
1	Select Slow Scan Counter (SRS)
0	Compatibility Mode Support (CMS)

Figure 251. CRTC mode control register format

REGISTER DESCRIPTION

When the double word mode (WD) field of the underline location register (see figure 248) is 0, bit 6 controls the addressing; and when the WD field of the underline location register is 1, the addressing is shifted by two bits.

Bit 3 creates either a byte or word refresh address for the display buffer in conjunction with the offset register (see section 12.6.21).

FIELD DEFINITION

Bit	Value	Meaning
7	0	Horizontal and vertical retrace cleared
	1	Horizontal and vertical retrace enabled
6	0	Word mode selected (MSB output on LSB address line depends on AW field)
	1	Byte mode selected
5	0	Address bit 13 sent as LSB to display memory in byte address mode and address bit 0 sent as LSB to display memory in word address mode
	1	Address bit 15 sent as LSB to display memory
3	0	Memory address counter clocked with character clock input
	1	Memory address counter clocked every other character clock input
2	0	Scan line counter clocked every horizontal retrace
	1	Scan line counter clocked every other horizontal retrace
1	0	Row scan counter bit 1 placed on memory address bus bit 14 during active display time
	1	Sequential output of memory addresses
0	0	Substitute row scan address bit 0 for memory address bit 13
	1	No substitution (memory address output bit 13 signal of CRT controller is memory address bit 13)

Figure 252. CRTC mode control register fields

12.6. CRT Controller Registers, continued

12.6.26. LINE COMPARE REGISTER

The *line compare register* allows a split-screen display.

REGISTER FORMAT

Line Compare — Port 3D5/3B5, CRT Index 18

7	6	5	4	3	2	1	0
LC							

Bit No.	Meaning
7..0	Line Compare Target (LC)

Figure 253. Line compare register format

FIELD DEFINITION

The value loaded into the line compare field is the low-order 8 bits of the horizontal scan line counter at the time when the horizontal line counter is to be cleared.

REGISTER DESCRIPTION

A ninth line compare bit is located in the LC field of the overflow register (see figure 228), and a tenth line compare bit is located in the LC field of the maximum scan line register (see figure 232).

12.6.27. POWER 9100 INTERLACE REGISTER

The *Power 9100 interlace register* determines when the vertical counter is clocked for the second time during interlace modes of operation.

The Power 9100 interlace register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Interlace — Port 03B5 (monochrome) CRT Index 19, Port 03D5 (color)

7	6	5	4	3	2	1	0
Interlace Extra Clock Position							

Bit No.	Meaning
7..0	Interlace Extra Clock Position (IECP)

Figure 254. Power 9100 interlace register format

FIELD DEFINITION

Bits [7..0] define the *horizontal counter value* at which the vertical counter is clocked for the second time during interlaced modes of operation.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex in monochrome mode and location 03D5 hex in color mode when the index field of the CRT controller index register is 19 hex.

12.6. CRT Controller Registers, continued

12.6.28. POWER 9100 SERIAL START ADDRESS HIGH REGISTER

The *Power 9100 serial start address high register* specifies the high-order bits of the start address of the frame buffer.

The Power 9100 serial start address high register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Serial Start Adress High	Port 03B5 (mono) Port 03D5 (color)	CRT Index 1A

7	6	5	4	3	2	1	0
Serial Video Start Address Bits [15..8]							

Bit No.	Meaning
7..0	Serial Video Start Address bits [15..8] (SVSAH)

Figure 255. Power 9100 serial start address high register format

FIELD DEFINITION

Bits [15..8] are the high-order bits of the 17-bit serial video start address that consists of the serial video start address low register (bits [7..0]), the serial video start high register (bits [15..8]) and bit 2 of the Power 9100 control register 0 (bit 16).The Power 9100 has two sets of start address registers. When bit 7 of the Power 9100 control register 0 is reset to 0, only the set or start address registers that occupies the standard VGA addresses is enabled. When bit 7 of the Power 9100 control register 0 is set to 1, both sets of start address registers are enabled. When both sets of registers are enabled, the serial video start address specifies the location of the first pixel displayed for the frame buffer which is accessed through the VRAM serial port. The remaining set of registers, the standard VGA start address low and high registers, specify the start address of the frame buffer which is accessed through the VRAM parallel port.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex in monochrome mode or location 03D5 hex in color mode when the index field of the CRT controller index register is 1A hex.

12.6. CRT Controller Registers, continued

12.6.29. POWER 9100 SERIAL START ADDRESS LOW REGISTER

The *Power 9100 serial start address low register* specifies the high-order bits of the start address of the frame buffer.

The Power 9100 serial start address low register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Serial Start Address Low	Port 03B5 (mono) Port 03D5 (color)	CRT Index 1B

7	6	5	4	3	2	1	0
Serial Video Start Address Bits [7..0]							

Bit No.	Meaning
7..0	Serial Video Start Address bits [7..0] (SVSAL)

Figure 256. Power 9100 serial start address low register format

FIELD DEFINITION

Bits [7..0] are the low-order bits of the 17-bit serial video start address that consists of the serial video start address low register (bits [7..0]), the serial video start high register (bits [15..8]) and bit 2 of the Power 9100 control register 0 (bit 16). The Power 9100 has two sets of start address registers. When bit 7 of the Power 9100 control register 0 is reset to 0, only the set that occupies the standard VGA addresses is enabled. When bit 7 of the Power 9100 control register 0 is set to 1, both sets of start address registers are enabled. When both sets of registers are enabled, the serial video start address specifies the location of the first pixel displayed for the frame buffer which is accessed through the VRAM serial port. The remaining set of registers, the standard VGA start address low and high registers, specify the start address of the frame buffer which is accessed through the VRAM parallel port.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex in monochrome mode or location 03D5 hex in color mode when the index field of the CRT controller index register is 1B hex.

12.6. CRT Controller Registers, continued

12.6.30. POWER 9100 SERIAL OFFSET REGISTER

The *Power 9100 serial offset register* specifies the logical line width of the frame buffer that is accessed through the VRAM serial port.

The Power 9100 serial offset register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Serial Offset | Port 03B5 (mono) Port 03D5 (color) | CRT Index 1C

7	6	5	4	3	2	1	0
Serial Video Offset							

Bit No.	Meaning
7..0	Serial Video Offset (SVO)

Figure 257. Power 9100 serial offset register format

FIELD DEFINITION

Bits [7..0] are the *serial video offset.* This value specifies the logical line width of the frame buffer which is accessed through the VRAM serial port. The starting memory address for the next character row is two or four times this amount. The serial video offset register is programmed with a word address. Depending on the clocking mode of the CRT controller, this is either a word address or a double-word address. The Power 9100 has two offset registers. When bit 7 of the Power 9100 control register 0 is clear, only one is enabled and it occupies the standard VGA addresses. When bit 7 of the Power 9100 control register 0 is set, both offset registers are enabled. In this case, the standard VGA offset register defines the offset for the frame buffer which is accessed through the VRAM parallel port; the other, the serial video offset register, defines the offset for the frame buffer which is accessed through the VRAM serial port.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex in monochrome mode and location 03D5 hex in color mode when the index field of the CRT controller index register is 1C hex.

12.6. CRT Controller Registers, continued

12.6.31. POWER 9100 TOTAL CHARACTERS PER LINE REGISTER

The *Power 9100 total characters per line register* specifies the total number of characters per line when the frame buffer is accessed through the parallel VRAM port.

The Power 9100 total characters per line register must be unlocked before it can be accessed. See section 12.1.1 for more information.

REGISTER FORMAT

Power 9100 Total Characters/Line | Port 03B5 (monochrome) Port 03D5 (color) | CRT Index 1D

7	6	5	4	3	2	1	0
Total Characters Per Line							

Bit No.	Meaning
7..0	Total Characters Per Line (TCPL)

Figure 258. Power 9100 total characters/line register format

FIELD DEFINITION

Bits [7..0] specify the *total characters per line*. The Power 9100 has two registers which can specify the number of characters in a line. When bit 7 of the Power 9100 control register 0 is clear, only the horizontal display enable end register is enabled and it occupies the standard VGA addresses. When bit 7 of the Power 9100 control register 0 is set, both registers are enabled. In this case, the standard VGA horizontal display end register defines the number of graphics bytes per line for the frame buffer which is accessed through the VRAM serial port; the other, the total characters per line register, defines the total number of characters per line for the frame buffer which is accessed through the VRAM parallel port.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03B5 hex in monochrome mode or 03D5 hex in color mode when the index field of the CRT controller index register is 1D hex. When DRAM is used in the packed pixel modes, this register must be programmed with a value equal to the total number of bytes in a line divided by 8.

12.6. CRT Controller Registers, continued

12.6.32. POWER 9100 ATTRIBUTES STATE REGISTER

The *Power 9100 attributes state register* is a read only register which returns the state of the Attributes toggles flip-flop.

Power 9100 Attributes State — Port 03B5 (monochrome) CRT Index 24, Port 03D5 (color)

7	6	5	4	3	2	1	0
ATFF	Reserved						

Bit No.	Meaning
7	Returns state of Attributes toggle flip-flop 0 = index mode 1 = data mode
6..0	Reserved

Figure 259. Power 9100 attribute state register

FIELD DEFINITION

Bit 7 of this read–only register returns the state of the attributes toggle flip-flop. 0 indicates index mode and 1 indicates data mode.

Bits [6..0] are reserved.

REGISTER DESCRIPTION

This video control read register is accessed through location 03B5 hex in monochrome mode and location 03D5 hex in color mode when the index field of the CRT controller index register is 24 hex.

9663826 0003899 877

12.7. Graphics Controller Registers

The graphics controller register group provide hardware assistance to graphics drawing operations. These registers are accessed via the graphics controller index register port at hex address 3CE and the graphics controller data registers port at hex address 3CF.

12.7.1. GRAPHICS INDEX REGISTER

The *graphics index register* provides the address index for the graphics controller registers.

REGISTER FORMAT

Graphics Index Port 3CE

7	6	5	4	3	2	1	0
Res	Reserved			GRA			

Bit No.	Meaning
7	Reserved
6..4	Reserved
3..0	Graphics Controller Register Index (GRA)

Figure 260. Graphics index register format

FIELD DEFINITION

Bit 7 is reserved.

Bits [6..0] are defined as in the standard VGA.

Field	Register	Section
0	Set/reset register	12.7.2
1	Enable set/reset register	12.7.3
2	Color compare register	12.7.4
3	Data rotate register	12.7.5
4	Read map select register	12.7.6
5	Graphics mode register	12.7.7
6	Miscellaneous register	12.7.8
7	Color don't care register	12.7.9
8	Bit mask register	12.7.10
9	Reserved	
A	Reserved	
B	Reserved	
C	Reserved	
D	Reserved	
E	Reserved	
F	Reserved	

Figure 261. Bit 3..0 field

REGISTER DESCRIPTION

The *graphics index register* is a pointer register which is located at address 03CE hex. The value loaded in this register determines which sequencer register is accessed when I/O operations are performed to address 03CF hex. This value is referred to as the index.

12.7. Graphics Controller Registers, continued

12.7.2. SET/RESET REGISTER

The *set/reset register* provides color fill data to the display memory maps.

REGISTER FORMAT

Set/Reset Port 3CF, Gra Index 00

7	6	5	4	3	2	1	0
Reserved				S/R			

Bit No.	Meaning
7..4	Reserved
3..0	Set/Reset (S/R)

Figure 262. Set/reset register format

FIELD DEFINITION

Bit	Value	Meaning
3	*x*	Fill data for memory map 3
2	*x*	Fill data for memory map 2
1	*x*	Fill data for memory map 1
0	*x*	Fill data for memory map 0

Figure 263. S/R field

REGISTER DESCRIPTION

Bits 3..0 are enabled by the enable set/reset (ESR) field of the enable set/reset register (see figure 265). Memory maps that are disabled by the ESR field receive regular data from the CPU.

The bit mask (BM) field of the bit mask register (see figure 280) write-protects individual memory bits from set/reset fill operations.

12.7.3. ENABLE SET/RESET REGISTER

The *enable set/reset register* determines the memory maps that receive fill data from the set/reset register.

REGISTER FORMAT

Enable Set/Reset Port 3CF, Gra Index 01

7	6	5	4	3	2	1	0
Reserved				ESR			

Bit No.	Meaning
7..4	Reserved
3..0	Enable Set reset (ESR)

Figure 264. Enable set/reset register format

FIELD DEFINITION

Bit	Value	Meaning
3	0	Disable set/reset for Map 3
	1	Enable set/reset for Map 3
2	0	Disable set/reset for Map 2
	1	Enable set/reset for Map 2
1	0	Disable set/reset for Map 1
	1	Enable set/reset for Map 1
0	0	Disable set/reset for Map 0
	1	Enable set/reset for Map 0

Figure 265. ESR field

REGISTER DESCRIPTION

Bits 3..0 enable the memory maps that receive fill data from the set/reset (S/R) field of the set/reset register (see figure 263). Memory maps that are disabled by bits 3..0 receive regular data from the CPU.

The bit mask (BM) field of the bit mask register (see figure 280) write-protects individual memory bits from set/reset fill operations.

12.7. Graphics Controller Registers, continued

12.7.4. COLOR COMPARE REGISTER

The *color compare register* permits the comparison of data on all four color maps to a reference color and reports whether a match was found for each pixel position.

REGISTER FORMAT

Color Compare — Port 3CF, Gra Index 02

7	6	5	4	3	2	1	0
Reserved				CC			

Bit No.	Meaning
7..4	Reserved
3..0	Color Compare (CC)

Figure 266. Color compare register format

FIELD DEFINITION

Bit	Value	Meaning
3	*x*	Color compare value for Map 3
2	*x*	Color compare value for Map 2
1	*x*	Color compare value for Map 1
0	*x*	Color compare value for Map 0

Figure 267. Bits 3..0 field

12.7. Graphics Controller Registers, continued

12.7.5. DATA ROTATE REGISTER

The *data rotate register* modifies data as it is being transferred from the CPU to the display memory.

REGISTER FORMAT

Data Rotate — Port 3CF, Gra Index 03

7	6	5	4	3	2	1	0
Res	Res	Res	FS		DR		

Bit No.	Meaning
7	Reserved
6	Reserved
5	Reserved
4..34	Function Select (FS)
2..0	Data Rotate (DR)

Figure 268. Data rotate register format

REGISTER DESCRIPTION

This graphics read/write register is accessed through location 03CF hex when the index field of the graphics address register is 03.

The data rotate operation occurs before other operations such as:

set/reset controlled by the set/reset (S/R) field of the set/reset register (see figure 263) and the enable set/reset (ESR) field of the enable set/reset register see (figure 265)

write mode selected by the write mode (WM) field of the mode register (see figure 273)

logical read before write controlled by the FS field

bit mask selected by the bit mask (BM) field of the bit mask register (see figure 280)

FIELD DEFINITION

Bits [7..5] are reserved.

Bits [4..0] are defined as in the standard VGA.

Bits	Value	Meaning
7		Reserved
6		Reserved
5		Reserved
4..3	00	Data written unmodified
	01	Data ANDed with latched data
	10	Data ORed with latched data
	11	Data XORed with latched data
D2..0	000	Rotate right shift 0 bits
	001	Rotate right shift 1 bit
	010	Rotate right shift 2 bits
	011	Rotate right shift 3 bits
	100	Rotate right shift 4 bits
	101	Rotate right shift 5 bits
	110	Rotate right shift 6 bits
	111	Rotate right shift 7 bits

Figure 269. Data rotate register fields

12.7. Graphics Controller Registers, continued

12.7.6. READ MAP SELECT REGISTER

The *read map select register* enable the memory map to be read by the CPU.

REGISTER FORMAT

Read Map Select   Port 3CF, Gra Index 04

7	6	5	4	3	2	1	0
Reserved						RMS	

Bit No.	Meaning
7..2	Reserved
1..0	Read Map Select (RMS)

Figure 270. Read map select register format

FIELD DEFINITION

Bits	Value	Meaning
1..0	00	Select Map 0
	01	Select Map 1
	10	Select Map 2
	11	Select Map 3

Figure 271. Read map select register fields

REGISTER DESCRIPTION

Bits 1..0 are not active in color compare mode (see figures 267).

The chain four (C4) field of the memory mode register of the sequencer registers group (see figure 204) controls display memory bit plane access. In write modes, the display plane is selected normally by the EM3..EM0 fields of the map mask register of the sequencer registers group (see figure 198). In read modes, the active bit plane is selected normally by bits 1..0.

12.7. Graphics Controller Registers, continued

12.7.7. GRAPHICS MODE REGISTER

The *graphics mode register* controls read and write modes.

REGISTER FORMAT

Graphics Mode Port 3CF, Gra Index 05

7	6	5	4	3	2	1	0
0	256CM	SRM	O/E	RM	R	WM	

Bit No.	Meaning
7	Reserved (must be 0)
6	256 Color Mode (256CM)
5	Shift Register Mode (SRM)
4	Odd/Even (O/E)
3	Read Mode (RM)
2	Reserved
1..0	Write Mode (WM)

Figure 272. Graphics mode register format

REGISTER DESCRIPTION

This graphics read/write register is accessed through location 03CF hex when the index field of the graphics address register is 05.

Bit 4 of the graphics mode register must be the complement of the O/E field of the memory mode register in the sequencer registers group (see figure 204).

Operation of bit 3 depends on read map select (RMS) field of read map select register (see figure 271), chain four (C4) field of sequencer memory mode register (see figure 204), and color compare (CC) field of color compare register (see figure 267).

Operation of bits 1..0 depend on data rotate (DR) field of rotate register (see figure 269), set/reset register (see figure 263), enable set/reset register (see figure 265), and bit mask register (see figure 280).

FIELD DEFINITION

Bits [6..0] are defined as in the standard VGA.

Bits	Value	Meaning
6	0	SRM field controls shift register loading
	1	Shift register load supports 256-color mode
5	0	Normal serial data stream formatting
	1	Serial data stream formatted with even-numbered bits from both maps on even-numbered maps and odd-numbered bits from both maps on odd-numbered maps
4	0	Normal VGA operating mode
	1	Even host addresses even display planes 0 and 2, odd host address odd display planes 1 and 3
3	0	Read mode 0: CPU reads memory map selected by read map select register (see figure 271), but has no effect when C4 field of sequencer memory mode register (see figure 204) = 1
	1	Read mode 1: CPU reads results of comparison of the four memory maps with CC field of color compare register (see figure 267)
1..0	00	Direct CPU write (write mode 0) either rotated by DR field of rotate register (see figure 269), or by the contents of the set/reset register (see figure 263) when enabled (see figure 265)
	01	Use latch content as write data (write mode 1)
	10	Color plane *n* filled with bit *n* value in processor write data (write mode 2)
	11	Write each plane with 8 identical bits (write mode 4) from set/reset register (see figure 263), with rotated CPU data ANDed with bit mask register data (see figure 280)

Figure 273. Graphics mode register fields

12.7. Graphics Controller Registers, continued

12.7.8. MISCELLANEOUS REGISTER

The *miscellaneous register* controls the display mode, monochrome graphics emulation, and memory mapping.

REGISTER FORMAT

Miscellaneous — Port 3CF, Gra Index 06

7	6	5	4	3	2	1	0
Reserved				MM		COE	G/A

Bit No.	Meaning
7..4	Reserved
3..2	Memory Map (MM)
1	Chain Odd-Even (COE)
0	Graphics/Alphanumeric Mode (G/A)

Figure 274. Miscellaneous register format

FIELD DEFINITION

Bits	Value	Meaning
3..2	00	Memory location A0000–BFFFF hex, 128K memory length
	01	Memory location A0000–AFFFF hex, 64K memory length
	10	Memory location B0000–B7FFF hex, 32K memory length
	11	Memory location B8000–BFFFF hex, 32K memory length
1	0	Standard addressing
	1	Higher-order address bit replaces host address bit (even and odd addresses access even and odd planes, respectively)
0	0	Select alphanumeric mode
	1	Select graphics mode

Figure 275. Miscellaneous register fields

REGISTER DESCRIPTION

Bit 0 should have the same contents as the G/A field of the attribute mode control register in the attribute controller registers group (see figure 286).

MM bit 1	MM bit 0	Display Buffer Starts at	Length (bytes)
0	0	A0000H	128K
0	1	A0000H	64K
1	0	B0000H	32K
1	1	B8000H	32K

Figure 276. Location of frame buffer in memory page

12.7. Graphics Controller Registers, continued

12.7.9. COLOR DON'T CARE REGISTER

The *color don't care register* masks particular memory maps from being tested during color compares.

REGISTER FORMAT

Color Don't Care — Port 3CF, Gra Index 07

7	6	5	4	3	2	1	0
Reserved				CDC			

Bit No.	Meaning
7..4	Reserved
3..0	Color Don't Care (CDC)

Figure 277. Color don't care register format

FIELD DEFINITION

Bit	Value	Meaning
3	0	Participate in color compare cycle, Map 3
	1	Ignore color compare cycle, Map 3
2	0	Participate in color compare cycle, Map 2
	1	Ignore color compare cycle, Map 2
1	0	Participate in color compare cycle, Map 1
	1	Ignore color compare cycle, Map 1
0	0	Participate in color compare cycle, Map 0
	1	Ignore color compare cycle, Map 0

Figure 278. Color don't care field

12.7.10. BIT MASK REGISTER

The *bit mask register* prevents certain bit positions from being modified during memory read-modify-write cycles.

REGISTER FORMAT

Bit Mask — Port 3CF, Gra Index 08

7	6	5	4	3	2	1	0
BM							

Bit No.	Meaning
7..0	Bit Mask (BM)

Figure 279. Bit mask register format

FIELD DEFINITION

Bit	Value	Meaning
x	0	Bit *x* immune to change
	1	Bit *x* writes enabled

Figure 280. Bit mask field

REGISTER DESCRIPTION

Bit *x* is immune to change when the bit mask field is 0 only if the location being written is the last location read.

The bit mask write-protects individual memory bits from set/reset fill operations specified by the set/reset (S/R) field of the set/reset register (see figure 263) and enabled by the enable set/reset (ESR) field of the enable set/reset register (see figure 265).

12.8. Attribute Controller Registers

The attribute controller register group controls display attributes such as color, blinking, and underlining. These registers are written at hex address 3C0, with access alternating between the attribute controller index register and the selected attribute controller data register. These registers are read at hex address 3C1, with access alternating between the address and data registers.

12.8.1. ATTRIBUTE INDEX REGISTER

The *attribute index register* provides the address index for the attribute controller registers.

REGISTER FORMAT

Attribute Index — Port 3C1 (Read), 3C0 (Write)

7	6	5	4	3	2	1	0
Reserved		PAS	ACR				

Bit No.	Meaning
7..6	Reserved
5	Palette Address Source (PAS)
4..0	Attribute Address (ACR)

Figure 281. Attribute index register format

FIELD DEFINITION

Field	Registers	Section
00000	Palette register 00	12.8.2
00001	Palette register 01	12.8.2
00010	Palette register 02	12.8.2
00011	Palette register 03	12.8.2
00100	Palette register 04	12.8.2
00101	Palette register 05	12.8.2
00110	Palette register 06	12.8.2
00111	Palette register 07	12.8.2
01000	Palette register 08	12.8.2
01001	Palette register 09	12.8.2
01010	Palette register 0A	12.8.2
01011	Palette register 0B	12.8.2
01100	Palette register 0C	12.8.2
01101	Palette register 0D	12.8.2
01110	Palette register 0E	12.8.2
01111	Palette register 0F	12.8.2
10000	Attribute mode control register	12.8.3
10001	Overscan control register	12.8.4
10010	Color plane enable register	12.8.5
10011	Horizontal pixel panning register	12.8.6
10100	Color select register	12.8.7
10101	Power 9100 overscan color high register	12.8.8
10110..11111	Reserved	

Figure 282. Attribute index field definition

12.8. Attribute Controller Registers, continued

12.8.2. PALETTE REGISTERS

The 16 *palette registers* allow the CPU to determine which colors are displayed at any time.

REGISTER FORMAT

Palette

Port 3C1 (Read), 3C0 (Write)
ACR Indices 00–0F

7	6	5	4	3	2	1	0
Reserved		SR	SG	SB	R	G	B

Bit No.	**Meaning**
7..6	Reserved
5	Secondary Red (SR)
4	Secondary Green (SG)
3	Secondary Blue (SB)
2	Red (R)
1	Green (G)
0	Blue (B)

Figure 283. Palette registers format

FIELD DEFINITION

Bit	Value	Meaning
5	0	Secondary red not present
	1	Secondary red present
4	0	Secondary green not present
	1	Secondary green present
3	0	Secondary blue not present
	1	Secondary blue present
2	0	Red not present
	1	Red present
1	0	Green not present
	1	Green present
0	0	Blue not present
	1	Blue present

Figure 284. Palette register fields

REGISTER DESCRIPTION

The palette registers should be modified only during the vertical retrace interval.

12.8.3. ATTRIBUTE MODE CONTROL REGISTER

The *attribute mode control register* controls VGA attributes.

REGISTER FORMAT

Attribute Mode Control — Port 3C1 (Read), 3C0 (Write) ACR Index 10

7	6	5	4	3	2	1	0
P54S	PELW	PELPC	R	EB	ELG	MEBB	EGM

Bit No.	Meaning
7	P5–P4 Select (Pr4S)
6	Pixel Width (PELW)
5	Pixel Panning Compatibility (PELPC)
4	Reserved
3	Enable Blink or Intensity (EB)
2	Enable Line Graphics Char Codes/Clip Disable (ELG)
1	Mono Mode/Black Background (MEBB)
0	Enable Graphics Mode (EGM)

Figure 285. Attribute mode control register format

FIELD DEFINITION

Bits [7..5] are defined as in the standard VGA.

Bit 4 is reserved, as defined in the standard VGA.

Bit 3 is defined as in the standard VGA.

Bit 2 is the *enable line graphics character code/clip disable* bit. In normal VGA modes, when this bit is is reset to 0, the ninth dot of nine-dot-wide characters is the background color. When this bit is set to 1, the eighth bit of the font data is duplicated into the ninth bit for character codes C0 through DF. When the Power 9100 character attributes are used, this bit defines whether a character is allowed to stretch over the edges of a character cell, as can be invoked by the bold and half-bit-shift attributes. When this bit is clear, the character is clipped at the edges of the character cell; when this bit is set, the character is not clipped.

Bit 1 is the *mono mode/black background* bit. (Note that this is also a standard VGA bit.) In normal VGA modes, when this bit is set to 1, monochrome mode is enabled. When this bit is reset to 0, a color mode is enabled. When the Power 9100 overlapping text and graphics mode is invoked, if this bit is set to 1, the character background color is the first entry in the attribute color palette.

Bit 0 is defined as in the standard VGA.

Bit	Value	Meaning
7	0	Palette register bits 4 and 5 provide address bits 4 and 5 to color registers (see figure 284)
	1	Color select register C45 field provides address bits 4 and 5 to color registers (see figure 293).
6	0	Pixel data changed each dot clock cycle
	1	Pixel data changed every other dot clock
5	0	Prevent line compare from affecting pixel panning register output
	1	Allow line compare to affect horizontal pixel panning and preset row scan register outputs (see figures 290 and 230)
3	0	Character attribute code bit 7 selects background color. inhibit blinking
	1	Character attribute code bit 7 enables or disables blinking
2	0	Set ninth character dot to background color
	1	Set ninth character dot to eighth character dot for all graphics characters
1	0	Select color display attributes
	1	Select IBM Monochrome Display Adapter attributes
0	0	Select alphanumeric mode
	1	Select graphics mode

Figure 286. Attribute mode control register fields

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03C0 hex (write) or 03C1 hex (read) when the index field of the attribute address register is 10 hex.

Bit 5 allows line compare (see figure 253) to affect horizontal pixel panning and preset row scan register outputs (see figure 230 and 290).

Bit 0 should have the same contents as the G/A field of the miscellaneous register in the graphics controller registers group (see figure 275).

12.8. Attribute Controller Registers, continued

12.8.4. OVERSCAN COLOR REGISTER

The *overscan color register* determines the color of the border area displayed on the CRT.

REGISTER FORMAT

Overscan Color — Port 3C1 (Read), 3C0 (Write) ACR Index 11

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

Bit No.	**Meaning**
7	Pixel Address 7 (P7)
6	Pixel Address 6 (P6)
5	Pixel Address 5 (P5)
4	Pixel Address 4 (P4)
3	Pixel Address 3 (P3)
2	Pixel Address 2 (P2)
1	Pixel Address 1 (P1)
0	Pixel Address 0 (P0)

Figure 287. Overscan color register format

REGISTER DESCRIPTION

The border area includes the horizontal lines occurring between the value in the start vertical blanking register (see section 12.6.23) and the value in the vertical display enable register of the CRT controller registers group (see section 12.6.20).

The overscan color high (OCH) field of the Power 9100 overscan color high register of the attribute controller registers group (see figure 294) provide 8 additional high-order border-color bits for Power 9100 modes.

12.8. Attribute Controller Registers, continued

12.8.5. COLOR PLANE ENABLE REGISTER

The *color plane enable register* controls which color planes will be enabled during the display process.

REGISTER FORMAT

Color Plane Enable — Port 3C1 (Read), 3C0 (Write)
ACR Index 12

7	6	5 4	3 2 1 0
SBE	HBM	VSM	CPE

Bit No.	Meaning
7	Power 9100 Slow Blink Enable (SBE)
6	Power 9100 Half Bright Mode (HBM)
5..4	Video Status MUX (VSM)
3..0	Color Planes Enable (CPE)

Figure 288. Color plane enable register format

FIELD DEFINITION

Bit 7 is the *slow-blink enable* bit. The character/line blinking rate on the standard VGA is determined by a counter which counts a fixed number of vertical frame periods. On modes with higher frame rates, this results in faster blinking. When this bit is set to 1, the blinking rate is halved.

Bit 6 is the *half-bright mode* bit. This mode is intended for use with monitors which have only two levels of grey. When this bit is set to 1, and the half-bright attribute is used, alternate lines of the character are blanked, simulating half brightness.

Bits [5..0] are defined as in the standard VGA.

Bits	Value	Meaning
7	0	Normal blinking rate
	1	Cut blinking rate in half
6	0	Normal brightness control
	1	Blank alternate lines of the character when the half bright attribute is being used
5..4	00	DU field shows color outputs P2/P0
	01	DU field shows color outputs P5/P4
	10	DU field shows color outputs P3/P1
	11	DU field shows color outputs P7/P6
3..0	*xxx*0	Do not select display plane 0
	*xxx*1	Select display plane 0
	*xx*0*x*	Do not select display plane 1
	*xx*1*x*	Select display plane 1
	*x*0*xx*	Do not select display plane 2
	*x*1*xx*	Select display plane 2
	0*xxx*	Do not select display plane 3
	1*xxx*	Select display plane 3

Figure 289. Color plane enable register fields

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03C0 hex (write) or 03C1 hex (read) when the index field of the attribute address register is 12 hex.

Bits 5..4 specify the color output contents of the diagnostic use (DU) field of the input status 1 register in the general registers group (see figure 177).

12.8. Attribute Controller Registers, continued

12.8.6. HORIZONTAL PIXEL PANNING REGISTER

The *horizontal pixel panning register* selects the number of pixels by which to left-shift the video data horizontally.

REGISTER FORMAT

Horizontal Pixel Panning — Port 3C1 (Read), 3C0 (Write) ACR Index 13

7	6	5	4	3	2	1	0
Reserved				HPP			

Bit No.	Meaning
7..4	Reserved
3..0	Horizontal Pixel Pan (HPP)

Figure 290. Horizontal pixel panning register format

FIELD DEFINITION

Bits 3..0	Number of Pixels Shifted to Left		
	0+, 1+, 2+, 3+, 7, 7+	All Other Modes	Mode 13
0000	1	0	0
0001	2	1	—
0010	3	2	1
0011	4	3	—
0100	5	4	2
0101	6	5	—
0110	7	6	3
0111	8	7	—
1000	0	—	—
1001–1111	—	—	—

Figure 291. Bits 3..0 field

REGISTER DESCRIPTION

Bits 3..0 determine the number of pixels to pan, and the preset row scan register byte panning (BP) field in the CRT controller registers group controls the number of bytes to pan (see figure 230). The pixel panning compatibility (PPC) field of the attribute mode control register of the attribute controller registers group (see figure 286) allows line compare (see figure 253) to affect horizontal pixel panning and preset row scan register outputs (see figure 230).

12.8.7. COLOR SELECT REGISTER

The *color select register* combines with the palette registers to address the video DAC registers.

REGISTER FORMAT

Color Select — Port 3C1 (Read), 3C0 (Write) — ACR Index 14

7	6	5	4	3	2	1	0
Reserved				C67		C45	

Bit No.	Meaning
7..4	Reserved
3..2	Color Register Address Bits 6 and 7 (C67)
1..0	Color Register Address Bits 4 and 5 (C45)

Figure 292. Color select register format

FIELD DEFINITION

Bits	Meaning
3..2	Combined with 6 color palette output bits to form 8-bit address to color registers
1..0	Combined with C67 field and 4 low-order color palette output bits for form 8-bit color register address (see figure 284)

Figure 293. Color select register fields

REGISTER DESCRIPTION

Bits 1..0 are used only when the internal palette size (IPS) field of the attribute mode control register is set to 1 (see figure 286).

12.8.8. POWER 9100 OVERSCAN COLOR HIGH REGISTER

The *Power 9100 overscan color high register*, in conjunction with the standard VGA overscan color register, determines the border color.

REGISTER FORMAT

Power 9100 Overscan Color High — Port 03C0 (write), Port 03C1 (read) — Att Index 15

7	6	5	4	3	2	1	0
Overscan Color High							

Bit No.	Meaning
7..0	Overscan Color High (OCH)

Figure 294. Power 9100 overscan color high register format

FIELD DEFINITION

Bits [7..0] are the *overscan color high* bits. This 8-bit value determines the 8 high-order bits of the overscan or border color. The border is a band of color, one 80-column character wide, around the edges of the display area. Borders are not supported in 40-column alphanumeric modes or in 320 PEL graphics modes, except mode 13 (256 color). This register is usable in 16-bit direct color modes.

REGISTER DESCRIPTION

This video control read/write register is accessed through location 03C0 hex (write) or location 03C1 hex (read) when the index field of the attribute address register is 15 hex. This Power 9100 register is not a standard VGA register.

The 8 low-order bits specifying border colors for either standard VGA modes or Power 9100 modes is provided by the overscan color register of the attribute controller registers group (see figure 287).

Chapter 13. Specifications

13.1. DC Specifications

13.1.1. ABSOLUTE MAXIMUM RATINGS

Parameter	Value	Unit
Supply Voltage	–0.5 to +7.0	Volts
Input Voltage	–0.5 to Vcc + 0.5	Volts
Output Voltage	–0.5 to Vcc + 0.5	Volts
Storage Temperature Range	–40 to +125	°C
Lead Temperature (~10 seconds)	250	°C

Figure 295. Absolute maximum ratings

13.1.2. RECOMMENDED OPERATING CONDITIONS

Parameter	Minimum Value	Nominal Value	Maximum Value
Supply Voltage (V_{CC})	4.75 VDC	5.0 VDC	5.25 VDC
Operating Case Temperature (T_{CASE})	0° C		85° C
θ_{CA} @ 0 lfm		16° C/W	
θ_{CA} @ 250 lfm		7° C/W	

Figure 296. Recommended operating conditions

13.1.3. PIN CAPACITANCE

Parameter	Description	Value
C_{IN}	Input capacitance	10 pF
C_{OUT}	Output capacitance	10 pF
C_{IO}	Bidirectional capacitance	10 pF
C_{CLK}	Clock capacitance	12 pF
C_{IDSEL}	IDSEL capacitance	8 pF

Figure 297. Pin capacitance (Capacitance not tested, $T_A = 25^\circ$ C, f = 1 MHz, VCC = 5 VDC)

13.1.4. DC CHARACTERISTICS

Parameter	Description	Test Conditions	Minimum	Maximum
V_{IH}	High-level input voltage	V_{CC} = MAX	2.0 VDC	
V_{IHC}	High-level input voltage for clock signals	V_{CC} = MAX	2.4 VDC	
V_{IL}/V_{ILC}	Low-level input voltage	V_{CC} = MIN		0.8 VDC
V_{OH}	High-level output voltage	V_{CC} = MIN, I_{OH} = – 1.0 mA	2.8 VDC	
V_{OL}	Low-level output voltage	V_{CC} = MIN, I_{OL} = 4.0 mA		0.4 VDC
I_{LO}	Output leakage current	V_{CC} = MAX, V_{OUT}= 0 to V_{CC}	–10 uA	+10 uA
I_{LI}	Input leakage current	V_{CC} = MAX, V_{IN} = 0 to V_{CC}	–10 uA	+10 uA
I_{CC}	Standby current	V_{CC} = MAX, f = 1 MHz		150 mA
I_{DD}	Switching current	V_{CC} = MAX, f = 50 Mhz		400 mA
I_{OHD}	HD[31..0] drive current		–8 mA	+8 mA
I_{OLD}	HD[31..0] drive current		–8 mA	+8 mA
I_{PLL}	Phase lock loop current			5 mA

Figure 298. DC characteristics over the operating range

13.2. Supported Components

WEITEK has analyzed the AC specifications of a number of VRAM, RAMDAC, and clock generator parts. The products listed below have published AC specifications that are compatible with the Power 9100.

13.2.1. COMPATIBLE VRAMS

Instead of separate CASs, the Power 9100 uses separate WEs for byte control; 256 KB x16 VRAMs need separate WEs. The Power 9100 is compatible with Extended Data Out VRAMs, but does not require it and using it no effect on performance. The Power 9100 does use the write-per-pixel feature found in most VRAMs. Half and full size shift registers are supported.

The following VRAMs work with the Power 9100. Some are not fast enough for 50 MHz operation, but will run at 40 or 45 MHz. Contact WEITEK for the latest information .

VRAM Type	40 MHz	45 MHz	50 MHz
Micron MT42C8256–6	✓	✓	✓
Micron MT42C8256–7	✓	✓	
Micron MT42C8255–7	✓		
Samsung KM428C256–6	✓	✓	✓
IBM IBM025170–60	✓	✓	✓
IBM IBM025170–70	✓	✓	
IBM IBM025171–60	✓	✓	✓
IBM IBM025171–70	✓	✓	
NEC µPD482234–60	✓	✓	✓
NEC µPD482234–70	✓		
NEC µPD482235–60	✓	✓	✓
NEC µPD482235–70	✓		
Mitsubishi M5M482256–70	✓		
Mitsubishi M5M482257–70	✓		
Hitachi HM538253–70	✓		
Hitachi HM538254–70	✓		
Toshiba TC528257–70	✓		
OKI M5M548263–70	✓		

Figure 299. VRAMs known to work with the Power 9100

13.2.2. COMPATIBLE RAMDACS

The following RAMDACs are electrically compatible with the Power 9100. See the "Comments" field for information about software driver support for individual RAMDACs.

Type	Comments
Brooktree Bt485	32-bit RAMDAC
AT&T 20C505	Bt485 compatible, 32-bit
Brooktree Bt885	32-bit RAMDAC
IBM RGB525 family	64-bit RAMDAC
Brooktree Bt489	64-bit RAMDAC
AT&T 511	64-bit RAMDAC

Figure 300. Power 9100-compatible RAMDACs

13.2.3. COMPATIBLE CLOCK GENERATORS

The following clock generators are compatible with the Power 9100.

Type	Comments
IC Designs ICD2061A	
IC Designs ICD2062	
ICS9161	ICD2061A compatible
IBM525	(internal to DAC)

Figure 301. Power 9100-compatible clock generators

13.3. AC Specifications

M = one memclk period (ns).

Timings for 50MHz mode depend on the following settings:

mem_config.vram_miss_adj=1

mem_config.vram_read_adj=1

mem_config.vram_write_adj=1

mem_config.vram_sample_adj=1

Param.	Description	Reference Signal	50 MHz		Unit	Loading
			MIN	MAX		
VESA Local Bus Interface						
T_{LCT}	LCLK cycle time		20		ns	
T_{LCHL}	LCLK high/low time		8		ns	
T_{G1S}	ADR[31..2], BE[3..0]–, M/IO–, RESET–, RDYRNT– input setup time	LCLK		7	ns	
T_{G1H}	BE[3..0]–, M/IO–, RESET–, RDYRNT– input hold time		0		ns	
T_{LRDD}	LRDY– output delay time	LCLK		10	ns	100 pF
T_{LRDV}	LRDY– output valid time	LCLK	3		ns	100 pF
T_{DS}	DATA[31..0] input setup time	LCLK		7	ns	
T_{DH}	DATA[31..0] input hold time		0		ns	
T_{DD}	DATA[31..0] output delay time	LCLK		15+1 LCLK	ns	100 pF
T_{DTO}	DATA[31..0] output turn-off time	LCLK		3	ns	100 pF
T_{LDD}	LDEV– output delay time	ADR[31..2], BE–		20	ns	33 pF
T_{LRDV}	LDEV– output valid time		0		ns	33 pF
PCI Bus Interface						
T_{BCT}	CLK cycle time		20		ns	
T_{BCHL}	CLK high/low time		8		ns	
T_{INS}	C/BE[3..0]–, FRAME–, IRDY–, IDSL input setup times	CLK		7	ns	
T_{INH}	C/BE[3..0]–, FRAME–, IRDY–, IDSL input hold times		0		ns	
T_{IOS}	AD[31..0], PAR input setup times	CLK		7	ns	
T_{IOS}	AD[31..0], PAR input hold times		0		ns	
T_{IOD}	AD[31..0], PAR output delay times	CLK		11+1 CLK	ns	50 pF
T_{IOV}	AD[31..0], PAR output valid times		2		ns	50 pF
T_{IOTO}	AD[31..0], PAR turn-off times		2	15	ns	50 pF

Figure 302. AC Specifications for VL and PCI bus interfaces

13.3. AC Specifications, continued

Param.	Description	Reference Signal	MIN	MAX	Unit	Loading
Video Interface						
T_{DCPY}	DIVPIXCLK period		11.76		ns	
T_{DPCH}	DIVPIXCLK high		6.12		ns	
T_{DPCL}	DIVPIXCLK low		4.23		ns	
T_{PCY}	PIXCLK period		12.5		ns	
T_{PCH}	PIXCLK high		6.5		ns	
T_{PCL}	PIXCLK low		4.5		ns	
T_{HSD}	HSYNC output delay	VIDOUTCLK+		5	ns	
T_{VSD}	VSYNC output delay	VIDOUTCLK+		5	ns	
T_{BD}	BLANK– output delay	VIDOUTCLK		5	ns	
T_{HSS}	HSYNC– setup time		5		ns	
T_{HSH}	HSYNC hold time		5		ns	
T_{VSS}	VSYNC setup time		5		ns	
T_{VSH}	VSYNC– hold time		3		ns	
T_{SCD}	SC output delay	VIDOUTCLK	1.5	5	ns	
T_{SED}	SE output delay	VIDOUTCLK	1.5	5		
T_{OH}			0		ns	
T_{OE}			6M			
T_{ACC}			6M			
RAMDAC Interface						
T_{DWS}	RS[3..0] to DACWR– setup time		14		ns	
TDWH	DACWR– deasserted to MD[23..16] hold time		17		ns	

Figure 303. Video and RAMDAC interface specifications

13.3. AC Specifications, continued

Parameter	Description	Reference Signal	Value at 50 MHz, ns		Formula (M = MEMCLK period in ns)
			MIN	MAX	
Cycle Time Parameters					
T_{RC}	random cycle time		140		7M
T_{PC}	page-mode cycle time		40		2M
Read Access Parameters					
T_{RAC}	access time from RAS–			72	4.25M–13
T_{CAC}	access time from CAS–			18	1.5M–12
T_{AA}	access time from column address			35	2.25M–10
T_{OEA}	access time from OE–			27	1.75M–8
T_{CPA}	access time from CAS– precharge			35	2.25M–10
T_{RCS}	read command setup	CAS–	9		0.75M–6
T_{RCH}	read command hold to CAS–	CAS–	11		1M–9
T_{RRH}	read command hold to RAS–	RAS–	11		1M–9
T_{OFF}	output turn-off delay from CAS–			15	0.75M
RAS– Parameters					
T_{RAS}	RAS– pulse width		75		4M–5
T_{RASP}	RAS– pulse width (fast page mode)		75		4M–5
T_{RSH}	RAS– hold time		20		1.25M–5
T_{RP}	RAS– precharge		54		3M–6
T_{ASR}	row address setup to RAS–	RAS–	31		2M–9
T_{RAH}	row address hold to RAS–		10		1M–10
T_{AR}	column address hold to RAS–	RAS–	75		4M–5
T_{RCD}	RAS– to CAS– delay		31		2M–9
T_{ROH}	OE– to RAS– delay	OE–	21		1.5M–9
T_{RAL}	column address to RAS– lead time		30		1.75M–5
CAS– Parameters					
T_{CAS}	CAS– pulse width		20		1.25M–5
T_{CSH}	CAS– hold time	RAS–	70		3.75M–5
T_{CP}, T_{CPN}	CAS– precharge		12		0.75M–3
T_{ASC}	column address setup to CAS–	CAS–	0		0.25M–5
T_{CAH}	column address hold to CAS–	CAS–	10		0.75M–5
T_{CRP}	CAS– to RAS– precharge time		51		3M–9

Figure 304. VRAM interface parameters (1 of 2)

13.3. AC Specifications, continued

Parameter	Description	Reference Signal	Value at 50 MHz, ns		Formula (M = MEMCLK period in ns)
			MIN	MAX	
Write Cycle Parameters					
T_{CWL}	write command to CAS– lead		31		2M–9
T_{DS}	data setup	CAS–	0		0.25M–5
T_{DH}	data hold	CAS–	10		0.75M–5
T_{DHR}	data hold to RAS–	RAS–	56		3.25M–9
T_{MS}	mask data to RAS– setup		26		1.75M–9
T_{MH}	mask data to RAS– hold		31		2M–9
T_{WCS}	write command setup	CAS–	1		0.5M–9
T_{WCH}	write command hold	CAS–	21		1.5M–9
T_{WCR}	write command hold to RAS–	RAS–	71		4M–9
T_{WP}	write command pulse width		31		2M–9
T_{WSR}	WE– to RAS– setup		31		2M–9
T_{RWH}	WE– to RAS– hold		31		2M–9
T_{RWL}	write command to RAS– lead		31		2M–9
OE– Parameters					
T_{OEZ}	output disable time	OE–		15	0.75M
T_{THS},T_{YS}	OE– high to RAS– setup		51		3M–9
T_{THH},T_{YH}	OE– high to RAS– hold		41		2.5M–9
T_{TLS}	OE– low to RAS– setup		31		2M–9
T_{TLH}	OE– low to RAS– hold		10		1M–10
T_{TRW},T_{TP}	OE– precharge		161		8.5M–9
T_{TRP}	OE– to RAS– precharge time		111		6M–9
Refresh Parameters (CAS– before RAS–)					
T_{CSR}	CAS– setup during refresh		31		2M–9
T_{CHR}	CAS– hold during refresh		51		3M–9
T_{RPC}	RAS– to CAS– precharge		11		1M–9
DSF Parameters					
T_{FSR}	DSF to RAS– setup	RAS–	51		3M–9
T_{RFH}	DSF to RAS– hold	RAS–	15		1M–5

Figure 304, continued. VRAM interface parameters (2 of 2)

13.4. VL Bus Pin Configuration

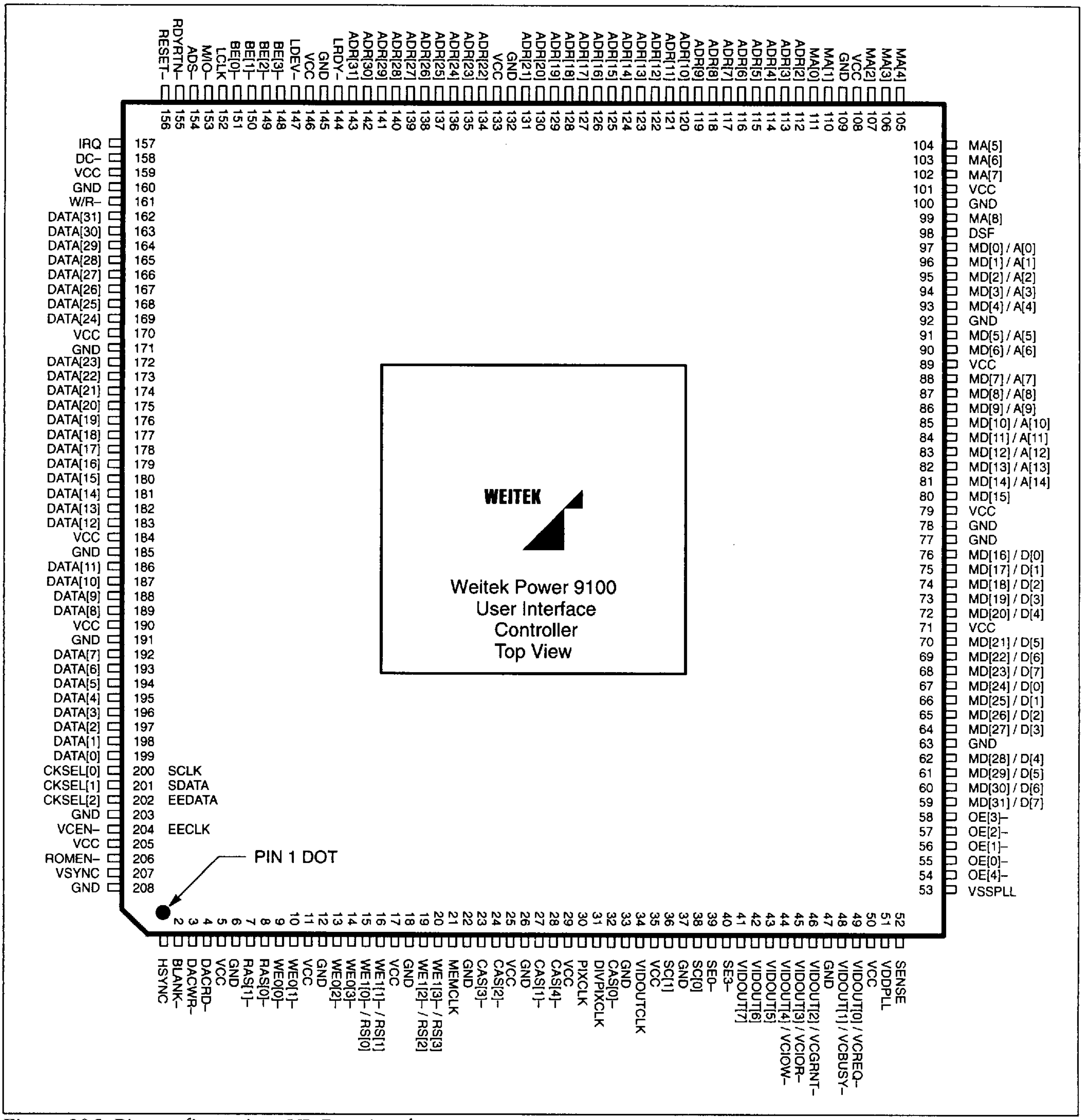


Figure 305. Pin configuration: VL Bus signals

13.5. PCI Bus Pin Configuration

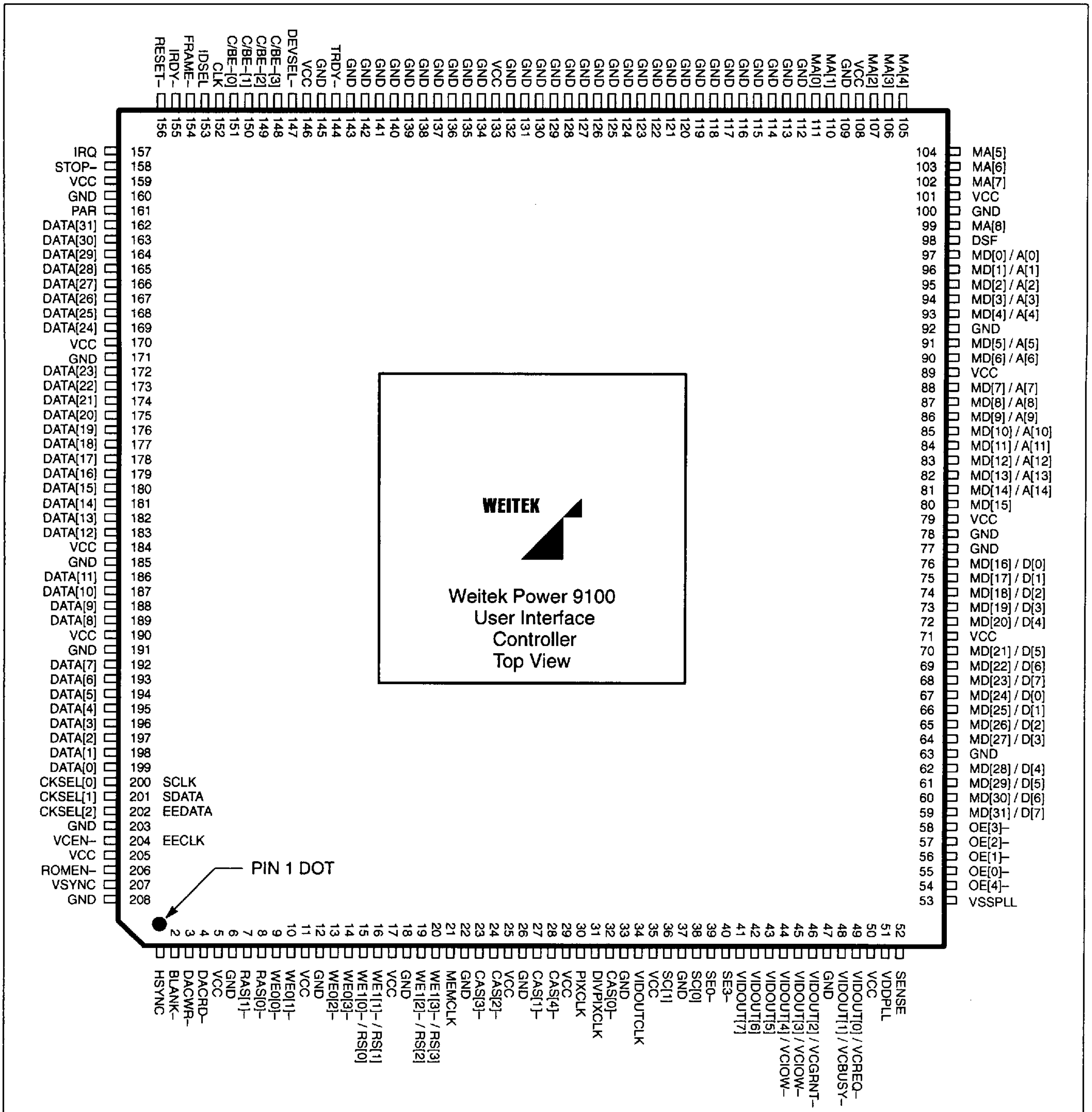


Figure 306. Pin configuration: PCI Bus signals

13.6. Pin Assignments

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
1	HSYNC	Input/ Output							HSYNC	Input/ Output
2	BLANK–	Output								
3					DAC-WR–	Output				
4					DAC-RD–	Output				
5	VCC									
6	GND									
7	RAS[1]–	Tri-stated/ Output							RAS[1]–	Tri-stated/ Output
8	RAS[0]–	Tri-stated/ Output							RAS[0]–	Tri-stated/ Output
9	WE0[0]–	Tri-stated/ Output							WE0[0]–	Tri-stated/ Output
10	WE0[1]–	Tri-stated/ Output							WE0[1]–	Tri-stated/ Output
11	VCC									
12	GND									
13	WE0[2]–	Tri-stated/ Output							WE0[2]–	Tri-stated/ Output
14	WE0[3]–	Tri-stated/ Output							WE0[3]–	Tri-stated/ Output
15	WE1[0]–	Tri-stated/ Output			RS[0]	Output			WE1[0]–	Tri-stated/ Output
16	WE1[1]–	Tri-stated/ Output			RS[1]	Output			WE1[1]–	Tri-stated/ Output
17	VCC									
18	GND									
19	WE1[2]–	Tri-stated/ Output			RS[2]	Output			WE1[2]–	Tri-stated/ Output
20	WE1[3]–	Tri-stated/ Output			RS[3]	Output			WE1[3]–	Tri-stated/ Output
21	MEMCLK	Input								
22	GND									
23	CAS[3]–	Tri-stated/ Output							CAS[3]–	Tri-stated/ Output
24	CAS[2]–	Tri-stated/ Output							CAS[2]–	Tri-stated/ Output
25	VCC									

Figure 307. Pin assignments (1 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
26	GND									
27	CAS[1]–	Tri-stated/ Output							CAS[1]–	Tri-stated/ Output
28	CAS[4]–	Tri-stated/ Output							CAS[4]–	Tri-stated/ Output
29	VCC									
30	PIXCLK	Input								
31	DIV-PIXCLK	Input								
32	CAS[0]–	Tri-stated/ Output							CAS[0]–	Tri-stated/ Output
33	GND									
34	VID-OUTCLK	Output								
35	VCC									
36	SC[1]	Output								
37	GND									
38	SC[0]	Output								
39	SE[0]–	Output								
40	SE[3]–	Output								
41	VIDOUT[7]	Tri-stated/ Output								
42	VIDOUT[6]	Tri-stated/ Output								
43	VIDOUT[5]	Tri-stated/ Output								
44	VIDOUT[4]	Tri-stated/ Output							VCIOW–	Output
45	VIDOUT[3]	Tri-stated/ Output							VCIOR–	Output
46	VIDOUT[2]	Tri-stated/ Output							VCGRNT–	Output
47	GND									
48	VIDOUT[1]	Tri-stated/ Output							VCBUSY–	Input
49	VIDOUT[0]	Input/ Output							VCREQ–	Input

Figure 307, continued. Pin assignments (2 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
50	VCC									
51	PLLVDD									
52	SENSE	Input								
53	PLLGND									
54	OE[4]–	Tri-stated/ Output							OE[4]–	Tri-stated/ Output
55	OE[0]–	Tri-stated/ Output							OE[0]–	Tri-stated/ Output
56	OE[1]–	Tri-stated/ Output							OE[1]–	Tri-stated/ Output
57	OE[2]–	Tri-stated/ Output							OE[2]–	Tri-stated/ Output
58	OE[3]–	Tri-stated/ Output							OE[3]–	Tri-stated/ Output
59	MD[31]	Input/ Output					D[7]	Input	VDDAT[31]	Input/ Output
60	MD[30]	Input/ Output					D[6]	Input	VDDAT[30]	Input/ Output
61	MD[29]	Input/ Output					D[5]	Input	VDDAT[29]	Input/ Output
62	MD[28]	Input/ Output					D[4]	Input	VDDAT[28]	Input/ Output
63	GND									
64	MD[27]	Input/ Output					D[3]	Input	VDDAT[27]	Input/ Output
65	MD[26]	Input/ Output					D[2]	Input	VDDAT[26]	Input/ Output
66	MD[25]	Input/ Output					D[1]	Input	VDDAT[25]	Input/ Output
67	MD[24]	Input/ Output					D[0]	Input	VDDAT[24]	Input/ Output
68	MD[23]	Input/ Output			D[7]	Input/ Output			VDDAT[23]	Input/ Output
69	MD[22]	Input/ Output			D[6]	Input/ Output			VDDAT[22]	Input/ Output
70	MD[21]	Input/ Output			D[5]	Input/ Output			VDDAT[21]	Input/ Output
71	VCC									
72	MD[20]	Input/ Output			D[4]	Input/ Output			VDDAT[20]	Input/ Output

Figure 307, continued. Pin assignments (3 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
73	MD[19]	Input/ Output			D[3]	Input/ Output			VDDAT[19]	Input/ Output
74	MD[18]	Input/ Output			D[2]	Input/ Output			VDDAT[18]	Input/ Output
75	MD[17]	Input/ Output			D[1]	Input/ Output			VDDAT[17]	Input/ Output
76	MD[16]	Input/ Output			D[0]	Input/ Output			VDDAT[16]	Input/ Output
77	GND									
78	GND									
79	VCC									
80	MD[15]	Input/ Output							VDDAT[15]	Input/ Output
81	MD[14]	Input/ Output					A[14]	Output	VDDAT[14]	Input/ Output
82	MD[13]	Input/ Output					A[13]	Output	VDDAT[13]	Input/ Output
83	MD[12]	Input/ Output					A[12]	Output	VDDAT[12]	Input/ Output
84	MD[11]	Input/ Output					A[11]	Output	VDDAT[11]	Input/ Output
85	MD[10]	Input/ Output					A[10]	Output	VDDAT[10]	Input/ Output
86	MD[9]	Input/ Output					A[9]	Output	VDDAT[9]	Input/ Output
87	MD[8]	Input/ Output					A[8]	Output	VDDAT[8]	Input/ Output
88	MD[7]	Input/ Output					A[7]	Output	VDDAT[7]	Input/ Output
89	VCC									
90	MD[6]	Input/ Output					A[6]	Output	VDDAT[6]	Input/ Output
91	MD[5]	Input/ Output					A[5]	Output	VDDAT[5]	Input/ Output
92	GND									
93	MD[4]	Input/ Output					A[4]	Output	VDDAT[4]	Input/ Output
94	MD[3]	Input/ Output					A[3]	Output	VDDAT[3]	Input/ Output
95	MD[2]	Input/ Output					A[2]	Output	VDDAT[2]	Input/ Output

Figure 307, continued. Pin assignments (4 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
96	MD[1]	Input/ Output					A[1]	Output	VDDAT[1]	Input/ Output
97	MD[0]	Input/ Output					A[0]	Output	VDDAT[0]	Input/ Output
98	DSF	Output							DSF	Tri-stated
99	MA[8]	Tri-stated/ Output							VDADR[8]	Tri-stated/ Output
100	GND									
101	VCC									
102	MA[7]	Tri-stated/ Output							VDADR[7]	Tri-stated/ Output
103	MA[6]	Tri-stated/ Output							VDADR[6]	Tri-stated/ Output
104	MA[5]	Tri-stated/ Output							VDADR[5]	Tri-stated/ Output
105	MA[4]	Tri-stated/ Output							VDADR[4]	Tri-stated/ Output
106	MA[3]	Tri-stated/ Output							VDADR[3]	Tri-stated/ Output
107	MA[2]	Tri-stated/ Output							VDADR[2]	Tri-stated/ Output
108	VCC									
109	GND									
110	MA[1]	Tri-stated/ Output							VDADR[1]	Tri-stated/ Output
111	MA[0]	Tri-stated/ Output							VDADR[0]	Tri-stated/ Output
112	ADR[2]	Input	GND							
113	ADR[3]	Input	GND							
114	ADR[4]	Input	GND							
115	ADR[5]	Input	GND							
116	ADR[6]	Input	GND							
117	ADR[7]	Input	GND							
118	ADR[8]	Input	GND							
119	ADR[9]	Input	GND							

Figure 307, continued. Pin assignments (5 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
120	ADR[10]	Input	GND							
121	ADR[11]	Input	GND							
122	ADR[12]	Input	GND							
123	ADR[13]	Input	GND							
124	ADR[14]	Input	GND							
125	ADR[15]	Input	GND							
126	ADR[16]	Input	GND							
127	ADR[17]	Input	GND							
128	ADR[18]	Input	GND							
129	ADR[19]	Input	GND							
130	ADR[20]	Input	GND							
131	ADR[21]	Input	GND							
132	GND									
133	VCC									
134	ADR[22]	Input	GND							
135	ADR[23]	Input	GND							
136	ADR[24]	Input	GND							
137	ADR[25]	Input	GND							
138	ADR[26]	Input	GND							
139	ADR[27]	Input	GND							
140	ADR[28]	Input	GND							
141	ADR[29]	Input	GND							
142	ADR[30]	Input	GND							
143	ADR[31]	Input	GND							
144	LRDY–	Tri-stated	TRDY–	Tri-stated						
145	GND									
146	VCC									
147	LDEV–	Output	DEVSEL–	Tri-stated						
148	BE[3]–	Input	C/BE[3]–	Input						
149	BE[2]–	Input	C/BE[2]–	Input						

Figure 307, continued. Pin assignments (6 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
150	BE[1]–	Input	C/BE[1]–	Input						
151	BE[0]–	Input	C/BE[0]–	Input						
152	LCLK	Input	CLK	Input						
153	M/IO–	Input	IDSEL	Input						
154	ADS–	Input	FRAME–	Input						
155	RDYRTN–	Input	IRDY–	Input						
156	RESET–	Input								
157	IRQ	Output	IRQ–	Output						
158	D/C–	Input	STOP–	Input/ Output						
159	VCC									
160	GND									
161	W/R–	Input	PAR	Input/ Output						
162	DATA[31]	Input/ Output								
163	DATA[30]	Input/ Output								
164	DATA[29]	Input/ Output								
165	DATA[28]	Input/ Output								
166	DATA[27]	Input/ Output								
167	DATA[26]	Input/ Output								
168	DATA[25]	Input/ Output								
169	DATA[24]	Input/ Output								

Figure 307, continued. Pin assignments (7 of 9)

13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
170	VCC									
171	GND									
172	DATA[23]	Input/ Output								
173	DATA[22]	Input/ Output								
174	DATA[21]	Input/ Output								
175	DATA[20]	Input/ Output								
176	DATA[19]	Input/ Output								
177	DATA[18]	Input/ Output								
178	DATA[17]	Input/ Output								
179	DATA[16]	Input/ Output								
180	DATA[15]	Input/ Output								
181	DATA[14]	Input/ Output								
182	DATA[13]	Input/ Output								
183	DATA[12]	Input/ Output								
184	VCC									
185	GND									
186	DATA[11]	Input/ Output								
187	DATA[10]	Input/ Output								
188	DATA[9]	Input/ Output								
189	DATA[8]	Input/ Output								

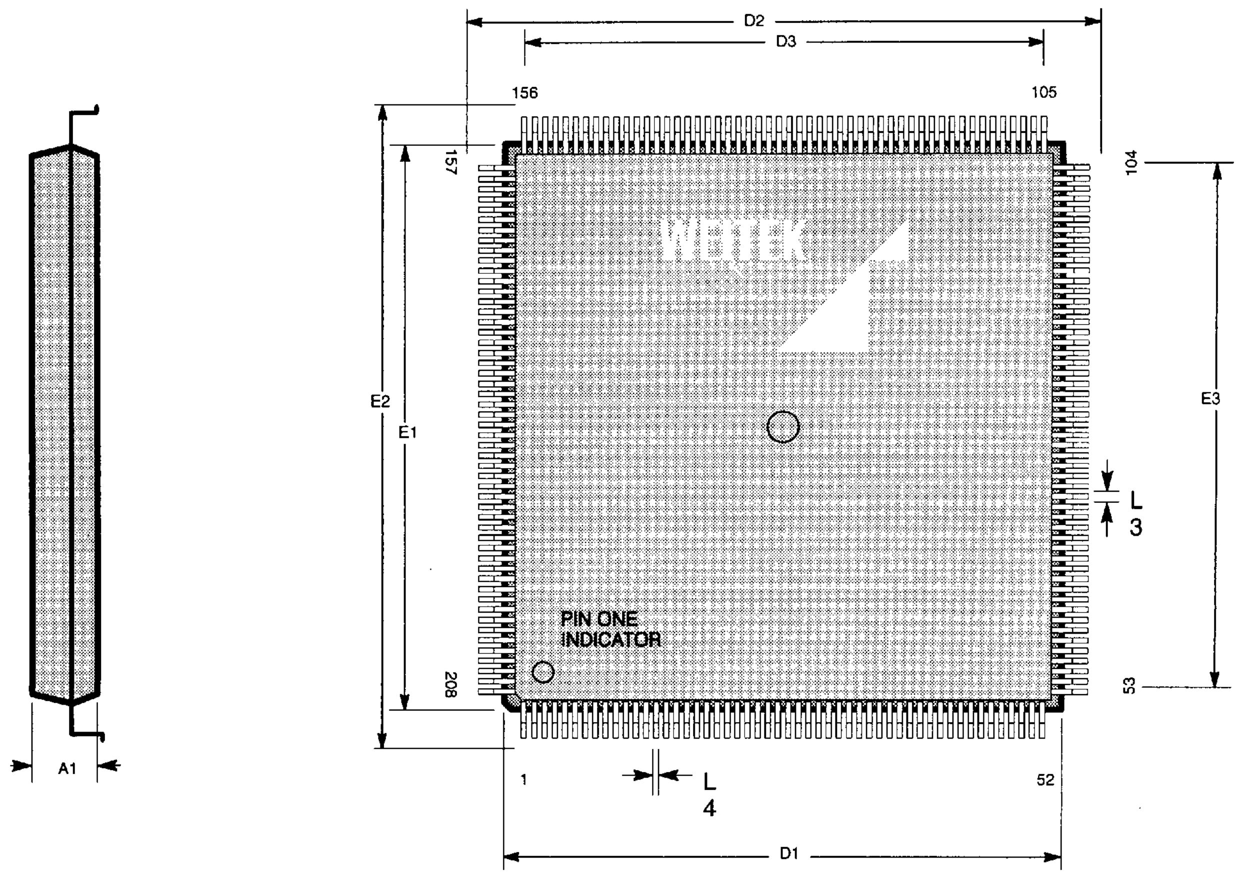
Figure 307, continued. Pin assignments (8 of 9)

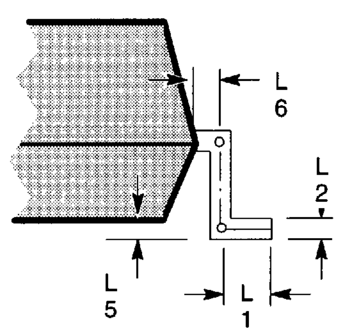
13.6. Pin Assignments, continued

Pin	Signal / VL Bus		PCI Bus (blank same as VL)		RAMDAC		BIOS / Clock / EEPROM		Video Coprocessor	
	Signal	Type	Signal	Type	Signal	Type	Signal	Type	Signal	Type
190	VCC									
191	GND									
192	DATA[7]	Input/ Output								
193	DATA[6]	Input/ Output								
194	DATA[5]	Input/ Output								
195	DATA[4]	Input/ Output								
196	DATA[3]	Input/ Output								
197	DATA[2]	Input/ Output								
198	DATA[1]	Input/ Output								
199	DATA[0]	Input/ Output								
200					CK-SEL[0]	Output	EEDATA	Input Output		
201					CK-SEL[1]	Output				
202					CK-SEL[2]	Input/ Output	EECLK	Output		
203	GND									
204									VCEN–	Output
205	VCC									
206							ROMEN–	Output		
207	VSYNC–	Input/ Output							VSYNC	Input/ Output
208	GND									

Figure 307, continued. Pin assignments (9 of 9)

13.7. Mechanical Specifications





SYMBOL	INCHES		MM	
	MAX	MIN	MAX	MIN
A1	0.130 TYP		3.49 TYP	
D1	1.106	1.098	28.10	27.90
D2	1.215	1.195	30.85	30.35
D 3	1.004 REF		25.5 REF	
E1	1.106	1.098	28.10	27.90
E2	1.215	1.195	30.85	30.35
E3	1.004 REF		25.5 REF	
L1	0.026	0.016	0.650	0.400
L2	0.008	0.004	0.2	0.1
L3	0.0197 TYP		0.50 TYP	
L4	0.009 TYP		0.22 TYP	
L5	0.020	0.010	0.5	0.25
L6	—	0.016	—	0.40

Figure 308. Physical dimensions (208-pin QFP)

13.8. I/O Characteristics

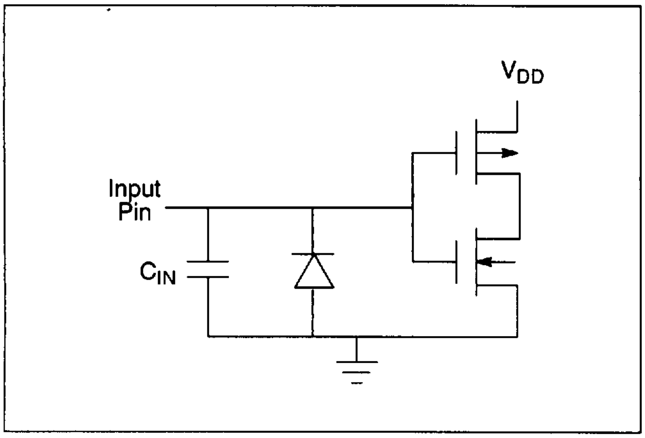


Figure 309. Input equivalent circuits

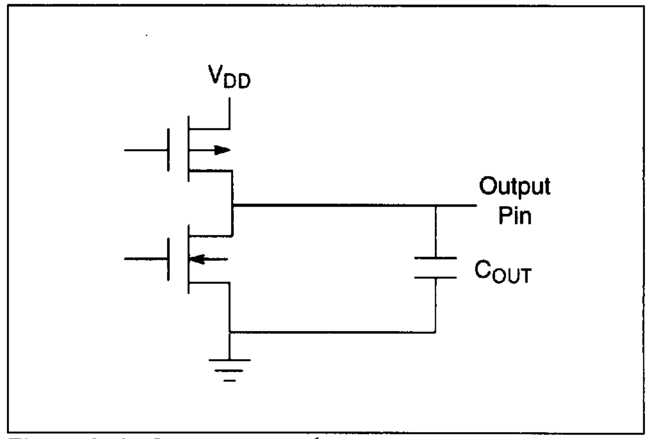


Figure 312. Output equivalent circuits

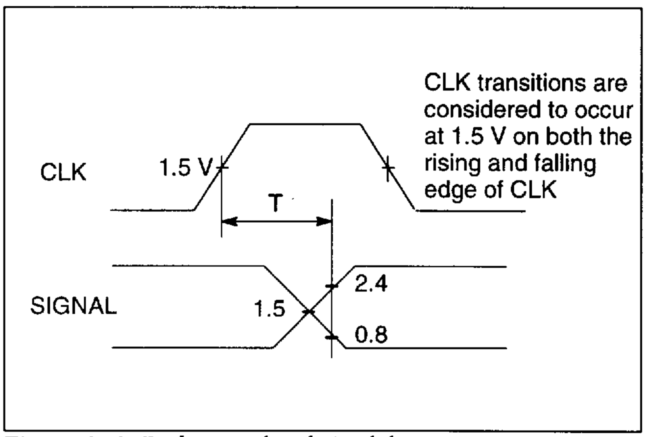


Figure 310. Reference levels in delay measurements on frame buffer controller pins

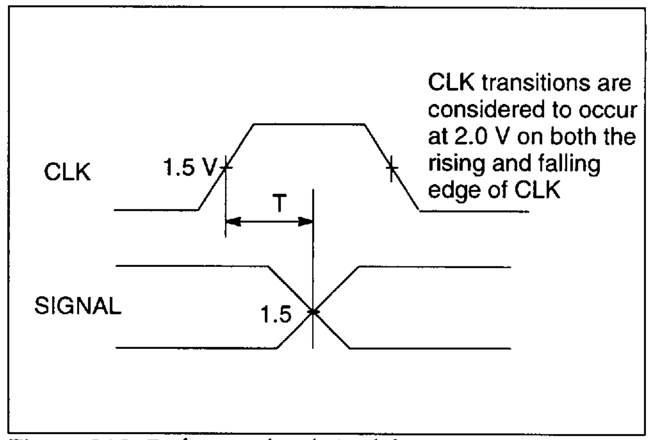


Figure 313. Reference levels in delay measurements on all pins other than frame buffer controller pins

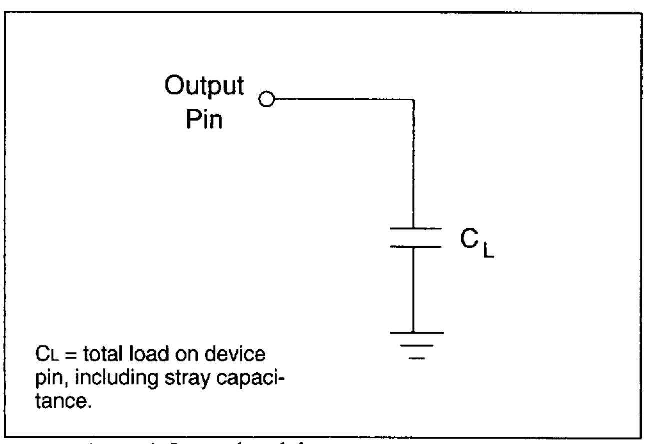


Figure 311. AC test load for test measurement

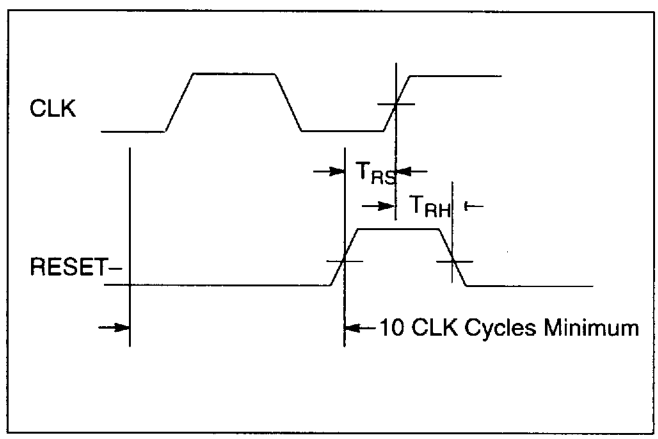


Figure 314. Reset timing

13.8. I/O Characteristics , continued

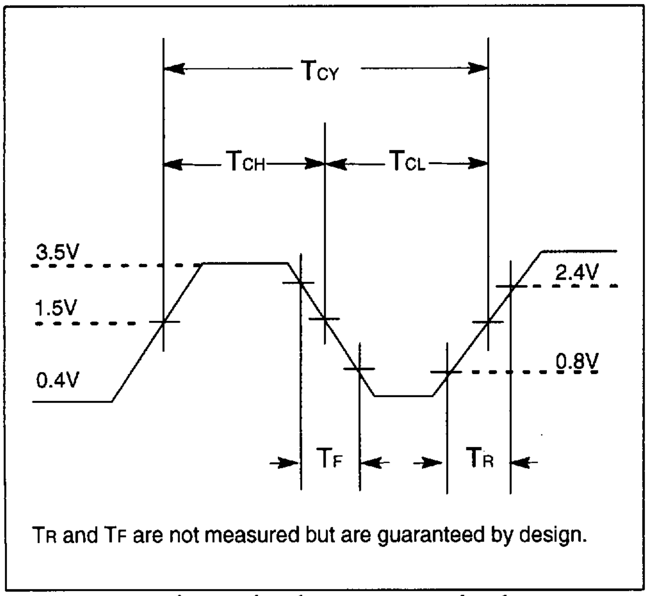


Figure 315. Relationship between signal voltages, rise and fall times, and cycle times of the various clocks

13.9. Ordering Information

Package Type	Speed Grade	Temperature Range (Case)	Order Number
208-pin PQFP	50 MHz	0 – 85°C	P9100–050–PFP

Figure 316. Ordering information