3Dlabs®

PERMEDIA® 2

Hardware Reference Manual

Issue 7

Email: info@3dlabs.com
Web site: http://www.3dlabs.com

3D*labs Inc.*
181 Metro Drive, Suite 520
San Jose, CA 95110
United States

Tel: (408) 436 3455
Fax: (408) 436 3458

3D*labs Ltd.*
Meadlake Place
Thorpe Lea Road, Egham
Surrey, TW20 8HE
United Kingdom

Tel: +44 (0) 1784 470555
Fax: +44 (0) 1784 470699

Change History

Document	Issue	Date	Change
147.1.0	1	1 May 97	Initial issue
147.1.0	2-5	30 July 97	Minor corrections and style changes
147.1.0	6	8 Aug 97	Add section on stereo. Added clock timing information. Added tables for VSN control. Improved information on PLL programming, thermal characteristics, VSAVideo registers. Corrected Fig 9.1 and Pixel and Memory Clock Register diagrams. Added note about unused pins.
147.1.0	7	8 Dec 97	Removed "Reserved" from config register definitions in 2.14. Corrected bit numbering in 2.14.24. Corrected VDD in 8.2, IIL and IIH in 8.2.2, fMClk in 11.1 and epoxy type in 11.4.2. Added AGPSt0-2 to 8.3.2, PCIIntAN to 8.3.3 and 8.3.4, pin types to 9.1, tolerances to 10.1. Added height dimension and corrected number of lead balls. Minor typographical corrections.

Contents

Figures

Tables

1. Introduction

PERMEDIA 2 is a high performance PCI/AGP graphics processor that balances high quality 3D polygon and textured graphics acceleration, windows acceleration and state-of-the-art MPEG1/MPEG2 playback with a fast integrated SVGA core, integrated RAMDAC and video ports.

PERMEDIA 2 sets the standard for 3D and multimedia acceleration, making it the ideal solution to meet the increasingly pervasive need for balanced 3D and multimedia acceleration - and all in a single, low cost PCI device.

PERMEDIA 2 is the second generation PERMEDIA device. Compared with PERMEDIA 1, it provides greater flexibility, additional features and enhanced performance. Throughout this manual the terms PERMEDIA 2 and PERMEDIA are used interchangeably.

2. Functional Overview

2.1 Block Diagram

The major PERMEDIA 2 functional blocks are shown below:

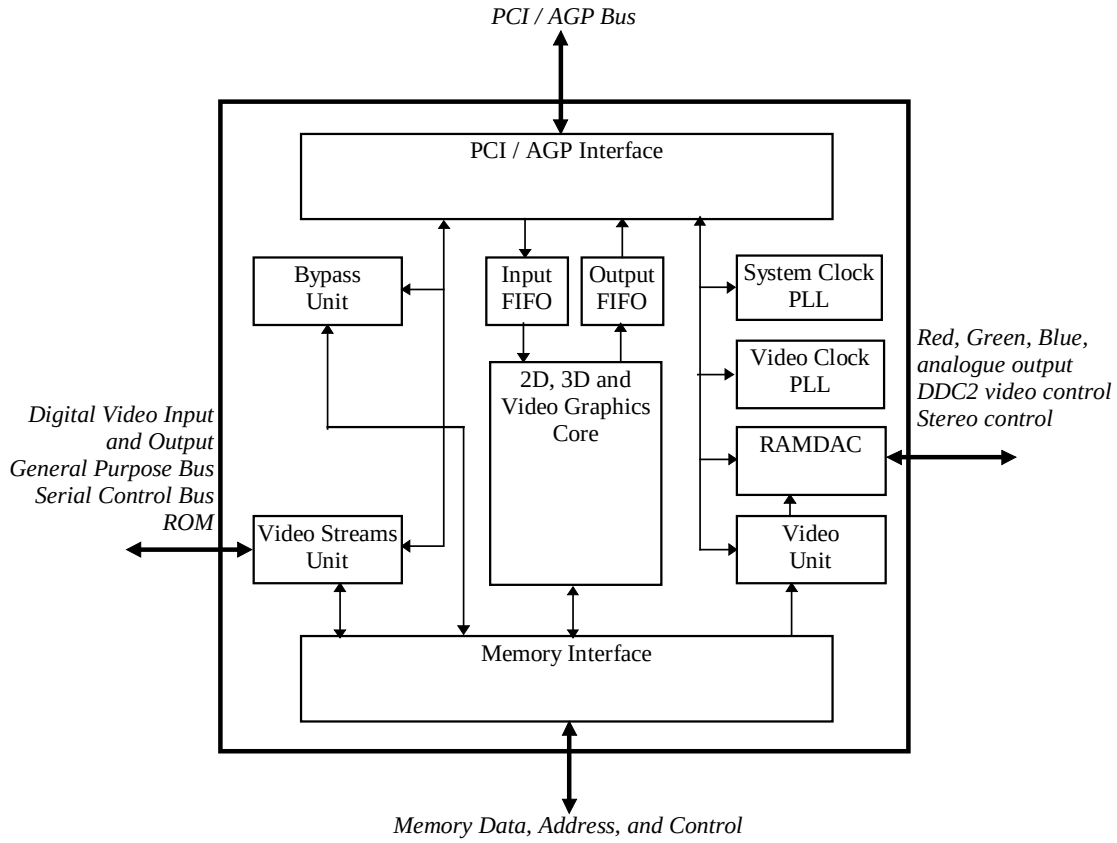


Figure 2.1 PERMEDIA 2 Functional Units

2.2 PCI Interface

The PCI interface conforms to the PCI Local Bus standard Revision 2.1. PERMEDIA 2 is a PCI Local Bus Target, a PCI Local Bus Read Master, and a PCILocal Bus Write Master. It is also an AGP read master with support for pipelined reads and sideband addressing.

The PCI interface has an input FIFO for passing data to the Graphics Core, and an output FIFO for buffering up data to be read from the Graphics Core. The input FIFO is 256 words deep, the output FIFO is 8 words deep. A DMA controller is provided in the PCI interface to allow PERMEDIA to read data directly into the Graphics Core input FIFO or directly out of the output FIFO.

2.3 Graphics Core

The graphics core in PERMEDIA accelerates the key operations for 3D and 2D applications. For further information on the functionality of the graphics core refer to the *PERMEDIA 2 Programmer's Reference Manual.*

2.4 Memory Interface

The local memory is used to store color, depth, stencil, and texture data. For more information on the different data types and their usage refer to the *PERMEDIA 2 Programmer's Reference Manual.*

The memory is organized as 1 to 4 banks of synchronous graphics RAM (SGRAM). Each bank is 64 bits wide and made up of two devices, each 32 bits wide by 256K entries deep. This gives 2Mbytes per bank, with a maximum memory array of 8Mbytes.

Bank zero must always be fitted as the SVGA uses this area for local storage. Any other combination of banks may be fitted, but for contiguous memory banks should be added from 1 to 3.

PERMEDIA will make use of special SGRAM features including block fill and write-per-bit masking. SDRAM may be used in place of SGRAM if it is identical to SGRAM except for missing block write and write per bit masks.

2.5 Video Timing Generation

PERMEDIA has an internal video timing generator and RAMDAC. The RAMDAC has a maximum pixel rate of 230MHz. As all data, including video refreshes, come from the same memory port, the higher video frequencies take away bandwidth that would be used by the graphics processor and reduce performance. Auxiliary Device Support

PERMEDIA can act as a gateway to an additional device which shares address and data lines with the RAMDAC. This device can be mapped into IO space as well as memory space. The auxiliary bus protocol is asynchronous and supports a wait signal that the slave device may use to insert wait states into a transaction.

2.6 Video Streams

PERMEDIA has support for the input and output of digital video. Both these video streams are independent of each other and the video from the RAMDAC. Input video may be used as a texture map for special video effects. Output video is designed to drive TV encoder chips.

Vertical blanking interval data may be separated from the input stream or inserted into the output stream as required.

2.7 Bypass Unit

The bypass unit allows direct access of the memory from the PCI bus. It includes data formatting and a read master DMA controller which is optimized for texture loading.

2.8 Reset Configuration Control

A minimal number of parameters may be configured at reset by resistors attached to the Video Streams data port. Other configuration information should be loaded from a ROM or configuration file following reset. See the Reset Control section for more details.

The PCI Sub-System ID and Sub-System Vendor ID may be loaded directly from the ROM immediately after reset. The information is loaded from the highest address in the ROM.

If a ROM is not fitted these registers are made write-once so that the BIOS or driver software may load them as required.

2.9 ROM support

PERMEDIA supports a Flash ROM. This ROM may store code needed for device-specific initialization and the SVGA BIOS.

2.10 Stereo Support

Stereo functionality on PERMEDIA is controlled via the VideoControl and ScreenBaseRight registers.

The StereoEnable bit in the VideoControl register is used to enable and disable the stereo functionality. When the StereoEnable bit is set to 1, alternate frames are displayed from different framebuffer locations and the VidRightEye pin will indicate which frame is being displayed.

If the RightEyeCtl bit in the VideoControl register is set to 0, then Active High on the VidRightEye pin will indicate that the "Right" frame is being displayed. If the RightEyeCtl bit is set to 1, then Active Low on the VidRightEye pin will indicate that the "Right" frame is being displayed.

The RightFrame bit in the VideoControl register is a read only register which also indicates which frame is being displayed: 0 indicates "Left" frame, 1 indicates "Right" frame.

The ScreenBaseRight register should be loaded with the base address in the framebuffer of the "Right" frame. The base address in the framebuffer of the "Left" frame is given by the standard ScreenBase address.

2.11 Address Map

PERMEDIA has six PCI base address regions:

Region	Description
Configuration	PCI configuration region
0	GC control region
1	Bypass access to memory
2	Bypass access to memory
ROM	Expansion ROM
SVGA	SVGA Addresses

Table 2.1 PCI Address Regions

Two memory apertures are provided, each is a PCI region with a fixed size of 8 MBytes. A variety of access modes are available, including byte swapped, half-word swapped, and packed 16-bit pixel modes to support per-window double buffering with a suitable RAMDAC. Each aperture can be programmed to address the memory controller directly, or to address the memory through the VGA subsystem. The two separately controlled apertures allow different views of the memory to co-exist without register re-programming; for example, one aperture could be set for localbuffer data accesses and the other for framebuffer data accesses.

The two memory apertures can also be programmed to allow reading and writing of the ROM instead of memory. This ensures that the ROM is visible beyond system boot time, making it possible to program a Flash ROM device in situ.

When displaying images in 16-bit per-window double buffered mode the framebuffer area of memory is divided into two interleaved buffers, A and B. Each pixel uses 32 bits: the bottom 16 bits (15:0) form buffer A, and the top 16 bits (31:16) form buffer B. The top bit in Buffer B is used by the RAMDAC to select which buffer is displayed, on a per-pixel basis.

The control registers for each of Memory Apertures One and Two can be set to allow reading and writing of buffers A and B as contiguous 16-bit packed buffers although they are pixel interleaved in the memory. Each 32-bit read or write access over the PCI bus thus transfers two pixels to/from the selected 16-bit packed buffer. The apertures can be programmed to access either buffer A or buffer B, to write to both buffer A and buffer B, or to read from the active buffer as specified by bit 31.

A further control bit is provided to route the memory address via the SVGA controller, rather than directly to the main memory controller. This allows the memory address to be interpreted as a SVGA address. This mechanism allows the SVGA to be relocated away from the standard fixed addresses. The fixed addresses may be optionally disabled so that two SVGA systems can co-exist on the same bus.

2.12 PERMEDIA 2 VGA Modes

The following standard VGA modes are supported:

Mode (hex)	Alpha Format	Char Size	Colors	Max Page	Type Format	Resolution
00 0	40 by 25	8 by 8	16/256K bw	8	Alpha	320 by 200
0*	40 by 25	8 by 14	16/256K bw	8	Alpha	320 by 350
0+	40 by 25	9 by 16	16/256K bw	8	Alpha	360 by 400
01 1	40 by 25	8 by 8	16/256K	8	Alpha	320 by 200
1*	40 by 25	8 by 14	16/256K	8	Alpha	320 by 350
1+	40 by 25	9 by 16	16/256K	8	Alpha	360 by 400
02 2	80 by 25	8 by 8	16/256K bw	8	Alpha	640 by 200
2*	80 by 25	8 by 14	16/256K bw	8	Alpha	640 by 350
2+	80 by 25	9 by 16	16/256K bw	8	Alpha	720 by 400
03 3	80 by 25	8 by 8	16/256K	8	Alpha	720 by 200
3*	80 by 25	8 by 14	16/256K	8	Alpha	640 by 350
3+	80 by 25	9 by 16	16/256K	8	Alpha	720 by 400
04 4	40 by 25	8 by 8	4/256K	1	Graph	320 by 200
05 5	40 by 25	8 by 8	4/256K bw	1	Graph	320 by 200
06 6	80 by 25	8 by 8	2/256K bw	1	Graph	640 by 200
07 7	80 by 25	9 by 14	bw	8	Alpha	720 by 350
7+	80 by 25	9 by 16	bw	8	Alpha	720 by 400
0D D	40 by 25	8 by 8	16/256K	8	Graph	320 by 200
0E E	80 by 25	8 by 8	16/256K	4	Graph	640 by 200
0F F	80 by 25	8 by 14	bw	2	Graph	640 by 350
10 10	80 by 25	8 by 14	16/256K	2	Graph	640 by 350
11 11	80 by 30	8 by 16	2/256K	1	Graph	640 by 480
12 12	80 by 30	8 by 16	16/256K	1	Graph	640 by 480
13 13	40 by 25	8 by 8	256/256K	1	Graph	320 by 200

Table 2.2 Standard VGA modes

The following VESA SVGA modes are supported:

Mode (hex)	Pixels	Colors
100	640 by 400	256
101	640 by 480	256

Table 2.3 VESA/SVGA modes

ModeX is also supported.

2.13 PCI Configuration Region

The PCI Configuration Region provides information which satisfies the needs of current and anticipated system configuration mechanisms.

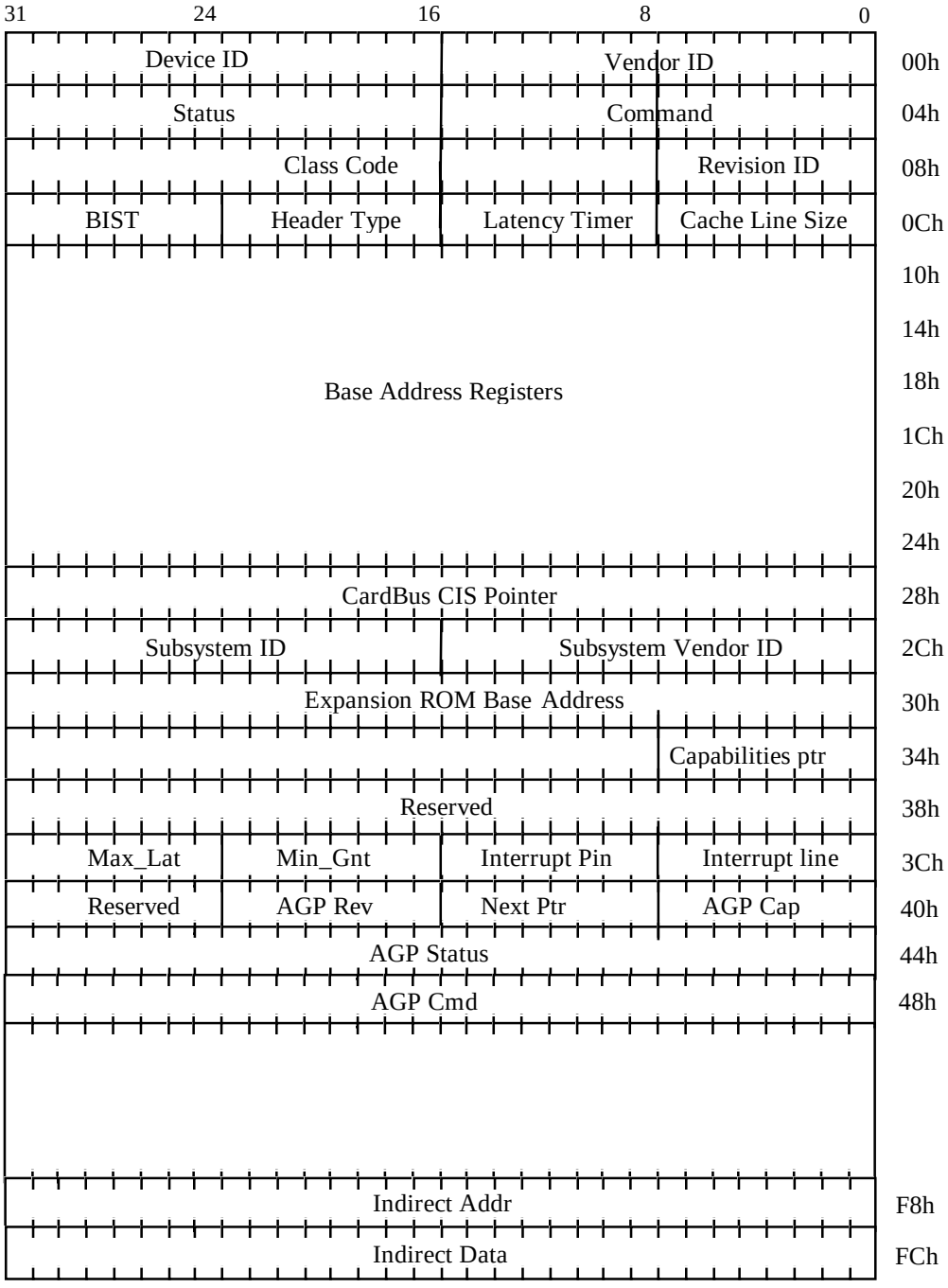


Figure 2.1 PCI Configuration Region

2.14 PCI Register Set

For more information about the use of the registers in this section refer to the PCI Specification.

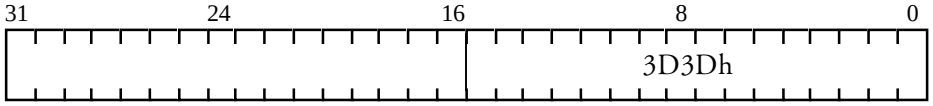
2.14.1 Vendor ID

Vendor identification number.

CFGVendorId

Region: Configuration Read Only

Offset: 00h Reset Value: 3D3Dh



Bits 0-15 3D3Dh
3Dlabs company code

Bits 16-31 See CFGDeviceId

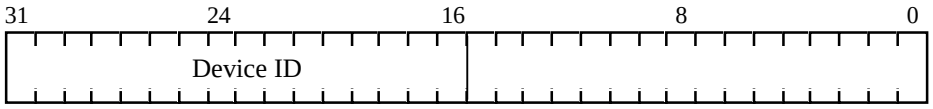
2.14.2 Device ID

Device identification number.

CFGDeviceId

Region: Configuration Read Only

Offset: 02h Reset Value: 0004h



Bits 0-15 See CFGVendorId

Bits 16-31 0007h
PERMEDIA Device number

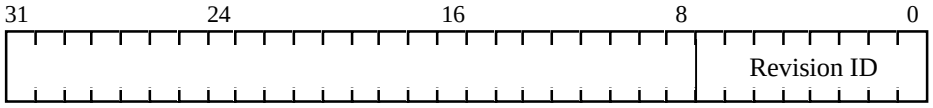
2.14.3 Revision ID

Revision identification number.

CFGRevisionId

Region: Configuration Read Only

Offset: 08h Reset Value: Revision Number



Bits 0-7 Revision ID 01h

Bits 8-31 See CFGClassCode

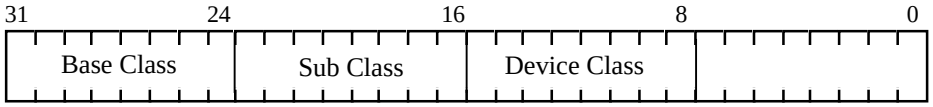
2.14.4 Class Code Register

This register is used to identify the generic function of PERMEDIA, which depends on the setting of configuration data detailed below.

CFGClassCode

Region: Configuration Read

Offset: 09h Reset Value: from configuration data



Bits 0-7 See CFGRevisionId

Bits 8-15 00h
See table below.

Bits 16-23 80h
See table below.

Bits 24-31 03h
See table below.

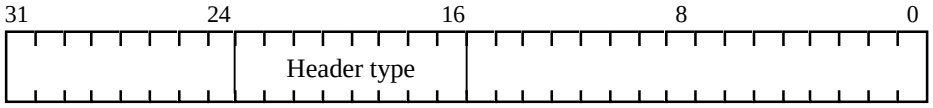
BaseClassZero (config bit)	Fixed VGA Addressing	Bass Class	Sub Class	Inter face	Meaning (see PCI Spec Appendix D)
0	Disabled	03h	80h	00h	"Other" display controller
0	Enabled	03h	01h	00h	VGA-compatible controller
1	Disabled	00h	00h	00h	Non VGA-compatible device
1	Enabled	00h	01h	00h	VGA-compatible device

2.14.5 Header Type

CFGHeaderType

Region: Configuration Read

Offset: 0Eh Reset Value: 00h



Bits 0-15 See CFGLatTimer and CFGCacheLine

Bits 16-23 00h
Header Type. PCI Definition: Single function device

Bits 24-31 See CFGBist

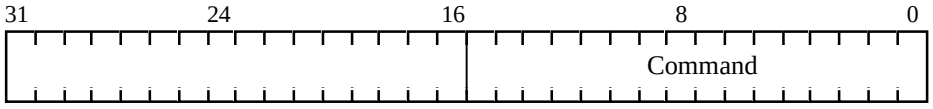
2.14.6 Command Register

The command register provides control over a device's ability to generate and respond to PCI cycles. Writing zero to this register disconnects the device from the PCI for all except configuration accesses. PERMEDIA 2 supports all necessary bits within the command register for the functionality it contains.

CFGCommand

Region: Configuration Read/Write

Offset: 04h Reset Value: 00h



Bit 0 I/O Space Enable
0 = Disable I/O space accesses.
1 = Enable I/O space accesses.
If fixed SVGA addressing is disabled this bit will be 0 (read-only).

Bit 1 Memory Space Enable
0 = Disable memory space accesses.
1 = Enable memory space accesses.

Bit 2 Bus Master Enable
0 = Disable master accesses.
1 = Enable master accesses.

Bit 3 Special Cycle Enable (Read Only)
0 = PERMEDIA never responds to special cycle accesses.

Bit 4 Memory Write and Invalidate Enable (Read Only)
0 = "Memory Write and Invalidate" is never generated.

Bit 5 SVGA Palette Snoop Enable
0 = Treat palette accesses like all other SVGA accesses.
1 = Enable SVGA palette snooping.
If fixed SVGA addressing is disabled, this bit is 0 (read-only).

Bit 6 Parity Error Response enable (Read Only)
0 = PERMEDIA does not support parity error reporting.

Bit 7 Address/Data stepping enable (Read Only)
0 = PERMEDIA does not perform stepping.

Bit 8 SERR driver enable (Read Only)
0 = PERMEDIA does not support parity error reporting.

Bit 9 Master Fast Back-to-Back Enable (Read Only)
0 = The PERMEDIA master does not do fast back-to-back accesses.

Bits 10-15 Reserved (Read Only)
00.0000b.

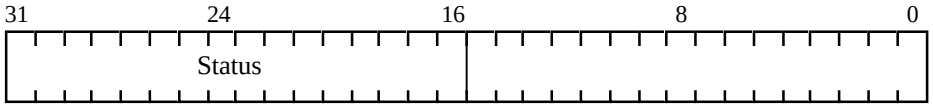
Bits 16-31 See CFGStatus.

2.14.7 Status Register

CFGStatus

Region: Configuration Read

Offset: 06h Reset Value: 00h



Bits 0-15	See CFGCommand	
Bits 16-20	Reserved	(Read Only)
Bit 21	66 MHz Capable 0 = PERMEDIA is 33 MHz capable only.	(Read Only)
Bit 22	UDF Supported 0 = PERMEDIA does not support user-definable configurations.	(Read Only)
Bit 23	Fast Back-to-Back Capable 1 = PERMEDIA can accept fast back-to-back PCI transactions.	(Read Only)
Bit 24	Data Parity Error Detected 0 = Parity checking not implemented on PERMEDIA .	(Read Only)
Bits 25-26	DEVSEL Timing 01b = PERMEDIA asserts DEVSEL# at medium speed.	(Read Only)
Bit 27	Signaled Target Abort 0 = PERMEDIA never signals Target-Abort.	(Read Only)
Bit 28	Received Target Abort This bit is set by the PERMEDIA bus master whenever its transaction is terminated with Target-Abort.	
Bit 29	Received Master Abort This bit is set by the PERMEDIA bus master whenever its transaction is terminated with Master-Abort.	
Bit 30	Signaled System Error 0 = PERMEDIA never asserts a system error.	(Read Only)
Bit 31	Detected Parity Error 0 = Parity checking is not implemented by PERMEDIA. Writes to this register cause bits to be reset, but not set. A bit is reset whenever the register is loaded with the corresponding bit position set to one.	(Read Only)

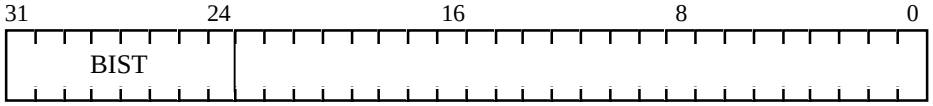
2.14.8 BIST

Optional register used for control and status of BIST.

CFGBist

Region: Configuration Read

Offset: 0Fh Reset Value: 00h



Bits 0-23 See CFGLatTimer and CFGCacheLine

Bits 24-31 BIST
00h. BIST unsupported by PERMEDIA over the PCI interface.

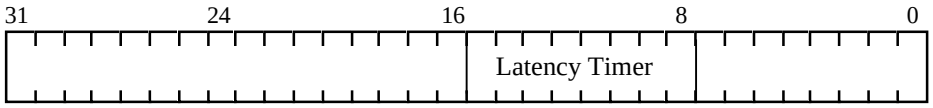
2.14.9 Latency Timer

This register specifies, in PCI bus clocks, the value of the Latency Timer for this PCI bus master.

CFGLatTimer

Region: Configuration Read/Write

Offset: 0Dh Reset Value: 00h



Bits 0-7 See CFGCacheLine

Bits 8-15 Latency Timer Count
Sets the maximum number of PCI clock cycles for master burst accesses.

Bits 16-31 See CFGBist and CFGHeaderType

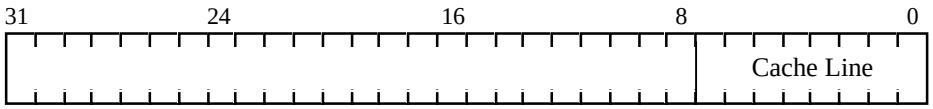
2.14.10 Cache Line Size

This register specifies the cache line size in units of 32 bit words. It is only implemented for masters which use the 'Memory write and invalidate' command. PERMEDIA 2 does not use this command.

CFGCacheLine

Region: Configuration Read

Offset: 0Ch Reset Value: 00h



Bits 0-7 Cache Line Size
00h. Cache line size unsupported.

Bits 8-31 See CFGBist, CFGHeaderType and CFGLatTimer

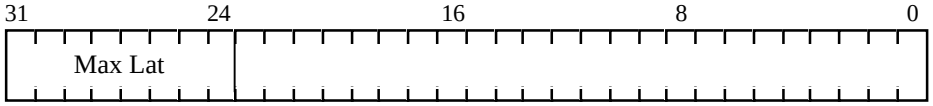
2.14.11 Maximum Latency

This register specifies how often the PCI device needs to gain access to the PCI bus.

CFGMaxLat

Region: Configuration Read

Offset: 3Fh Reset Value: from configuration data



Bits 0-23 See CFGMinGrant, CFGIntPin and CFGIntLine

Bits 24-31 Maximum Latency
Possible values are: 00h, 40h, 80h, and C0h.

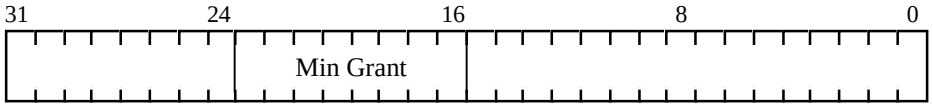
2.14.12 Minimum Grant

This register specifies how long a burst period the PCI device needs.

CFGMinGrant

Region: Configuration Read

Offset: 3Eh Reset Value: from configuration data



Bits 0-15 See CFGIntPin and CFGIntLine

Bits 16-23 Minimum Grant
Possible values are: 00h, 40h, 80h, and C0h.

Bits 24-31 See CFGMaxLat

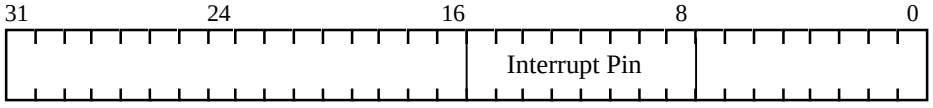
2.14.13 Interrupt Pin

The Interrupt Pin register specifies which line PERMEDIA uses.

CFGIntPin

Region: Configuration Read

Offset: 3Dh Reset Value: 01h



Bits 0-7 See CFGIntLine

Bits 8-15 Interrupt Pin
01h PERMEDIA uses Interrupt pin INTAN

Bits 16-31 See CFGMinGrant and CFGMaxLat

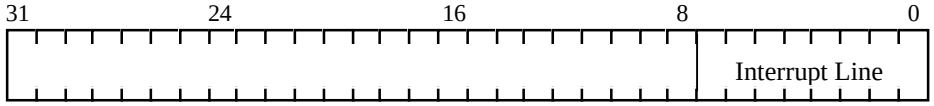
2.14.14 Interrupt Line

The Interrupt Line register in an 8-bit register used to communicate interrupt line routing information.

CFGIntLine

Region: Configuration Read/Write

Offset: 3Ch Reset Value: 00h



Bits 0-7 Interrupt Line

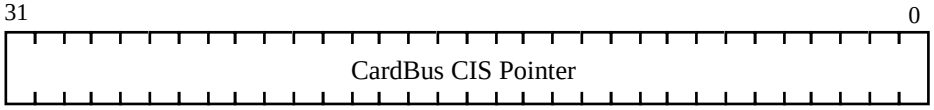
Bits 8-31 See CFGMinGrant, CFGIntPin and CFGMaxLat

2.14.15 CardBus CIS Pointer

CFGCardBus

Region: Configuration Read

Offset: 28h Reset Value: 00h



Bits 0-31 CardBus Pointer
0000.0000h = Not implemented.

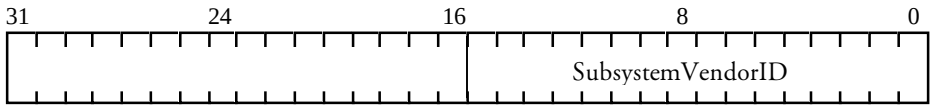
2.14.16 Subsystem Vendor ID

This register is used to identify the vendor of the add-in board on which the PERMEDIA device resides. It has two possible reset states: the value may be loaded from the ROM byte addresses FFFCh and FFFDh, or reset to the Vendor ID and then written to once before it becomes read only. The option is controlled by a configuration resistor.

CFGSubsystemVendorId

Region: Configuration Write Once

Offset: 02Ch Reset Value: configured



Bits 0-15 SubsystemVendorID

Bits 16-31 See CFGSubsystemId

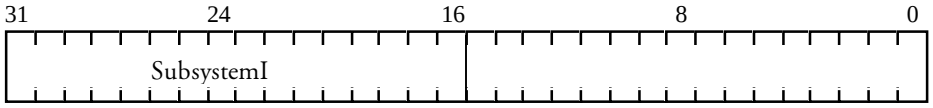
2.14.17 Subsystem ID

This register is used to identify the add-in board on which the PERMEDIA device resides. It has two possible reset states: the value may be loaded from the ROM byte addresses FFFEh and FFFFh, or reset to the Device ID and then written to once before it becomes read only. The option is controlled by a configuration resistor.

CFGSubsystemId

Region: Configuration Write Once

Offset: 02Eh Reset Value: configured



Bits 0-15 See CFGSubsystemVendorID

Bits 16-31 Subsystem ID

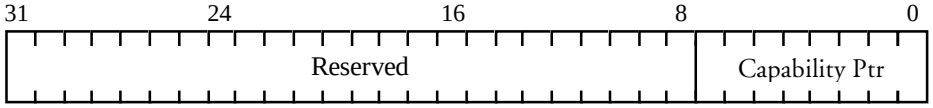
2.14.18 Capabilities Pointer

The Capabilities Pointer register is an eight bit register used to provide an offset into the configuration space for the first item a capabilities list. In an AGP system, it is used to point to the AGP capability registers.

CFGCapPtr

Region: Configuration Read Only

Offset: 034h Reset Value: configured



Bits 0-7 Capability Ptr
Configured by AGPCapable
00h when AGPCapable = 0
40h when AGPCapable = 1

Bits 8-31 Reserved

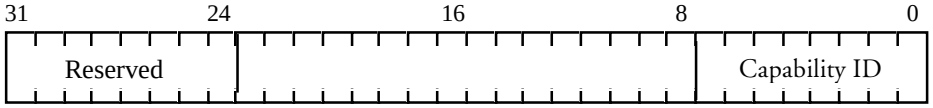
2.14.19 Capability ID

This register specifies that the device has AGP capability.

CFGCapID

Region: Configuration Read Only

Offset: 040h Reset Value: configured



Bits 0-7 Capability ID
Configured by AGPCapable
00h when AGPCapable = 0
02h when AGPCapable = 1

Bits 8-31 CFGNextPtr, CFGAGPRev and Reserved

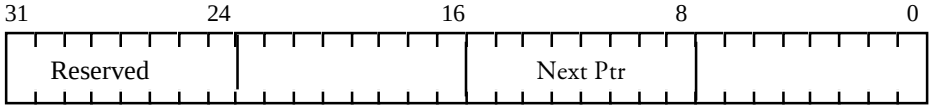
2.14.20 Next Pointer

This register points to the next capability data structure. However as there are no more, it is set to zero.

CFGNextPtr

Region: Configuration Read Only

Offset: 041h Reset Value: 00h



Bits 0-7 See CFGCapID

Bits 8-15 Next Ptr
00h = no further capabilities in list

Bits 16-31 CFGAGPRev and Reserved

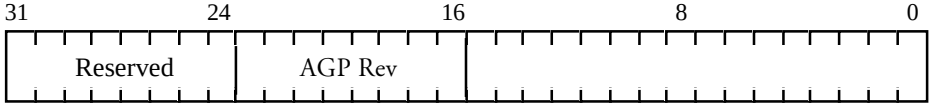
2.14.21 AGP Revision

This register reports the revision of the AGP specification that the device conforms to.

CFGAGPRev

Region: Configuration Read Only

Offset: 042h Reset Value: configured



Bits 0-15 See CFGCapID and CFGNextPtr

Bits 16-19 Minor Rev
Configured by AGPCapable

0h when AGPCapable = 0
0h when AGPCapable = 1

Bits 20-23 Major Rev
Configured by AGPCapable
0h when AGPCapable = 0
1h when AGPCapable = 1

Bits 24-31 Reserved

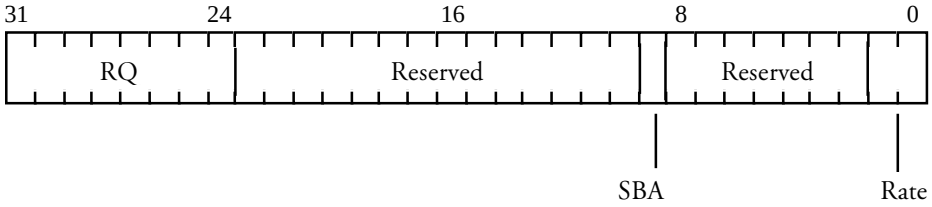
2.14.22 AGP Status

This register describes the AGP capabilities of the device.

CFGAGPStatus

Region: Configuration Read Only

Offset: 044h Reset Value: configured



Bits 0-1 Rate
Configured by AGPCapable
0h when AGPCapable = 0 or SBACapable = 0
1h when AGPCapable = 1 and SBACapable = 1

Bits 2-8 Reserved

Bit 9 SBA
Configured by AGPCapable
0h when AGPCapable = 0
1h when AGPCapable = 1

Bits 10-23 Reserved

Bits 24-31 RQ
Maximum number of AGP requests supported
Configured by AGPCapable
00h when AGPCapable = 0
1Fh when AGPCapable = 1

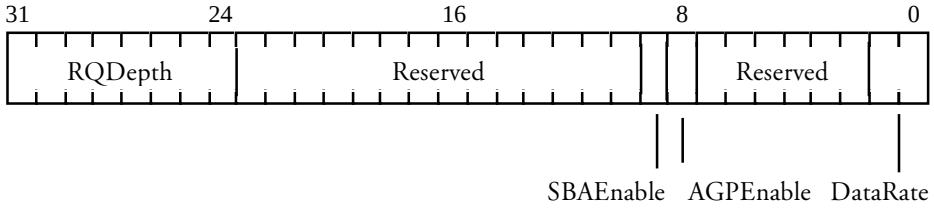
2.14.23 AGP Command

This register controls the operation of the AGP interface.

CFGAGPCommand

Region: Configuration Read/Write

Offset: 048h Reset Value: 00000000h



Bits 0-1	DataRate 0h = AGP disabled 1h = 1X transfer rate
Bits 2-7	Reserved
Bit 8	AGPEnable 0h = AGP mastering disabled 1h = AGP mastering enabled
Bit 9	SBAEnable 0h = sideband addressing disabled 1h = sideband addressing enabled
Bits 10-23	Reserved
Bits 24-31	RQDepth Maximum number of AGP requests which can be queued.

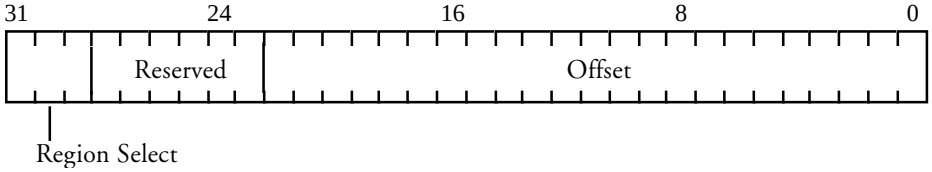
2.14.24 Indirect Address

This register and the Indirect Data register are used to access regions 0, 1, and 2, and the ROM region indirectly. The region and offset to be accessed are loaded into the Indirect Address register and the Indirect Data register read from or written to as appropriate.

CFGIndirectAddress

Region: Configuration Read/Write

Offset: 0F8h Reset Value: 00000000h



Bits 0-22	Offset within the region.
Bits 23-28	Reserved
Bits 29-31	Region Select 0 = Region 0 1 = Region 1 2 = Region 2 3-6 Reserved 7 = ROM Region

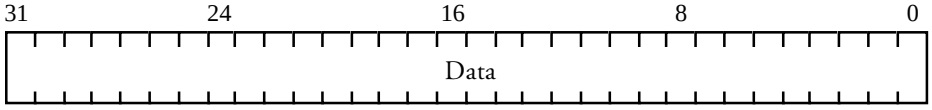
2.14.25 Indirect Data

This register and the Indirect Address register are used to access regions 0, 1, and 2, and the ROM region indirectly. The region and offset to be accessed are loaded into the Indirect Address register and the Indirect Data register read from or written to as appropriate.

CFGIndirectData

Region: Configuration Read/Write

Offset: 0FCh Reset Value: 00000000h



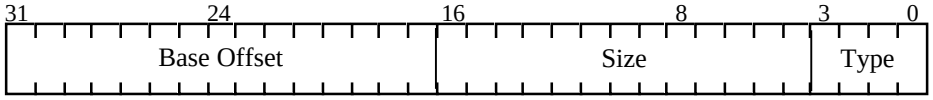
2.14.26 Base Address 0 Register

The Base Address 0 Register contains the PERMEDIA 2 control space offset. The control registers are in memory space. They are not prefetchable and can be located anywhere in 32 bit address space.

CFGBaseAddr0

Region: Configuration Read/Write

Offset: 10h Reset Value: 0000.0000h



Bits 0-3 Address Type Read Only
0h Memory Space, not prefetchable, in 32 bit address space

Bits 4-16 Size indication Read Only
000h Indicates that the control registers must be mapped into 128KBytes.

Bits 17-31 Base offset
Loaded at boot time to set offset of the control register space.

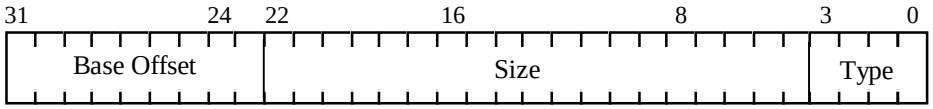
2.14.27 Base Address 1 Register

The Base Address 1 Register contains the PERMEDIA aperture one memory offset. It is not prefetchable and can be located anywhere in 32 bit address space.

CFGBaseAddr1

Region: Configuration Read/Write

Offset: 14h Reset Value: 0000.0000h



Bits 0-3 Address Type Read Only
0h Memory Space, not prefetchable, in 32 bit address space

Bits 4-22 Size indication Read Only
Fixed at 8Mbytes.

Bits 23-31 Base offset
Loaded at boot time to set offset of the memory space.

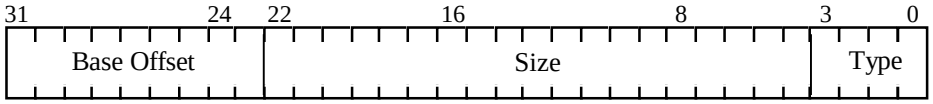
2.14.28 Base Address 2 Register

The Base Address 2 register contains PERMEDIA aperture two memory offset. It is not prefetchable and can be located anywhere in 32 bit address space.

CFGBaseAddr2

Region: Configuration Read/Write

Offset: 18h Reset Value: 0000.0000h



Bits 0-3 Address Type Read Only
0h Memory Space, not prefetchable, in 32 bit address space.

Bits 4-22 Size indication Read Only
Fixed at 8Mbytes.

Bits 23-31 Base offset
Loaded at boot time to set offset of the memory space.

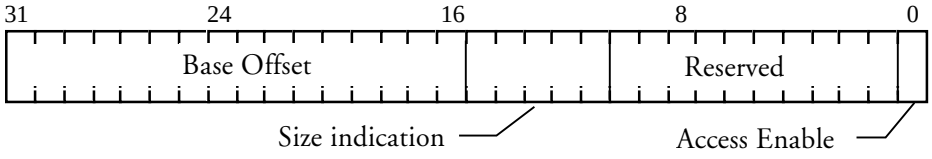
2.14.29 Expansion ROM Base Address

The Expansion ROM Base register is the offset address for the expansion ROM.

CFGRomAddr

Region: Configuration Read/Write

Offset: 30h Reset Value: 0000.0000h



Bit 0 Access enable
0 = Expansion ROM accesses disabled
1 = Expansion ROM accesses enabled

Bits 1-10 Reserved Read Only
000h PCI reserved register bits.

Bits 11-15 Size indication Read Only
00h Indicates that the control registers must be mapped into 64KBytes.

Bits 16-31 Base offset
Loaded at boot time to set offset of the Expansion ROM.

3. Region 0 - Registers

3.1 Region 0 Address Map

The PERMEDIA Region Zero is a 128Kbyte region containing the control registers, and ports to and from the graphics processor. The control space is mapped in twice within the 128KByte region. In the second 64K the registers are mapped to be byte swapped for big endian hosts.

Address Range	Region Select	Byte Swap
0000.0000 -> 0000.1FFF	Control Status	No
0000.1000 -> 0000.1FFF	Memory Control	No
0000.2000 -> 0000.2FFF	GP FIFO Access	No
0000.3000 -> 0000.3FFF	Video Control	No
0000.4000 -> 0000.4FFF	RAMDAC	No
0000.5000 -> 0000.57FF	Video Stream GP Bus	No
0000.5800 -> 0000.5FFF	Video Stream Control	No
0000.6000 -> 0000.6FFF	VGA Control	No
0000.7000 -> 0000.7FFF	Reserved	No
0000.8000 -> 0000.FFFF	GP Registers	No
0001.0000 -> 0001.1FFF	Control Status	Yes
0001.1000 -> 0001.1FFF	Memory Control	Yes
0001.2000 -> 0001.2FFF	GP FIFO Access	Yes
0001.3000 -> 0001.3FFF	Video Control	Yes
0001.4000 -> 0001.4FFF	RAMDAC	Yes
0001.5000 -> 0001.57FF	Video Stream GP Bus	Yes
0001.5800 -> 0001.5FFF	Video Stream Control	Yes
0001.6000 -> 0001.6FFF	VGA Control	Yes
0001.7000 -> 0001.7FFF	Reserved	Yes
0001.8000 -> 0001.FFFF	GP Registers	Yes

Table 3.1 Region 0 Address Map

3.2 Control Status Registers

3.2.1 Reset Status Register

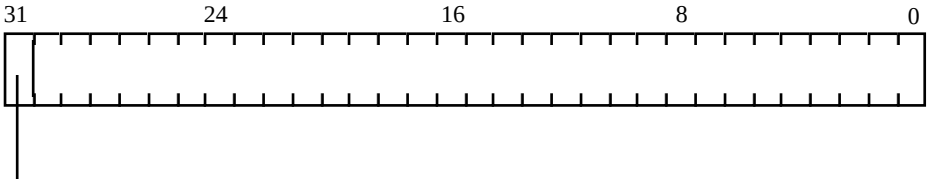
Writing to the reset status register forces a software reset of the PERMEDIA Graphics Processor. The software reset does not reset the PCI interface but is otherwise the same as a hardware reset.

The software reset takes a number of cycles and the Graphics Processor must not be used during the reset. A flag in the register is provided which shows that the software reset is still in progress.

ResetStatus

Region: Zero Read/Write

Offset: 0000.0000h Reset Value: 0000.0000h



Software Reset Flag

Bits 0-30	Reserved
Bit 31	Software reset flag 0 The GC is ready for use 1 The GC is being reset and must not be used

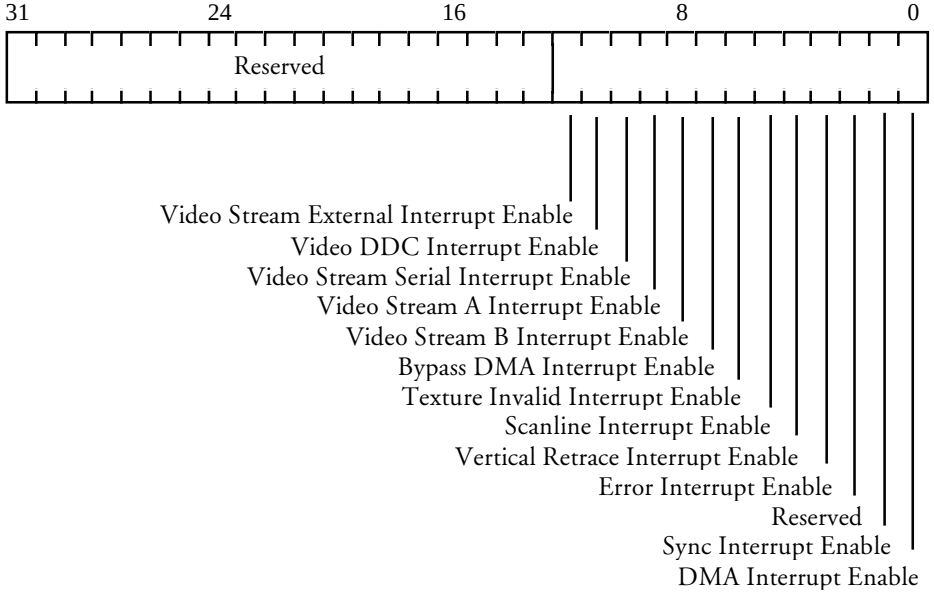
3.2.2 Interrupt Enable Register

The Interrupt Enable Register allows for a number of PERMEDIA flags to generate a PCI interrupt. At reset all interrupts sources are disabled.

IntEnable

Region: Zero Read/Write

Offset: 0000.0008h Reset Value: 0000.0000h



Bit 0 DMA interrupt enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 1 Sync interrupt enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 2 Reserved
Read as zero

Bit 3 Error interrupt enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 4 Vertical retrace interrupt enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 5 Scanline interrupt enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 6 Texture Invalid Interrupt Enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 7 Bypass DMA Interrupt Enable
- 0 Disable interrupt
- 1 Enable interrupt

Bit 8	Video Stream B Interrupt Enable 0 Disable interrupt 1 Enable interrupt
Bit 9	Video Stream A Interrupt Enable 0 Disable interrupt 1 Enable interrupt
Bit 10	Video Stream Serial Interrupt Enable 0 Disable interrupt 1 Enable interrupt
Bit 11	Video DDC Interrupt Enable 0 Disable interrupt 1 Enable interrupt
Bit 12	Video Stream External Interrupt Enable 0 Disable interrupt 1 Enable interrupt
Bits 13-31	Reserved Read as zero

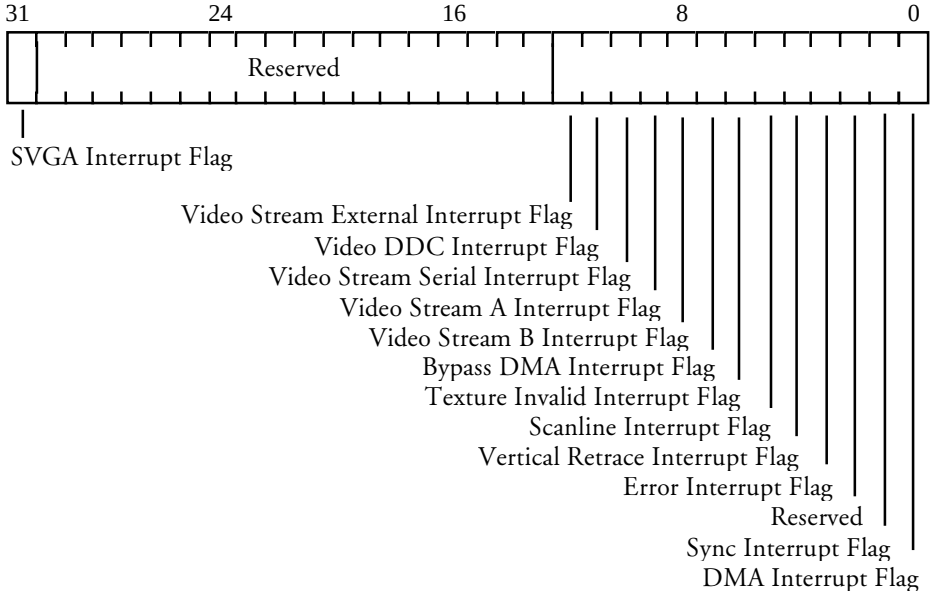
3.2.3 Interrupt Flags

The Interrupt Flags Register shows which interrupts are outstanding. Flag bits are reset by writing to this register with the corresponding bit set to a one. Flags at positions where the bits are set to zero will be unaffected by the write. (The exception is bit 31, which is read-only and reflects the state of the interrupt line from the SVGA Unit. The SVGA Interrupt must be enabled and reset by accessing the SVGA Unit directly, but is visible in this register for convenience.)

IntFlags

Region: Zero Read/Write

Offset: 0000.0010h Reset Value: 0000.0000h



Bit 0	DMA interrupt Flag 0 No interrupt 1 Interrupt outstanding
Bit 1	Sync interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 2	Reserved Read as zero
Bit 3	Error interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 4	Vertical retrace interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 5	Scanline interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 6	Texture Invalid Interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 7	Bypass DMA Interrupt Flag 0 No interrupt 1 Interrupt outstanding
Bit 8	Video Stream B Interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 9	Video Stream A Interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 10	Video Stream Serial Interrupt flag 0 No interrupt 1 Interrupt outstanding
Bit 11	Video DDC Interrupt Flag 0 No interrupt 1 Interrupt outstanding
Bit 12	Video Stream External Interrupt flag 0 No interrupt 1 Interrupt outstanding
Bits 13-30	Reserved Read as zero
Bit 31	SVGA Interrupt flag 0 No interrupt 1 Interrupt asserted

3.2.4 Input FIFO Space Register

The input FIFO space register indicates the number of words that can currently be written to the input FIFO. This register can be read at any time and used to allow the controlling software to efficiently send data to the PERMEDIA . If the DMA controller for the FIFO is in use, the value read is a snapshot of the current FIFO status.

InFIFOSpace

Region: Zero Read

Offset: 0000.0018h Reset Value: 0000.0020h

Bits 0-31 Input FIFO Space
The number of empty words in the input FIFO. This number of words can be written before checking again for FIFO space availability.

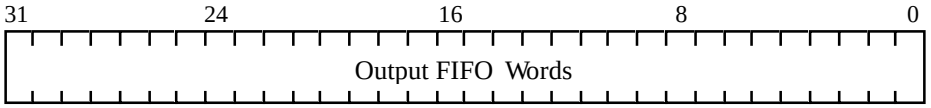
3.2.5 Output FIFO Words Register

The output FIFO words register indicates the number of words currently in the output FIFO. This register can be read at any time and used to allow the controlling software to efficiently read output data from the PERMEDIA .

OutFIFOWords

Region: 0 Read

Offset: 0000.0020h Reset Value: 0000.0000h



Bits 0-31 Output FIFO Words
The number of valid words in the output FIFO. This number of words can be read before checking for more words.

3.2.6 In DMA Start Address

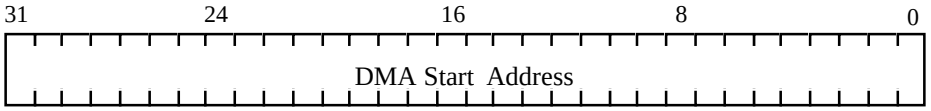
The DMA address should be loaded with the first PCI address for the buffer to be transferred to the GC when using the DMA controller.

Writing to the DMA count register loads the address into the DMA counter. Once a DMA has been set off the next DMA start address may be loaded. A read of this register returns the last start value loaded even if the DMA is underway.

DMAAddress

Region: Zero Read/Write

Offset: 0000.0028h Reset Value: 0000.0000h



Bits 0-31 DMA Start Address
PCI start address for PCI master read transfer to the Graphics Core.

3.2.7 In DMA Count

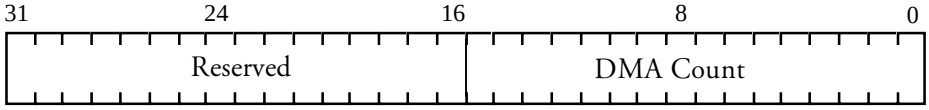
The DMA count register should be loaded with the number of words to be transferred in the DMA operation. The action of loading a word count greater than zero sets off the DMA operation. The value read back from this register indicates the current number of words left to be transferred.

This register should only be written to if the count is zero. It can be read at any time.

DMACount

Region: Zero Read/Write

Offset: 0000.0030h Reset Value: 0000.0000h



Bits 0-15 DMA Count
Valid Range: 0 to 65535
Number of words to be transferred in DMA operation.
Undefined action if this register is written to when it is not zero.

Bits 16-31 Reserved

3.2.8 Error Flags Register

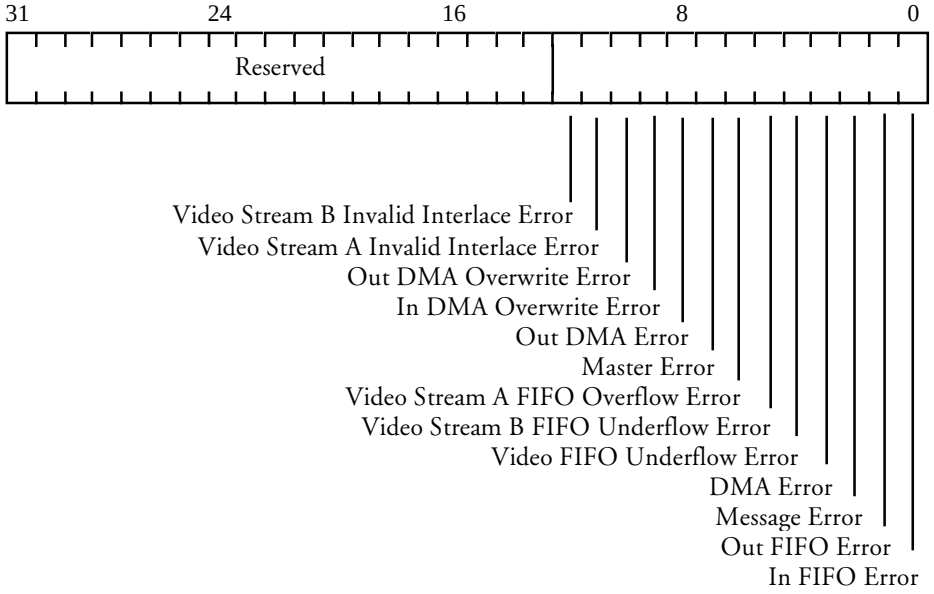
The Error Flags Register shows which errors are outstanding on PERMEDIA .

Flag bits are reset by writing to this register with the corresponding bit set to a 1. Flags at positions where the bits are set to 0 will be unaffected by the write.

ErrorFlags

Region: Zero Read/Write

Offset: 0000.0038h Reset Value: 0000.0000h



Bit 0 Input FIFO Error Flag
Flag set on write to full input FIFO
0 No error
1 Error outstanding

Bit 1 Output FIFO Error Flag
Flag set on read from empty output FIFO
0 No error
1 Error outstanding

Bit 2 Message Error Flag
Flag set on incorrect mixing of accesses to the input FIFO space and the GC register space
0 No error
1 Error outstanding

Bit 3 DMA Error Flag
Flag set for direct or register access to input FIFO while DMA is in progress.
0 No error
1 Error outstanding

Bit 4 Video FIFO Underflow Error
0 No error
1 Error outstanding

Bit 5	Video Stream B FIFO Underflow Error 0 No error 1 Error outstanding
Bit 6	Video Stream A FIFO Overflow Error 0 No error 1 Error outstanding
Bit 7	Master Error Set if master abort or target abort occurs as a consequence of a master access. 0 No error 1 Error outstanding
Bit 8	Out DMA Error Set if slave access is made to output FIFO while DMA is in progress (i.e. OutDMACount is not zero). 0 No error 1 Error outstanding
Bit 9	In DMA Overwrite Error Set if InDMACount register is written when it is not zero. 0 No error 1 Error outstanding
Bit 10	Out DMA Overwrite Error Set if OutDMACount register is written when it is not zero. 0 No error 1 Error outstanding
Bit 11	Video Stream A Invalid Interlace Error Set for invalid sequence of fields. 0 No error 1 Error outstanding
Bit 12	Video Stream B Invalid Interlace Error Set for invalid sequence of fields. 0 No error 1 Error outstanding
Bits 13-31	Reserved

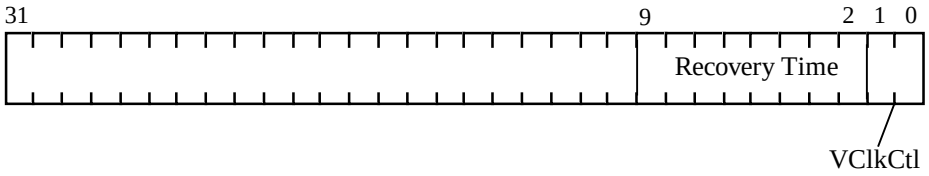
3.2.9 Video Clock Control Register

Vidctl 0 and 1 select the DClk PLL registers to use. An eight-bit field is provided in this register to program the number of PCI clocks to be counted between each RAMDAC access.

VClkCtl

Region: Zero Read/Write

Offset: 0000.0040h Reset Value: 0000.0000h



Bit 0	VidCtl(0)
Bit 1	VidCtl(1)
Bits 2-9	Recovery Time Number of PCI clocks to count between RAMDAC accesses.
Bits 10-32	Reserved - Read as zero (Read Only)

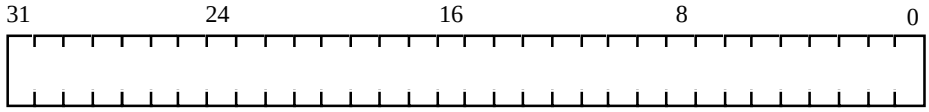
3.2.10 Test Register

Writes to this register have an undefined effect.

TestRegister

Region: Zero Read/Write

Offset: 0000.0048h Reset Value: 0000.0000h

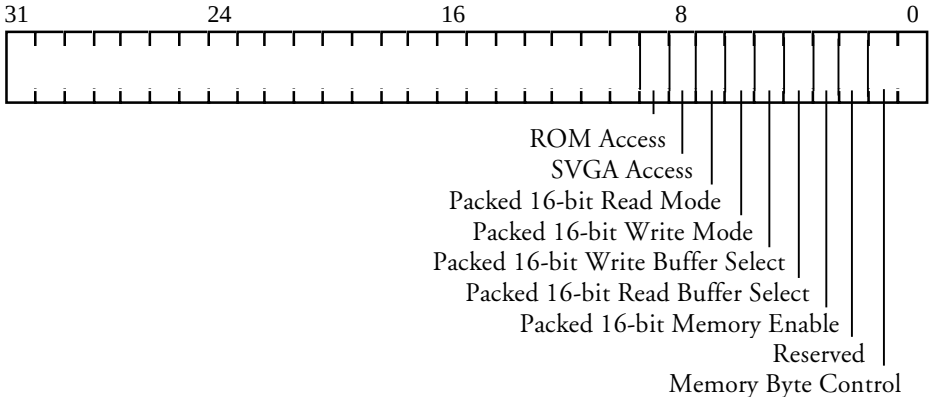


3.2.11 Aperture 1 Control Register

ApertureOne

Region: Zero Read/Write

Offset: 0000.0050h Reset Value: 0000.0000h



Bits 0-1 Memory Byte Control
- 0 Standard.
- 1 Byte Swapped.
- 2 Half Word Swapped.
- 3 Reserved.

Bit 2 Reserved (Read Only)

Bit 3 Packed 16-bit (1:5:5:5) Memory Enable
- 0 Disable packed 16-bit mode.
- 1 Enable packed 16-bit mode.

Bit 4 Packed 16-bit Read Buffer Select
- 0 Select Buffer A for Read Accesses.
- 1 Select Buffer B for Read Accesses.

Bit 5 Packed 16-bit Write Buffer Select
- 0 Select Buffer A for Write Accesses.
- 1 Select Buffer B for Write Accesses.

Bit 6 Packed 16-bit Write Mode
- 0 Disable double writes.
- 1 Enable double writes.

Bit 7 Packed 16-bit Read Mode
- 0 Read buffer selected by Bit 4 of this register.
- 1 Read buffer selected by memory contents (bit 31).

Bit 8 SVGA Access
- 0 Address memory controller directly.
- 1 Address memory through SVGA subsystem.

Bit 9 ROM Access
- 0 Use this aperture to access memory (SVGA or direct).
- 1 Use this aperture to access the Expansion ROM.

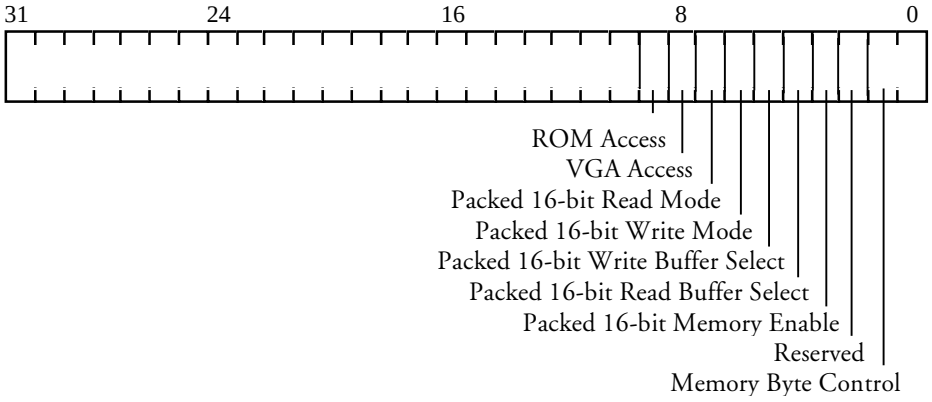
Bits 10-31 Reserved (all bits zero) (Read Only)

3.2.12 Aperture 2 Control Register

ApertureTwo

Region: Zero Read/Write

Offset: 0000.0058h Reset Value: 0000.0000h



Bits 0-1 Memory Byte Control
- 0 Standard.
- 1 Byte Swapped.
- 2 Half Word Swapped.
- 3 Reserved.

Bit 2 Reserved (Read Only)

Bit 3 Packed 16-bit (1:5:5:5) Memory Enable
- 0 Disable packed 16-bit mode.
- 1 Enable packed 16-bit mode.

Bit 4 Packed 16-bit Read Buffer Select
- 0 Select Buffer A for Read Accesses.
- 1 Select Buffer B for Read Accesses.

Bit 5 Packed 16-bit Write Buffer Select
- 0 Select Buffer A for Write Accesses.
- 1 Select Buffer B for Write Accesses.

Bit 6 Packed 16-bit Write Mode
- 0 Disable double writes.
- 1 Enable double writes.

Bit 7 Packed 16-bit Read Mode
- 0 Read buffer selected by Bit 4 of this register.
- 1 Read buffer selected by memory contents (bit 31).

Bit 8 SVGA Access
- 0 Address memory controller directly.
- 1 Address memory through SVGA subsystem.

Bit 9 ROM Access
- 0 Use this aperture to access memory (SVGA or direct).
- 1 Use this aperture to access the Expansion ROM.

Bits 10-31 Reserved (all bits zero) (Read Only)

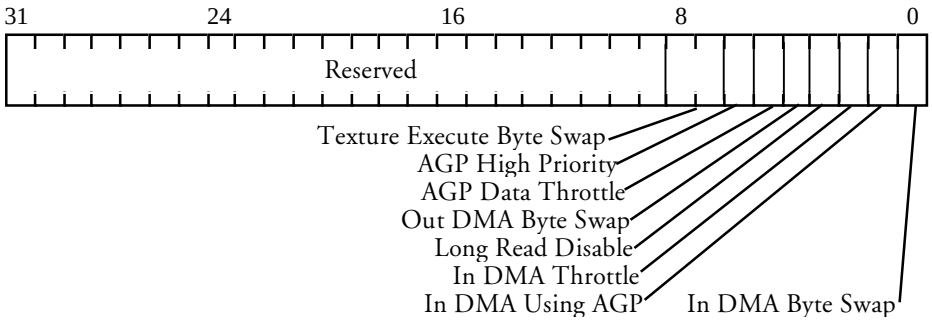
3.2.13 DMA Control

The DMA control Register sets up the data transfer modes for the DMA controller. The DMA controller can be set to little endian or big (byte swapped) endian.

DMAControl

Region: 0 Read/Write

Offset: 0000.0060h Reset Value: 0000.0000h



Bit 0 In DMA Byte Swap
- 0 Little endian
- 1 Big endian

Bit 1 In DMA Using AGP
- 0 Input DMA uses PCI master
- 1 Input DMA uses AGP master

Bit 2 In DMA Data Throttle

Applies to AGP transfers to GP input FIFO.
- 0 Control data flow using bus protocols
- 1 Throttle data requests based on input FIFO space

Bit 3 Long Read Disable
- 0 Long reads allowed
- 1 Long reads disabled

Bit 4 Out DMA Byte Swap Control
- 0 Little endian
- 1 Big endian

Bit 5 AGP Data Throttle

Applies to all AGP transfers.
- 0 Control data flow using bus protocols
- 1 Throttle data requests based on FIFO space

Bit 6 AGP High Priority
- 0 Use AGP low priority reads
- 1 Use AGP high priority reads

Bits 7-8 Texture Execute Byte Swap
- 0 Standard
- 1 Read buffer selected by bit 31 of memory contents
- 2 Half Word Swapped
- 3 Reserved

Bits 9-31 Reserved

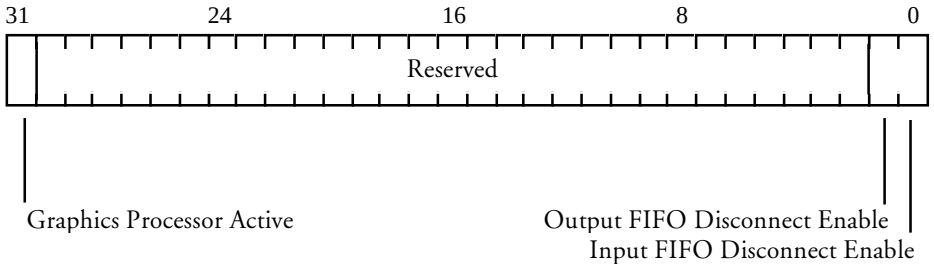
3.2.14 FIFO Disconnect

The FIFO disconnect register enables the input and output FIFO disconnect signals, which drive two physical pins on PERMEDIA. Disconnects are disabled at reset.

FIFODiscon

Region: Zero Read/Write

Offset: 0000.0068h Reset Value: 0000.0000h



Bit 0	Input FIFO Disconnect Enable 0 Disabled. 1 Enabled.	
Bit 1	Output FIFO Disconnect Enable 0 Disabled. 1 Enabled.	
Bits 2-30	Reserved Read as zero	(Read Only)
Bit 31	Graphics Processor Active 0 Idle. 1 Active.	

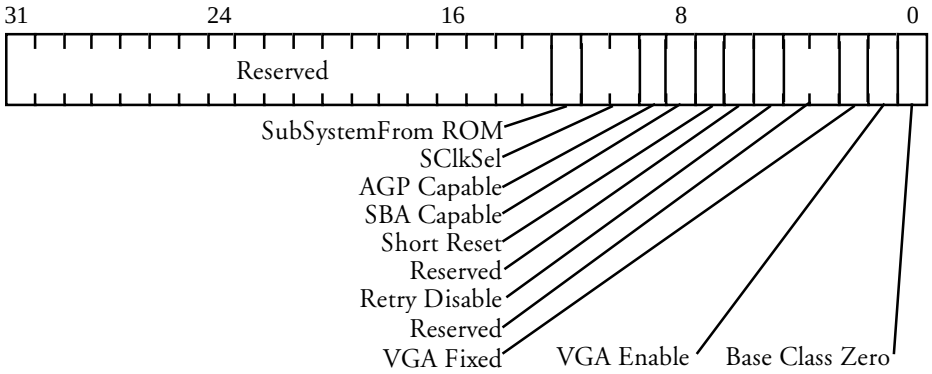
3.2.15 Chip Configuration

Most of the sampled values from the configuration pins are loaded into this register on the trailing edge of reset. This register can then be read back over the PCI bus to allow the host to determine how PERMEDIA has been configured and to modify fields of the configuration if required.

ChipConfig

Region: Zero Read/Write

Offset: 0000.0070h Reset Value: from configuration data



Bit 0 BaseClassZero
- 0 use the correct PCI Base Class Code
- 1 force PCI Base Class Code to be zero

Bit 1 VGAEnable
- 0 disabled internal SVGA subsystem
- 1 enabled internal SVGA subsystem

Bit 2 VGAFixed
- 0 disabled SVGA fixed address decoding
- 1 enabled SVGA fixed address decoding

Bits 3-4 Reserved

Bit 5 RetryDisable
- 0 enabled PCI Retry using “Disconnect-Without-Data”
- 1 disabled PCI Retry using “Disconnect-Without-Data”

Bit 6 Reserved

Bit 7 ShortReset
- 0 generate normal reset
- 1 generate short reset

Bit 8 SBA Capable
- 0 AGP sideband addressing disable
- 1 AGP sideband addressing enable

Bit 9 AGP Capable
- 0 Not AGP capable
- 1 AGP Capable

Bits 10-11 SClk Sel
- 0 PClk

1 PClk/2
2 MClk
3 MClk/2

Bit 12 Sub System From ROM
0 Leave subsystem registers at reset state
1 Load sub system registers from ROM immediately after reset

Bits 13-31 Reserved (Read Only)
Read as zero

3.2.16 Out DMA Start Address

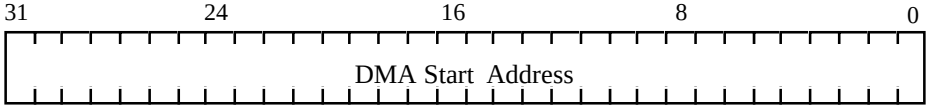
The DMA address should be loaded with the first PCI address for the buffer to be transferred into from the GCd when using the DMA controller.

Writing to the DMA count register loads the address into the DMA counter. Once a DMA has been set off the next DMA start address may be loaded. A read of this register returns the last start value loaded even if the DMA is underway.

OutDMAAddress

Region: Zero Read/Write

Offset: 0000.0080h Reset Value: 0000.0000h



Bits 0-31 Out DMA Start Address
PCI start address for PCI master write transfer from the Graphics Core.

3.2.17 Out DMA Count

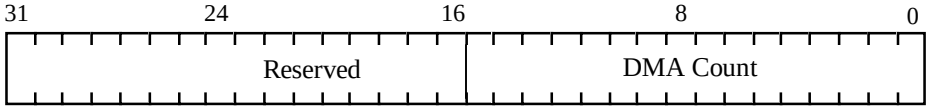
The DMA count register should be loaded with the number of words to be transferred in the DMA operation. The action of loading a word count greater than zero sets off the DMA operation. The value read back from this register indicates the current number of words left to be transferred.

This register should only be written to if the count is zero. It can be read at any time.

DMACount

Region: Zero Read/Write

Offset: 0000.0088h Reset Value: 0000.0000h



Bits 0-15 DMA Count
Valid Range: 0 to 65535
Number of words to be transferred in DMA operation.
Undefined action if this register is written to when it is not zero.

Bits 16-31 Reserved

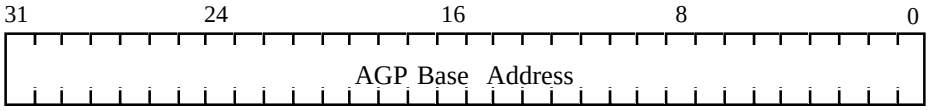
3.2.18 AGP Texture Base Address

Base address of the texture in system memory. When a texture map is accessed directly from system memory the address in this register is added to the address generated by the Texture Read Unit in the Graphics Core.

AGPTexBaseAddress

Region: Zero Read/Write

Offset: 0000.0090h Reset Value: 0000.0000h



Bits 0-31 AGP Texture Base Address
Base address in bytes for textures 'executed' from system memory.

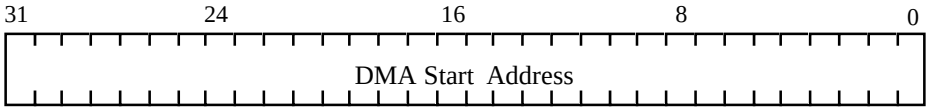
3.2.19 Bypass DMA Start Address

The DMA address should be loaded with the first PCI address for the data to be transferred into the bypass by the DMA controller.

ByDMAAddress

Region: Zero Read/Write

Offset: A0h Reset Value: 0000.0000h



Bits 0-31 DMA Start Address
PCI start address for PCI master read transfer in bytes (must be aligned to 32 bits).

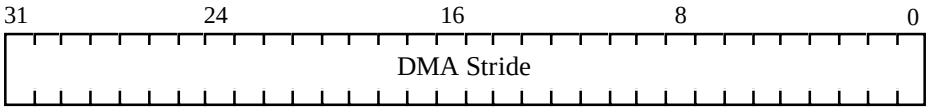
3.2.20 Bypass DMAStride

Sets the stride between scanlines of the data buffer to be transferred to by the bypass DMA controller.

ByDMAStride

Region: Zero Read/Write

Offset: B8h Reset Value: 0000.0000h



Bits 0-31 DMA stride
Stride between scanlines in system memory in bytes (must be aligned to 32 bits).

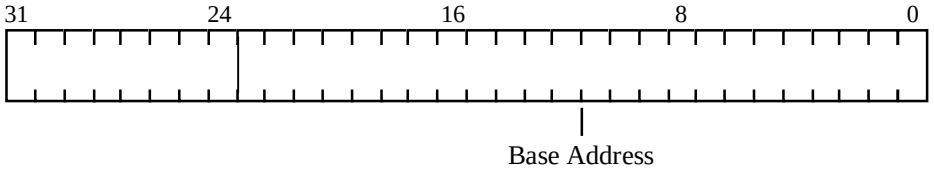
3.2.21 Bypass Memory Base Address

Sets the base address of the data buffer in PERMEDIA memory that data should be written to by the bypass DMA controller.

ByDMAMemAddr

Region: Zero Read/Write

Offset: C0h Reset Value: 0000.0000h



Bits 0-23	Base address In natural data units (i.e. texels).	
Bits 24-31	Reserved (all bits zero)	(Read Only)

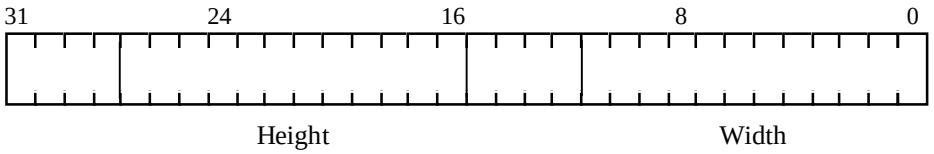
3.2.22 Bypass DMA Transfer Size

Sets the dimensions of the data buffer to be transferred by the DMA controller.

ByDMASize

Region: Zero Read/Write

Offset: C8h Reset Value: 0000.0000h



Bits 0-11	Width Number of data items to transfer in natural size (i.e. texels).	
Bits 12-15	Reserved (all bits zero)	(Read Only)
Bits 16-27	Height Number of scanlines to transfer.	
Bits 28-31	Reserved (all bits zero)	(Read Only)

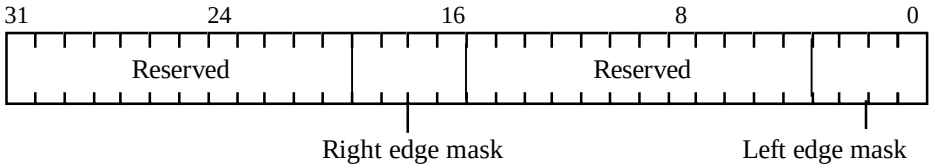
3.2.23 Bypass DMA Byte Mask

Masks the left and right edges of the data buffer transferred by the bypass DMA controller. The masks are applied to the 32 bit word at the start or end of the scanline as appropriate (the transfer is always done left to right).

ByDMAByteMask

Region: Zero Read/Write

Offset: D0h Reset Value: 0000.0000h



Bits 0-3	Left edge mask (all bits zero)	
Bits 4-15	Reserved	(Read Only)
Bits 16-19	Right edge mask	
Bits 20-31	Reserved (all bits zero)	(Read Only)

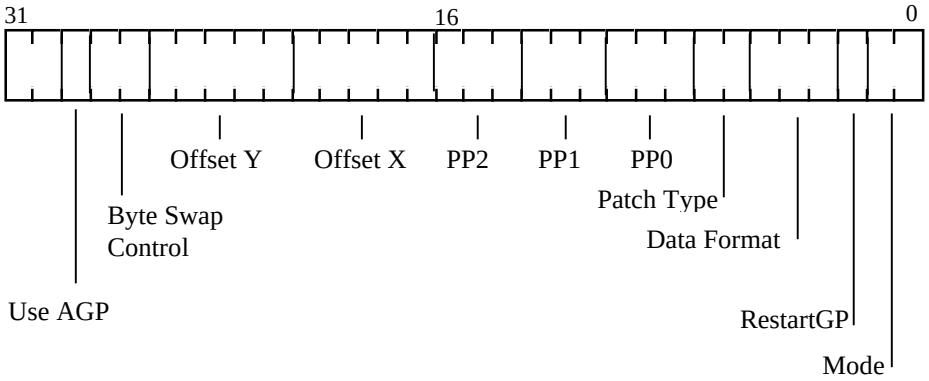
3.2.24 Bypass DMA Control

Controls the operation of the bypass DMA controller.

ByDMAControl

Region: Zero Read/Write

Offset: D8h Reset Value: 0000.0000h



Bits 0-1 Mode
- 0 Off
- 1 DMA (implicit auto addressing)
- 2 Aperture 1 auto addressing
- 3 Aperture 2 auto addressing

Bit 2 RestartGP
- 0 Do not restart graphics processor at end of DMA
- 1 Restart graphics processor at end of DMA

Bits 3-5 Data format
- 0 8 bits
- 1 16 bits
- 2 32 bits
- 3 4 bits
- 4 Y component - YUV planar mode
- 5 U component- YUV planar mode
- 6 V component- YUV planar mode
- 7 Reserved

Bits 6-7 Patch Type
- 0 Patching disabled
- 1 Patch mode (8x8)
- 2 Sub-patch mode (32x32)
- 3 Reserved

Bits 8-10 PP0
See table of partial product codes

Bits 11-13 PP1
See table of partial product codes

Bits 14-16 PP2
See table of partial product codes

Bits 17-21	Offset X Added to X value in address calculation	
Bits 22-26	Offset Y Added to Y value in address calculation	
Bits 27-28	Byte Swap Control 0 Standard (no swap) 1 Byte swapped 2 Half word swapped 3 Reserved	
Bits 29	Use AGP 0 = Use PCI Master 1 = Use AGP Master	
Bits 30-31	Reserved (all bits zero)	(Read Only)

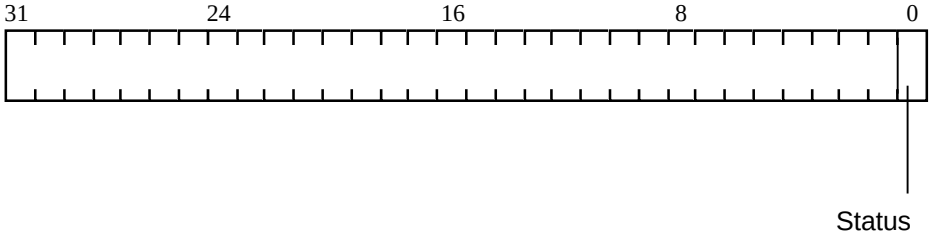
3.2.25 Bypass DMA Complete

Writing to this address indicates that the bypass load operations are complete and the graphics processor should be restarted. This is the manual version of RestartGP in ByDMAControl.

ByDMAComplete

Region: Zero Read/Write

Offset: E8h Reset Value: 0000.0000h



Bits 0	Status Reading this field returns the status of the texture invalid flag. 0 Texture Valid 1 Texture Invalid Flag set.	(Read only)
Bits 1-31	Read as zero, write data ignored	

3.3 Memory Control Registers

Refer to the memory system section for details on programming the memory control registers.

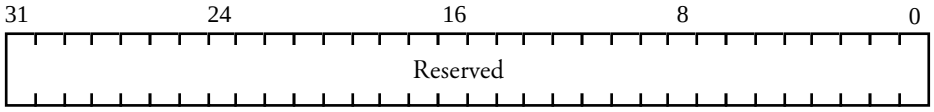
3.3.1 Re-Boot

Writing to this address instructs the memory controller to reboot the SGRAMs. This involves going through the reset sequence and loading the Boot Address register. A re-boot does not reload the configuration data; registers maintain their contents until a reset. A read from this register returns zero.

Reboot

Region: Region 0 Write

Offset: 1000h Reset Value: 0000.0000h



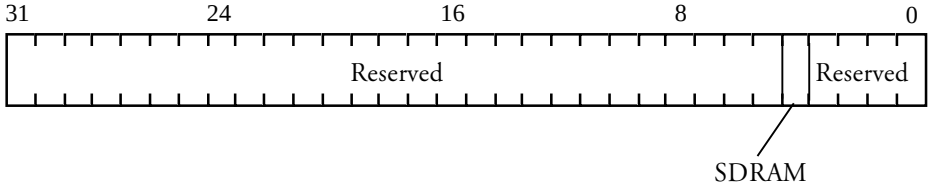
Bits 0-31 Reserved
Read as zero.

3.3.2 Memory Control

MemControl

Region: Region 0 Read/Write

Offset: 1040h Reset Value: 0000.0000h



Bits 0-3 Reserved
Read as zero.

Bit 4 SDRAM
0 SGRAM fitted
1 SDRAM fitted

Bits 5-31 Reserved
Read as zero.

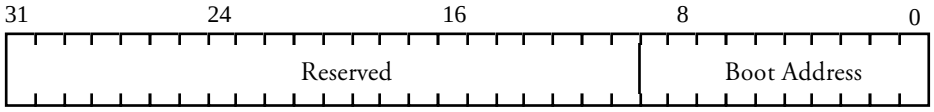
3.3.3 Boot Address

The boot address value specifies the contents of the SGRAM mode register at boot time. Boot time is either at chip reset or the reboot caused by writing to a register.

BootAddress

Region: Region 0 Read/Write

Offset: 1080h Reset Value: 0000.0031h



Bits 0-9 BootAddress
See memory data sheet for bit pattern.

Bits 10-31 Reserved
Read as zero.

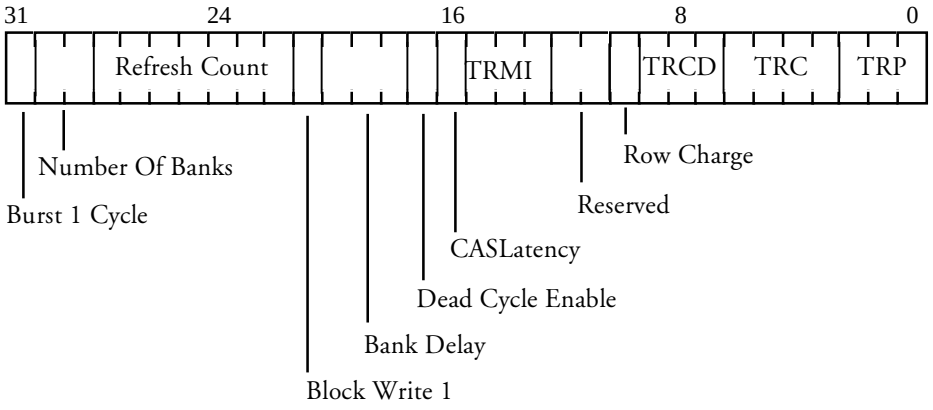
3.3.4 Memory Configuration

This register holds configuration data for the memory controller. If it is written to there is an automatic reboot of the memory. The correct sequence is to load the boot address, then change this register to match the boot address.

MemConfig

Region: Region 0 Read/Write

Offset: 10C0h Reset Value: 259F.FFFFh



Bits 0-2 TimeRP
Number of MClks from issuing PRECHARGE to issuing an ACTIVATE command. Should be set to tRP - 1.

Bits 3-6 TimeRC
Number of MClks from issuing AUTO-REFRESH to issuing another command. Should be set to tRC - 2.

Bits 7-9 TimeRCD
Number of MClks from issuing RAS to issuing CAS. Should be set to tRCD - 2.

Bit 10 RowCharge
- 0 row charge disabled
- 1 row charge enabled

Bits 11-12 Reserved

Bits 13-15 TimeRASMin
Number of active clocks for the minimum Row Active time. Should be set to tRAS - 3 unless this results in a value of zero.

Bit 16 CASLatency
- 0 CAS latency of 2.
- 1 CAS latency of 3.

Bit 17 DeadCycleEnable
- 0 do not insert dead cycle between reads and writes
- 1 insert dead cycle between reads and writes.

Bits 18-20 BankDelay
Defines read burst length. Should be set to burst length - 1.

Bit 21 Block Write 1
- 0 do not allow single cycle block write operations
- 1 allow single cycle block write operations

Bits 22-28 RefreshCount
Defines period between AUTO-REFRESH commands. The count is in MClks/32.

Bits 29-30 NumberBanks
- 0 1 bank (2Mbytes)
- 1 2 banks (4Mbytes)
- 2 3 banks (6Mbytes)
- 3 4 banks (8Mbytes)

Bit 31 Burst1Cycle
- 0 do not assume burst length of 1.
- 1 assume burst length of 1.

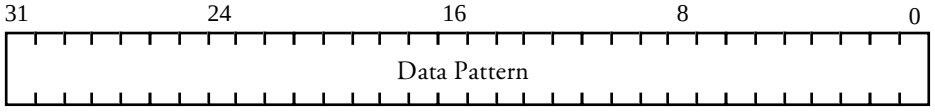
3.3.5 Bypass Write Mask

Mask used to protect bits from modification by bypass writes to memory.

BypassWriteMask

Region: Region 0 Read/Write

Offset: 1100h Reset Value: undefined



Bits 0-31 Data Pattern
Bit set to 0 = corresponding bit in memory protected.
Bit set to 1 = corresponding bit in memory writable

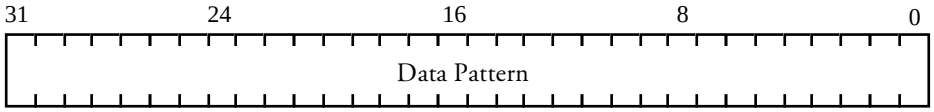
3.3.6 Framebuffer Write Mask

Mask used to protect bits from modification by framebuffer writes to memory. Can be read from the bypass, but can only be modified through framebuffer write unit in the graphics core.

FramebufferWriteMask

Region: Region 0 Read

Offset: 1140h Reset Value: undefined



Bits 0-31 Data Pattern
Bit set to 0 = corresponding bit in memory protected.
Bit set to 1 = corresponding bit in memory writable.

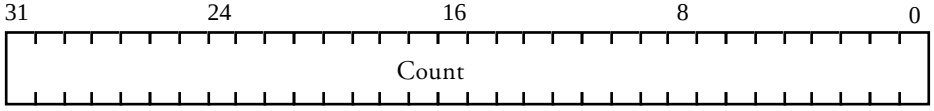
3.3.7 Count

A free running count that may be used for any purpose. The counter is driven by MClk and wraps to zero at overflow.

Count

Region: Region 0 Read

Offset: 1180h Reset Value: undefined



Bits 0-31 Count

3.4 Video Control Registers

Refer to the video unit section for details on programming the video control registers.

3.4.1 Screen Base

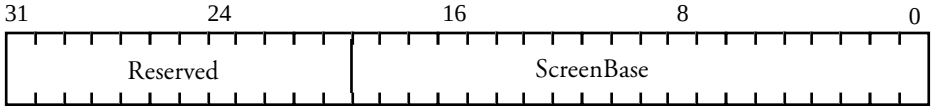
Address of pixel in top left of screen. The value of this register is ignored until vertical blank. When it is loaded the BypassPending bit is set in the VideoControl register until it is used.

This register must be loaded after the HgEnd register.

ScreenBase

Region: Region 0 Read/Write

Offset: 3000h Reset Value: undefined



Bits 0-19 ScreenBase
Base address of screen in 64 bit units.

Bits 20-31 Reserved
Read as zero

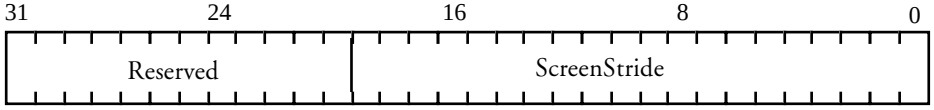
3.4.2 Screen Stride

Stride between scanlines of display.

ScreenStride

Region: Region 0 Read/Write

Offset: 3008h Reset Value: undefined



Bits 0-19 ScreenStride
Stride between scanlines in 64 bit units.

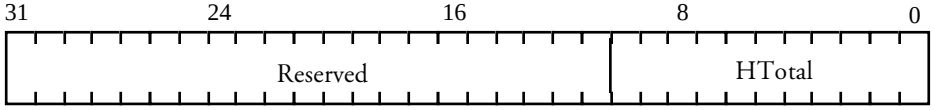
Bits 20-31 Reserved
Read as zero

3.4.3 Horizontal Total

HTotal

Region: Region 0 Read/Write

Offset: 3010h Reset Value: undefined



Bits 0-10 Htotal
Last 32 or 64 bit unit, including HBlank, on screen (i.e. horizontal total value - 1).

Bits 11-31 Reserved
Read as zero

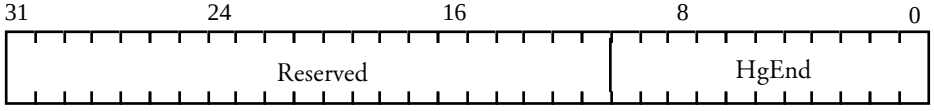
3.4.4 Horizontal Gate End

The gate period defines the period during which video data is not clocked from PERMEDIA. The value of this register is not used until vertical blank.

HgEnd

Region: Region 0 Read/Write

Offset: 3018h Reset Value: undefined



Bits 0-10 HgEnd
Last 32 or 64 bit unit in gate period (i.e. gate period - 1).

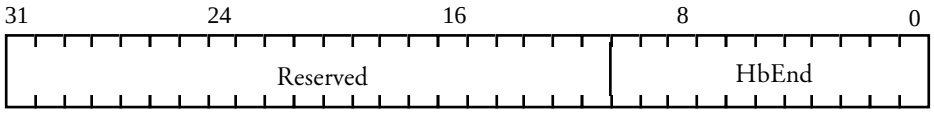
Bits 11-31 Reserved
Read as zero

3.4.5 Horizontal Blank End

HbEnd

Region: Region 0 Read/Write

Offset: 3020h Reset Value: undefined



Bits 0-10 HbEnd
Last 32 or 64 bit unit in horizontal blank period (i.e. blank period - 1).

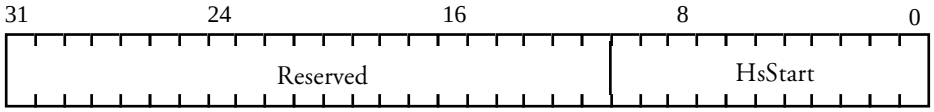
Bits 11-31 Reserved
Read as zero

3.4.6 Horizontal Sync Start

HsStart

Region: Region 0 Read/Write

Offset: 3028h Reset Value: undefined



Bits 0-10 HsStart
First 32 or 64 bit unit in horizontal sync period (i.e. front porch period).

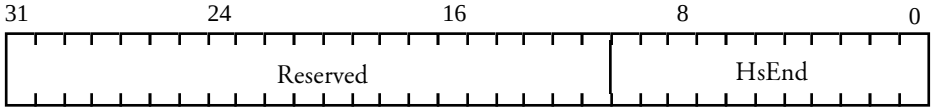
Bits 11-31 Reserved
Read as zero

3.4.7 Horizontal Sync End

HsEnd

Region: Region 0 Read/Write

Offset: 3030h Reset Value: undefined



Bits 0-10 HsEnd
First 32 or 64 bit unit out of horizontal sync period (i.e. front porch period + sync width).

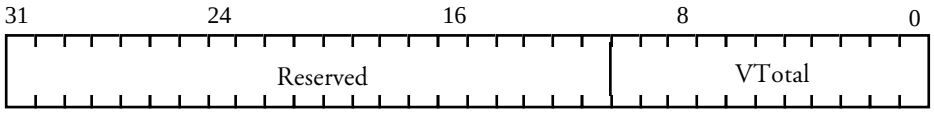
Bits 11-31 Reserved
Read as zero

3.4.8 Vertical Total

VTotal

Region: Region 0 Read/Write

Offset: 3038h Reset Value: undefined



Bits 0-10 Vtotal
Last scanline on screen, including vertical blank (i.e. number of lines - 1).

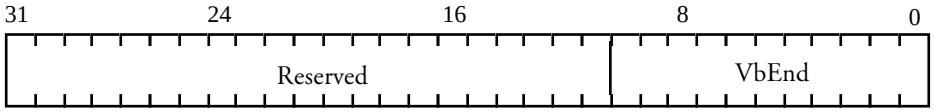
Bits 11-31 Reserved
Read as zero

3.4.9 Vertical Blank End

VbEnd

Region: Region 0 Read/Write

Offset: 3040h Reset Value: undefined



Bits 0-10 VbEnd
First scanline out of vertical blank (i.e. blank period).

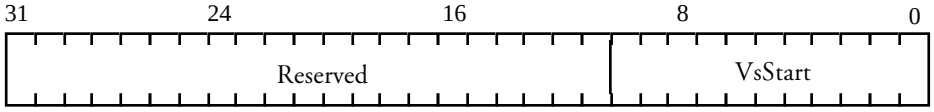
Bits 11-31 Reserved
Read as zero

3.4.10 Vertical Sync Start

VsStart

Region: Region 0 Read/Write

Offset: 3048h Reset Value: undefined



Bits 0-10 VsStart
Scanline before start of vertical sync (i.e. period of front porch - 1).

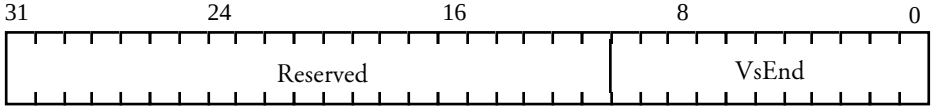
Bits 11-31 Reserved
Read as zero

3.4.11 Vertical Sync End

VsEnd

Region: Region 0 Read/Write

Offset: 3050h Reset Value: undefined



Bits 0-10 VsEnd
Last scanline in vertical sync (i.e. front porch + sync width - 1).

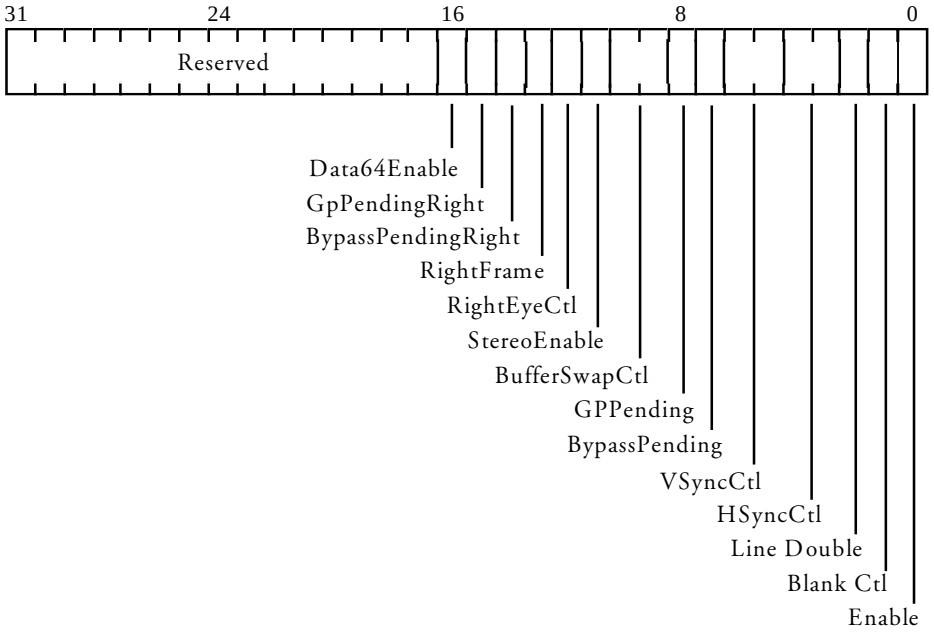
Bits 11-31 Reserved
Read as zero

3.4.12 Video Control

VideoControl

Region: Region 0 Read/Write

Offset: 3058h Reset Value: 0000.0000h



Bit 0 Enable
0 GP video disabled
1 GP video enabled

Bit 1 BlankCtl
0 Active High
1 Active Low

Bit 2 LineDouble
0 Line doubling disabled
1 Line doubling enabled
If enabled, each scanline is displayed twice to increase the effective frequency of low resolution screens.

Bits 3-4 HSyncCtl
- 0 Forced High
- 1 Active High
- 2 Forced Low
- 3 Active Low

Bits 5-6 VSyncCtl
- 0 Forced High
- 1 Active High
- 2 Forced Low
- 3 Active Low

Bit 7 BypassPending
- 0 ScreenBase value used.
- 1 New ScreenBase value waiting to be used.

Read only bit, set when ScreenBase is loaded through the bypass.

Bit 8 GPPending
- 0 ScreenBase value used.
- 1 New ScreenBase value waiting to be used.

Read only bit, set when ScreenBase is loaded through the Graphics Processor.

Bits 9-10 BufferSwapCtl
- 0 SyncOnFrameBlank
- 1 FreeRunning
- 2 LimitToFrameRate
- 3 Reserved

Bit 11 StereoEnable
- 0 Disabled
- 1 Enabled

Bit 12 RightEyeCtl
- 0 Active High
- 1 Active Low

Bit 13 RightFrame
- 0 Displaying left frame.
- 1 Displaying right frame.

Read only bit.

Bit 14 BypassPendingRight
- 0 ScreenBaseRight value used.
- 1 New ScreenBaseRight value waiting to be used.

Read only bit, set when ScreenBaseRight is loaded through the bypass.

Bit 15 GPPendingRight
- 0 ScreenBaseRight value used.
- 1 New ScreenBaseRight value waiting to be used.

Read only bit, set when ScreenBaseRight is loaded through the Graphics Processor.

Bit 16 Data64Enable
- 0 Data output to RAMDAC as 32 bit units.
- 1 Data output to RAMDAC as 64 bit units.

Bits 17-31 Reserved

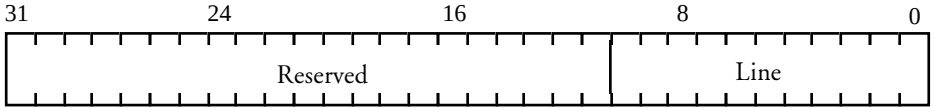
Read as zero

3.4.13 Interrupt Line

InterruptLine

Region: Region 0 Read/Write

Offset: 3060h Reset Value: undefined



Bits 0-10 Line
Generate interrupt at start of this line.

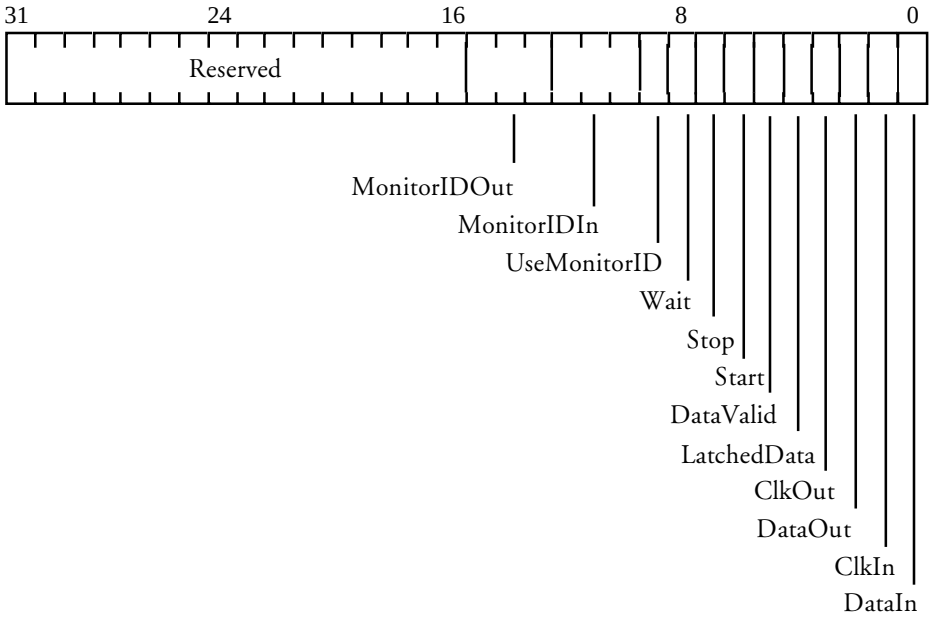
Bits 11-31 Reserved
Read as zero

3.4.14 Display Data

DisplayData

Region: Region 0 Read/Write

Offset: 3068h Reset Value: 0000.0000h



Bit 0 DataIn
0 Data is Low.
1 Data is High.

Bit 1 ClkIn
0 Clock is Low.
1 Clock is High.

Bit 2 DataOut
0 Drive Low.
1 Drive Tri-state.

Bit 3 ClkOut
- 0 Drive Low.
- 1 Drive Tri-state.

Bit 4 LatchedData
- 0 Data latched at 0.
- 1 Data latched at 1.

Bit 5 DataValid
- 0 DataIn not valid.
- 1 DataIn valid.

Cleared by writing 1 to this bit.

Bit 6 Start
- 0 DDC bus has not passed through Start state.
- 1 DDC bus has passed through Start state.

Cleared by writing 1 to this bit.

Bit 7 Stop
- 0 DDC bus has not passed through Stop state.
- 1 DDC bus has passed through Stop state.

Cleared by writing 1 to this bit.

Bit 8 Wait
- 0 Do not insert wait states in DDC.
- 1 Insert wait states.

Bit 9 UseMonitorID
- 0 Use DDC.
- 1 Use monitor ID.

Bits 10-12 MonitorIDIn
- 0 Signal is Low.
- 1 Signal is High.

Bits 13-15 MonitorIDOut
- 0 Drive Low.
- 1 Drive Tri-state.

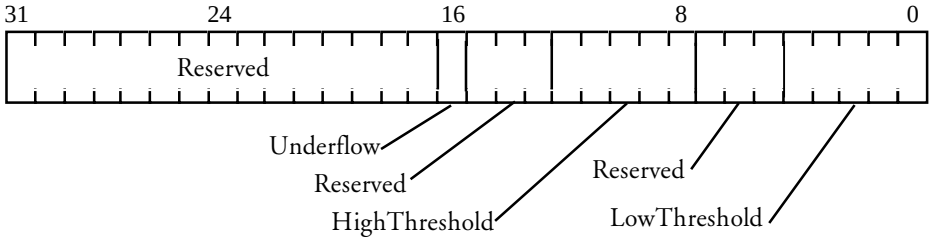
Bits 16-31 Reserved
Read as zero

3.4.15 Fifo Control

FifoControl

Region: Region 0 Read/Write

Offset: 3078h Reset Value: 0000.1010h



Bits 0-4 LowThreshold
Video data is accessed from memory at a low priority when there are this many or less spaces in the video FIFO.

Bits 5-7 Reserved
Read as zero.

Bits 8-12 HighThreshold
Video data is accessed from memory at a high priority when there are this many or less spaces in the video FIFO.

Bits 13-15 Reserved
Read as zero.

Bits 16 Underflow
0 Underflow has not occurred.
1 Underflow has occurred.
Cleared by writing 1 to this bit.

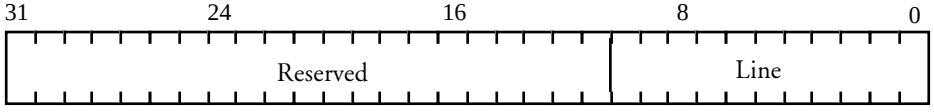
Bits 17-31 Reserved
Read as zero

3.4.16 Line Count

LineCount

Region: Region 0 Read/Write

Offset: 3070h Reset Value: undefined



Bits 0-10 Line
Current line

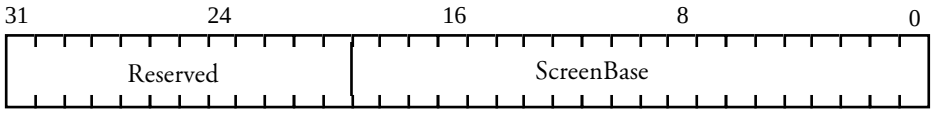
Bits 11-31 Reserved
Read as zero

3.4.17 Screen Base Right

ScreenBaseRight

Region: Region 0 Read/Write

Offset: 3080h Reset Value: undefined



Bits 0-19 ScreenBase
Base address of screen in 64 bit units.

Bits 20-31 Reserved
Read as zero

3.5 SVGA Interface

In addition to the standard SVGA registers, PERMEDIA supports two extended registers, the SVGA Control Register and the Mode 640 register.

3.5.1 SVGA Memory

The SVGA memory is accessed through the SVGA legacy memory addresses. The SVGA memory can also be accessed through the PERMEDIA memory apertures by setting the 'SVGA access' bit in either the ApertureOne or ApertureTwo register. The memory address for the SVGA is formed from bits 16 down to 2 of the incoming bus address. This results in the 128K Byte SVGA memory space being aliased within the 8Mbyte total region size. No byte swapping or other data formatting is performed when accessing the SVGA memory in this manner.

3.5.2 SVGA Registers

The PERMEDIA standard SVGA registers are accessed through the SVGA legacy IO addresses. These registers are also mapped into a 4K Byte space at offset 0x6000h in Region 0. The address for the SVGA unit is formed from bits 9 down to 2 of the incoming bus address. So, for example, SVGA register 0x3C4h can be addressed at offsets 0x63C4h, 0x67C4h, 0x6BC4h and 0x6FC4h.

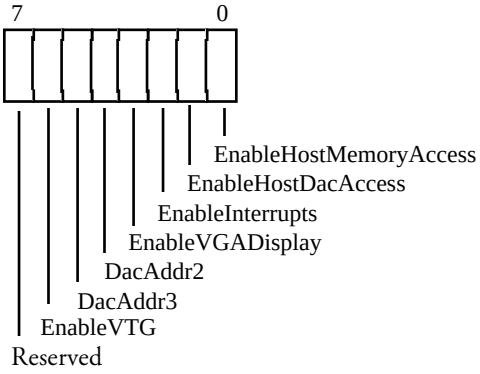
3.5.3 SVGA Control Register

This extended SVGA register is accessed as index 5 through the sequencer index register at port 0x3C4h. Data is written to port 3C5h.

VGAControlReg

Region: Region 0 Read/Write

Offset: Reset Value: 4Bh



Bit 0 EnableHostMemoryAccess
0 Disable host accesses to memory.
1 Enable host accesses to memory.

Bit 1 EnableHostDacAccess
0 Disable host accesses to RAMDAC.
1 Enable host accesses to RAMDAC.

Bit 2 EnableInterrupts
0 Disable interrupts from SVGA.
1 Enable interrupts from SVGA.

Bit 3 EnableVGADisplay
0 Disable SVGA display, enable graphics processor display.
1 Enable SVGA display, disable graphics processor display.

Bit 4 DacAddr2
Sets bit 2 of RAMDAC address.

Bit 5 DacAddr3
Sets bit 3 of RAMDAC address.

Bit 6 EnableVTG
0 Stops VTG running and producing sync pulses.
1 Enables VTG to produce sync pulses. Only has effect when EnableVGADisplay has been set to zero

Bit 7 Reserved
Read as zero

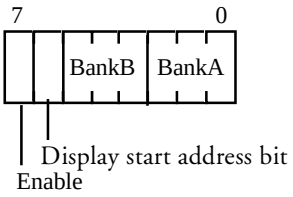
3.5.4 Mode 640 Register

This extended SVGA register is accessed as index 9 through the graphics index register at port 3CEh. Data is written to port 3CFh.

Mode640Reg

Region: Region 0 Read/Write

Offset: Reset Value: 00h



Bits 0-2 BankA
Additional address bits for accesses between A0000h and B0000h.

Bits 3-5 BankB
Additional address bits for accesses between B0000h and C0000h.

Bit 6 Bit 16 of display start address
Read as zero.

Bit 7 Enable
0 Mode640 disabled.
1 Mode640 enabled.

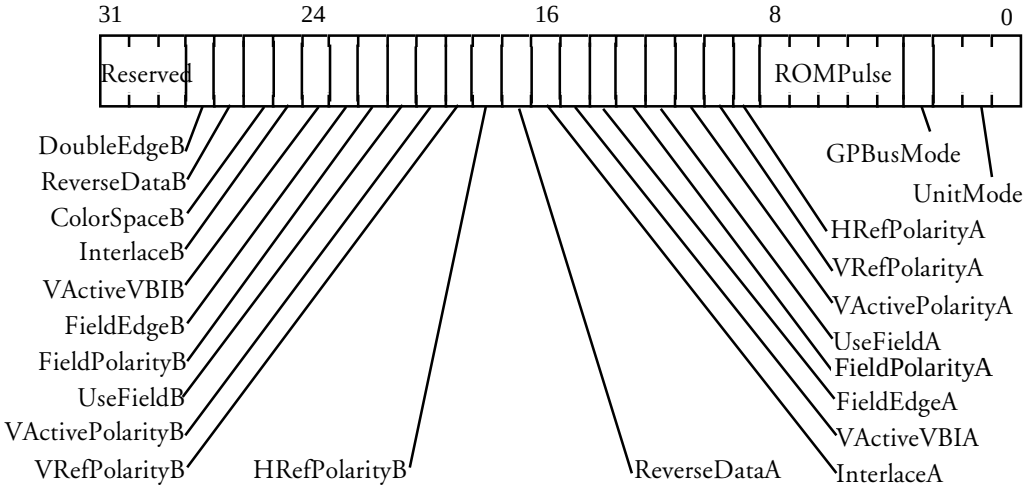
3.5.5 Video Streams Registers

Refer to the video streams section for details on programming these registers.

VSConfiguration

Region: Region 0 Read/Write

Offset: 5800h Reset Value: 0000.01F0h



Bits 0-2 Unit mode

Bit 3 GPBusMode
0 = Operate GP bus in Mode B
1 = Operate GP bus in Mode A

Bits 4-8 ROMPulse

Bit 9 HRefPolarityA
0 = Active low
1 = Active high

Bit 10 VRefPolarityA
0 = Active low
1 = Active high

Bit 11 VActivePolarityA
0 = Active low
1 = Active high

Bit 12 UseFieldA
0 = Disabled
1 = Enabled

Bit 13 FieldPolarityA
0 = Active low
1 = Active high

Bit 14 FieldEdgeA
0 = Inactive edge
1 = Active edge

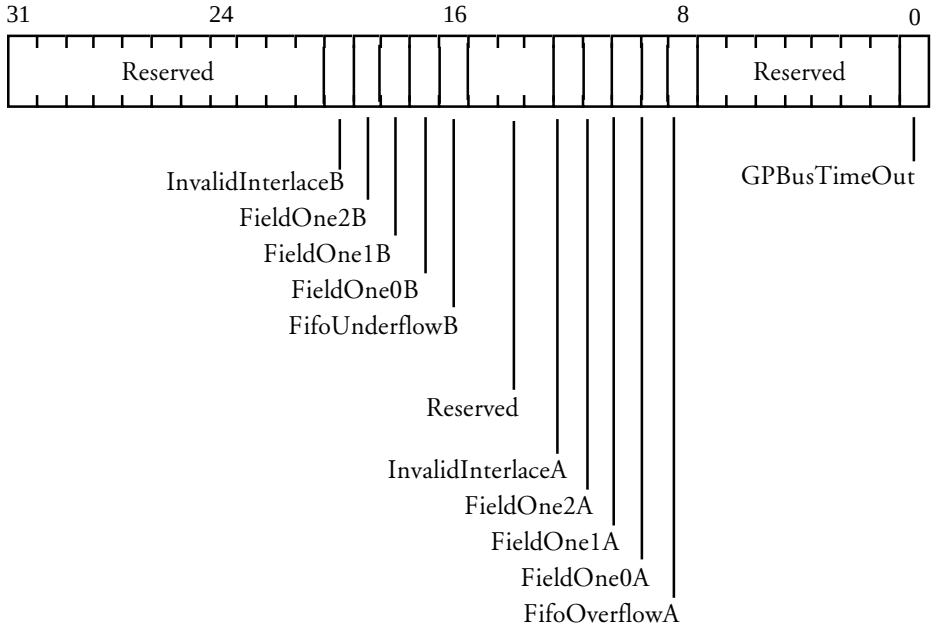
Bit 15	VActiveVBIA 0 = Ignore VActive for VBI data 1 = Gate VBI data with VActive
Bit 16	InterlaceA 0 = Video is not interlaced 1 = Video is interlaced
Bit 17	ReverseDataA 0 = Disabled 1 = Enabled
Bit 18	HRefPolarityB 0 = Active low 1 = Active high
Bit 19	VRefPolarityB 0 = Active low 1 = Active high
Bit 20	VActivePolarityB 0 = Active low 1 = Active high
Bit 21	UseFieldB 0 = Disabled 1 = Enabled
Bit 22	FieldPolarityB 0 = Active low 1 = Active high
Bit 23	FieldEdgeB 0 = Inactive edge 1 = Active edge
Bit 24	VActiveVBIB 0 = Ignore VActive for VBI data 1 = Gate VBI data with VActive
Bit 25	InterlaceB 0 = Video is not interlaced 1 = Video is interlaced
Bit 26	ColorSpaceB 0 = YUV 1 = RGB
Bit 27	ReverseDataB 0 = Disabled 1 = Enabled
Bit 28	DoubleEdgeB 0 = Disabled 1 = Enabled
Bits 29-31	Reserved

3.5.6 VSStatus

VSStatus

Region: Region 0 Read/Write

Offset: 5808h Reset Value: 0000.0000h



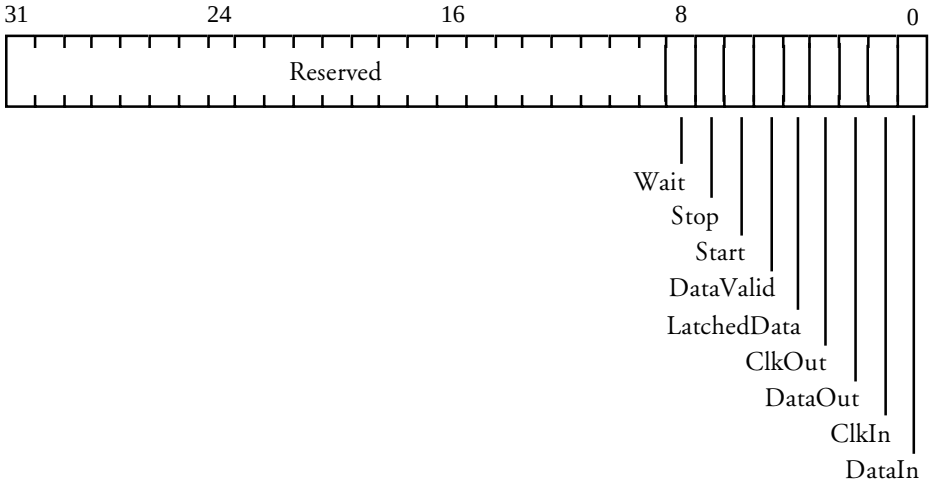
Bit 0	GPBusTimeOut	read/write, cleared by writing 1
Bits 1-7	Reserved	read as zero
Bit 8	FifoOverflowA	read/write, cleared by writing 1
Bit 9	FieldOne0A	read only
Bit 10	FieldOne1A	read only
Bit 11	FieldOne2A	read only
Bit 12	InvalidInterlaceA	read only
Bits 13-15	Reserved	read as zero
Bit 16	FifoUnderflowB	read/write, cleared by writing 1
Bit 17	FieldOne0B	read only
Bit 18	FieldOne1B	read only
Bit 19	FieldOne2B	read only
Bit 20	InvalidInterlaceB	read only
Bits 21-31	Reserved	read as zero

3.5.7 VSSerialBusControl

VSSerialBusControl

Region: Region 0 Read/Write

Offset: 5810h Reset Value: 0000.000Ch



Bit 0 DataIn
read only
0 = Data line is low
1 = Data line is high

Bit 1 ClkIn
read only
0 = Clock line is low
1 = Clock line is high

Bit 2 DataOut
0 = Drive data line low
1 = Tri-state data line

Bit 3 ClkOut
0 = Drive clock line low
1 = Tri-state clock line

Bit 4 LatchedData
0 = Data latched at 0
1 = Data latched at 1

Bit 5 DataValid
0 = DataIn not valid
1 = DataIn valid
Cleared by writing one to this bit.

Bit 6 Start
0 = Has not passed through start state
1 = Has passed through start state
Cleared by writing one to this bit.

Bit 7 Stop
0 = Has not passed through stop state

1 = Has passed through stop state
Cleared by writing one to this bit.

Bit 8 Wait
0 = Do not insert wait states
1 = Insert wait states

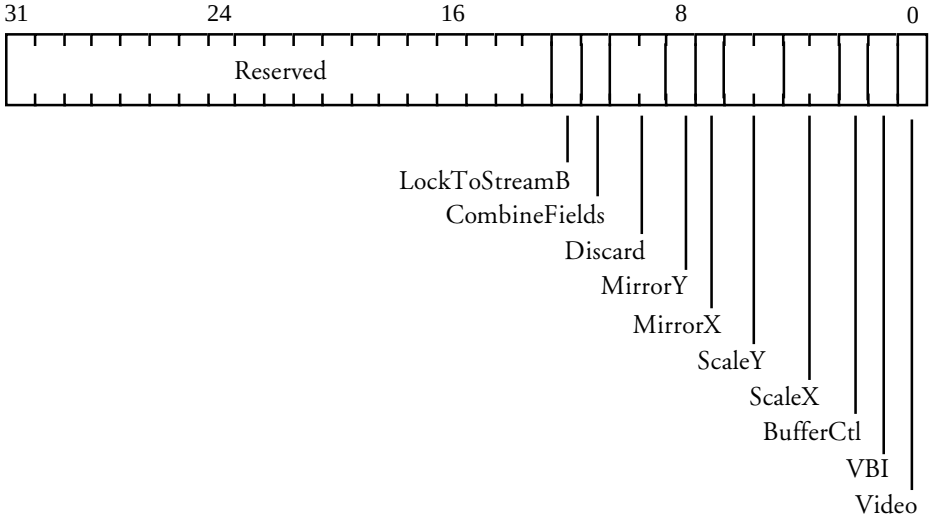
Bits 9-31 Reserved
Read back as zero.

3.5.8 VSAControl

VSAControl

Region: Region 0 Read/Write

Offset: 5900h Reset Value: 0000.0000h



Bit 0 Video
0 = Disable
1 = Enable

Bit 1 VBI
0 = Disable
1 = Enable

Bit 2 BufferCtl
0 = Double buffered
1 = Triple buffered

Bits 3-4 ScaleX
0 = 1:1
1 = 2:1
2 = 4:1
3 = 8:1

Bits 5-6 ScaleY
0 = 1:1
1 = 2:1
2 = 4:1
3 = 8:1

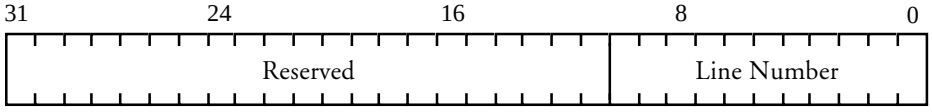
Bit 7	MirrorX 0 = Disable 1 = Enable
Bit 8	MirrorY 0 = Disable 1 = Enable
Bits 9-10	Discard 0 = None 1 = FieldOne 2 = FieldTwo 3 = Reserved
Bit 11	CombineFields 0 = Disable 1 = Enable
Bit 12	LockToStreamB 0 = Disable 1 = Enable
Bits 13-31	Reserved

3.5.9 VSAInterrupt

VSAInterrupt

Region: Region 0 Read/Write

Offset: 5908h Reset Value: undefined



Bits 0-10 Line number to generate interrupt.

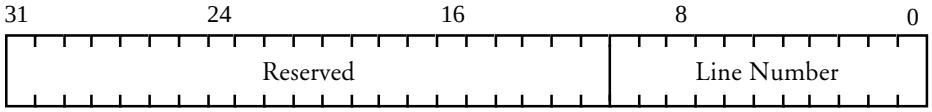
Bits 11-31 Reserved

3.5.10 VSACurrentLine

VSACurrentLine

Region: Region 0 Read/Write

Offset: 5910h Reset Value: undefined



Bits 0-10 Current line number.

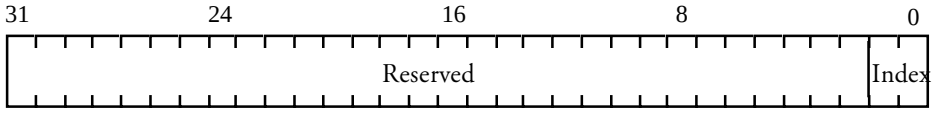
Bits 11-31 Reserved

3.5.11 VSAVideoAddressHost

VSAVideoAddressHost

Region: Region 0 Read/Write

Offset: 5918h Reset Value: undefined



Bits 0-1 Host base address register index.

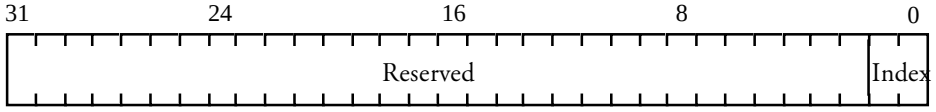
Bits 2-31 Reserved

3.5.12 VSAVideoAddressIndex

VSAVideoAddressIndex

Region: Region 0 Read/Write

Offset: 5920h Reset Value: 0000.0000h



Bits 0-1 Base address register index.

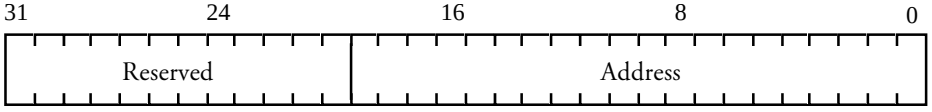
Bits 2-31 Reserved

3.5.13 VSAVideoAddress0

VSAVideoAddress0

Region: Region 0 Read/Write

Offset: 5928h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

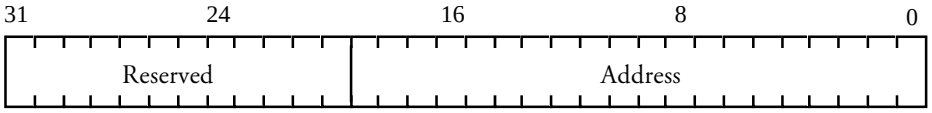
Bits 20-31 Reserved

3.5.14 VSAVideoAddress1

VSAVideoAddress1

Region: Region 0 Read/Write

Offset: 5930h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

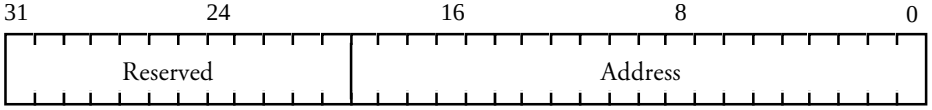
Bits 20-31 Reserved

3.5.15 VSAVideoAddress2

VSAVideoAddress2

Region: Region 0 Read/Write

Offset: 5938h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

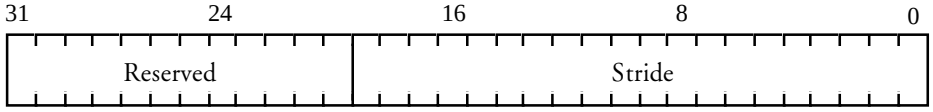
Bits 20-31 Reserved

3.5.16 VSAVideoStride

VSAVideoStride

Region: Region 0 Read/Write

Offset: 5940h Reset Value: undefined



Bits 0-19 Stride between scanlines (64 bit aligned)

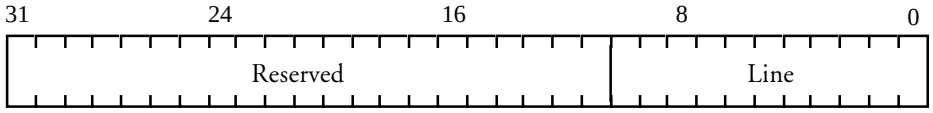
Bits 20-31 Reserved

3.5.17 VSAVideoStartLine

VSAVideoStartLine

Region: Region 0 Read/Write

Offset: 5948h Reset Value: undefined



Bits 0-10 First scanline of video data

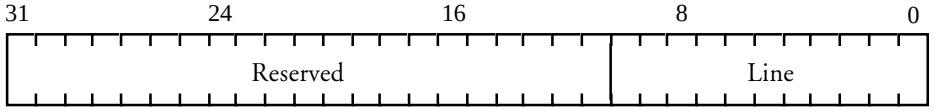
Bits 11-31 Reserved

3.5.18 VSAVideoEndLine

VSAVideoEndLine

Region: Region 0 Read/Write

Offset: 5950h Reset Value: undefined



Bits 0-10 First scanline after video data

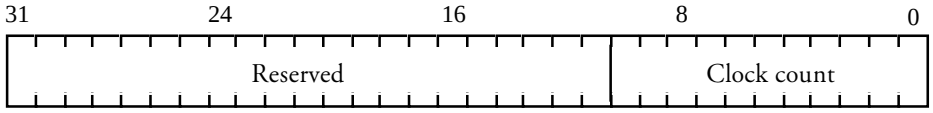
Bits 11-31 Reserved

3.5.19 VSAVideoStartData

VSAVideoStartData

Region: Region 0 Read/Write

Offset: 5958h Reset Value: undefined



Bits 0-10 First clock of valid data.

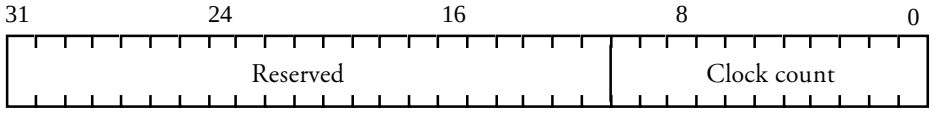
Bits 11-31 Reserved

3.5.20 VSAVideoEndData

VSAVideoEndData

Region: Region 0 Read/Write

Offset: 5960h Reset Value: undefined



Bits 0-10 First clock after active video.

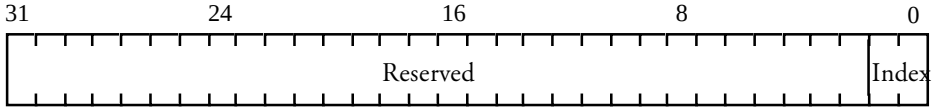
Bits 11-31 Reserved

3.5.21 VSAVBIAddressHost

VSAVBIAddressHost

Region: Region 0 Read/Write

Offset: 5968h Reset Value: undefined



Bits 0-1 Host base address register index.

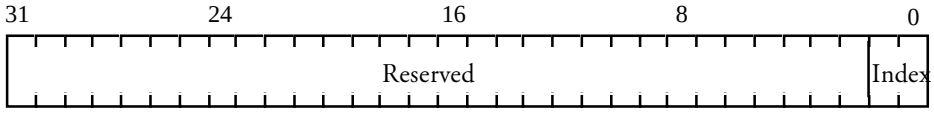
Bits 2-31 Reserved

3.5.22 VSAVBIAddressIndex

VSAVBIAddressIndex

Region: Region 0 Read/Write

Offset: 5970h Reset Value: 0000.0000h



Bits 0-1 Base address register index.

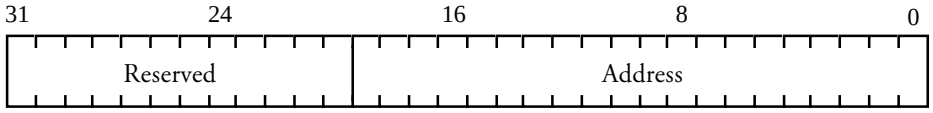
Bits 2-31 Reserved

3.5.23 VSAVBIAddress0

VSAVBIAddress0

Region: Region 0 Read/Write

Offset: 5978h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

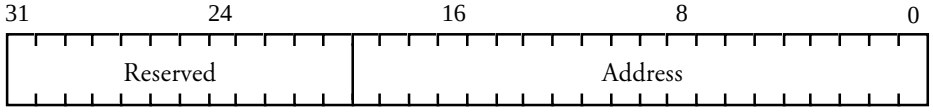
Bits 20-31 Reserved

3.5.24 VSAVideoAddress1

VSAVBIAddress1

Region: Region 0 Read/Write

Offset: 5980h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

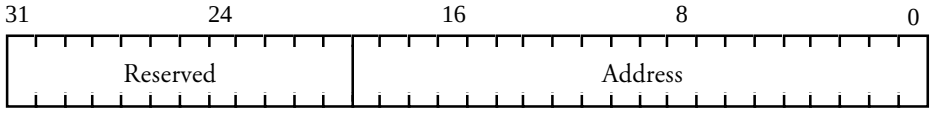
Bits 20-31 Reserved

3.5.25 VSAVBIAddress2

VSAVBIAddress2

Region: Region 0 Read/Write

Offset: 5988h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

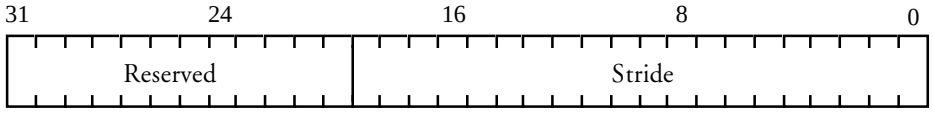
Bits 20-31 Reserved

3.5.26 VSAVBIStride

VSAVBIStride

Region: Region 0 Read/Write

Offset: 5990h Reset Value: undefined



Bits 0-19 Stride between scanlines (64 bit aligned)

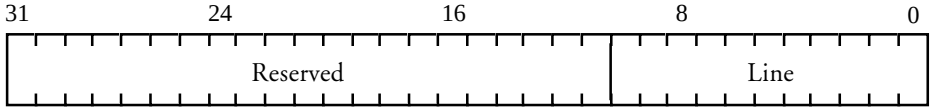
Bits 20-31 Reserved

3.5.27 VSAVBIStartLine

VSAVBIStartLine

Region: Region 0 Read/Write

Offset: 5998h Reset Value: undefined



Bits 0-10 First scanline of VBI data

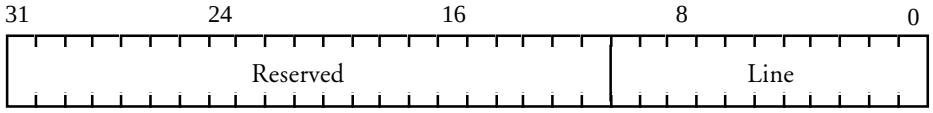
Bits 11-31 Reserved

3.5.28 VSAVBIEndLine

VSAVBIEndLine

Region: Region 0 Read/Write

Offset: 59A0h Reset Value: undefined



Bits 0-10 First scanline after VBI data

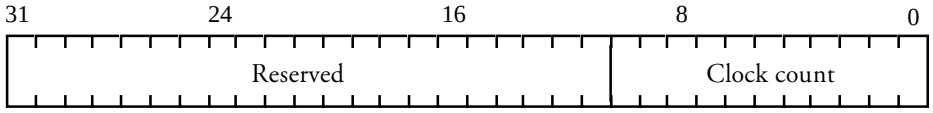
Bits 11-31 Reserved

3.5.29 VSAVBIStartData

VSAVBIStartData

Region: Region 0 Read/Write

Offset: 59A8h Reset Value: undefined



Bits 0-10 First clock of valid data in VBI line.

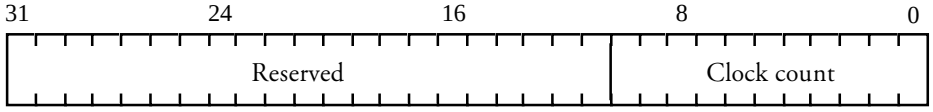
Bits 11-31 Reserved

3.5.30 VSAVBIEndData

VSAVBIEndData

Region: Region 0 Read/Write

Offset: 59B0h Reset Value: undefined



Bits 0-10 First clock after active VBI data.

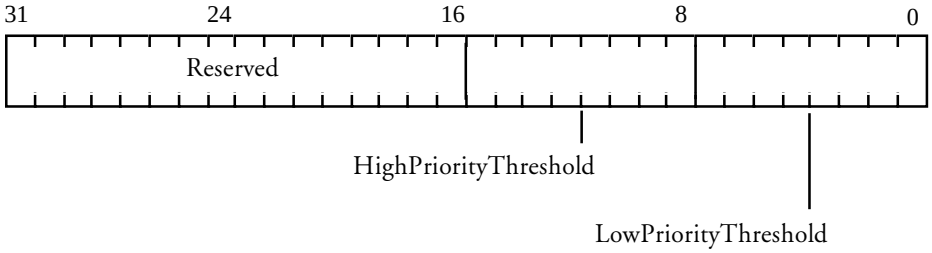
Bits 11-31 Reserved

3.5.31 VSAFifoControl

VSAFifoControl

Region: Region 0 Read/Write

Offset: 59B8h Reset Value: undefined



Bits 0-7 Low priority threshold

Bits 8-15 High priority threshold

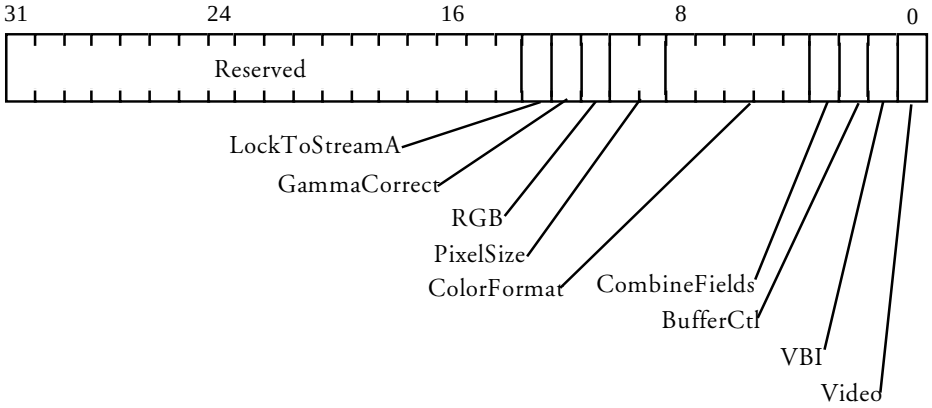
Bits 16-31 Reserved

3.5.32 VSBControl

VSBControl

Region: Region 0 Read/Write

Offset: 5A00h Reset Value: 0000.0000h



Bit 0 Video
0 = Disable
1 = Enable

Bit 1 VBI
0 = Disable
1 = Enable

Bit 2 BufferCtl
0 = Double buffered
1 = Triple buffered

Bit 3 CombineFields
0 = Disable
1 = Enable

Bits 4-8 ColorFormat
See table below.

			Internal Color Channels			
Color Format	**RGB**	**Name**	**R/Y**	**G/U**	**B/V**	**A**
0	0	8:8:8:8	8@0	8@8	8@16	8@24
1	0	5:5:5:1Front	5@0	5@5	5@10	1@15
2	0	4:4:4:4	4@0	4@4	4@8	4@12
5	0	3:3:2Front	3@0	3@3	2@6	0
6	0	3:3:2Back	3@8	3@11	2@14	0
9	0	2:3:2:1Front	2@0	3@2	2@5	1@7
10	0	2:3:2:1Back	2@8	3@10	2@13	1@15
11	0	2:3:2FrontOff	2@0	3@2	2@5	0
12	0	2:3:2BackOff	2@8	3@10	2@13	0
13	10	5:5:5:1Back	5@16	5@21	5@26	1@31
16	0	5:6:5Front	5@0	6@5	5@11	0
17	0	5:6:5Back	5@16	6@21	5@27	0
19	0	YUV422	8@0	8@8	8@8	0
0	1	8:8:8:8	8@16	8@8	8@0	8@24
1	1	5:5:5:1Front	5@10	5@5	5@0	1@15

2	1	4:4:4:4	4@8	4@4	4@0	4@12
5	1	3:3:2Front	3@5	3@2	2@0	0
6	1	3:3:2Back	3@13	3@10	2@8	0
9	1	2:3:2:1Front	2@5	3@2	2@0	1@7
10	1	2:3:2:1Back	2@13	3@10	2@8	1@15
11	1	2:3:2FrontOff	2@5	3@2	2@0	0
12	1	2:3:2BackOff	2@13	3@10	2@8	0
13	1	5:5:5:1Back	5@26	5@21	5@16	1@31
16	1	5:6:5Front	5@11	6@5	5@0	0
17	1	5:6:5Back	5@27	6@21	5@16	0
19	1	YUV422	8@8	8@0	8@0	0

Bits 9-10 PixelSize
0 = 8 bits
1 = 16 bits
2 = 32 bits

Bit 11 RGB
0 = BGR color order
1 = RGB color order

Bit 12 GammaCorrect
0 = Disable Gamma correction
1 = Enable Gamma correction

Bit 13 LockToStreamA
0 = Disable
1 = Enable

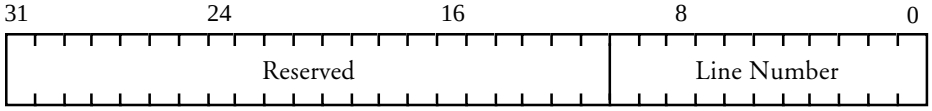
Bits 14-31 Reserved

3.5.33 VSBInterrupt

VSBInterrupt

Region: Region 0 Read/Write

Offset: 5A08h Reset Value: undefined



Bits 0-10 Line number to generate interrupt.

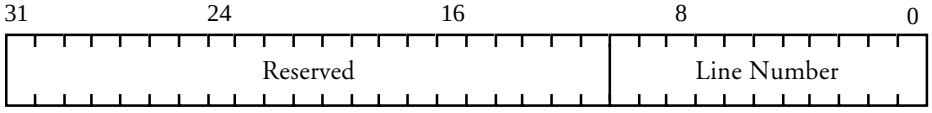
Bits 11-31 Reserved

3.5.34 VSBCurrentLine

VSBCurrentLine

Region: Region 0 Read/Write

Offset: 5A10h Reset Value: undefined



Bits 0-10 Current line number.

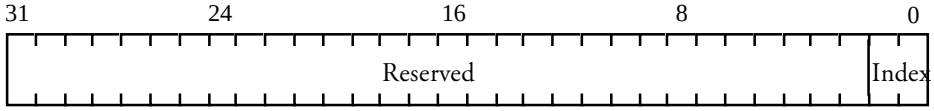
Bits 11-31 Reserved

3.5.35 VSBVideoAddressHost

VSBVideoAddressHost

Region: Region 0 Read/Write

Offset: 5A18h Reset Value: undefined



Bits 0-1 Host base address register index.

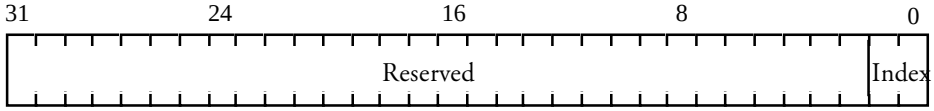
Bits 2-31 Reserved

3.5.36 VSBVideoAddressIndex

VSBVideoAddressIndex

Region: Region 0 Read/Write

Offset: 5A20h Reset Value: 0000.0000h



Bits 0-1 Base address register index.

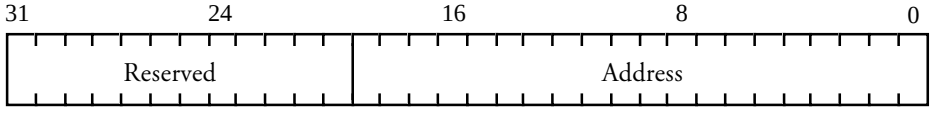
Bits 2-31 Reserved

3.5.37 VSBVideoAddress0

VSBVideoAddress0

Region: Region 0 Read/Write

Offset: 5A28h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

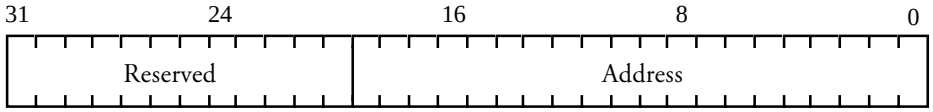
Bits 20-31 Reserved

3.5.38 VSBVideoAddress1

VSBVideoAddress1

Region: Region 0 Read/Write

Offset: 5A30h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

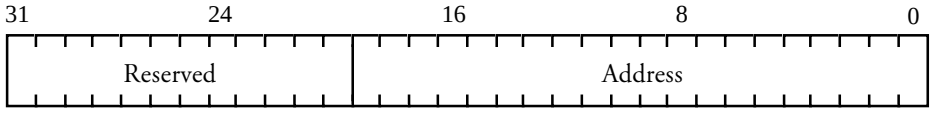
Bits 20-31 Reserved

3.5.39 VSBVideoAddress2

VSBVideoAddress2

Region: Region 0 Read/Write

Offset: 5938h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

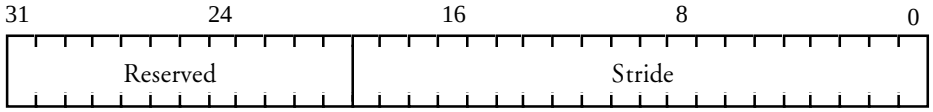
Bits 20-31 Reserved

3.5.40 VSBVideoStride

VSBVideoStride

Region: Region 0 Read/Write

Offset: 5A40h Reset Value: undefined



Bits 0-19 Stride between scanlines (64 bit aligned)

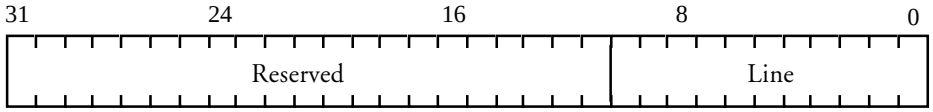
Bits 20-31 Reserved

3.5.41 VSBVideoStartLine

VSBVideoStartLine

Region: Region 0 Read/Write

Offset: 5A48h Reset Value: undefined



Bits 0-10 First scanline of video data

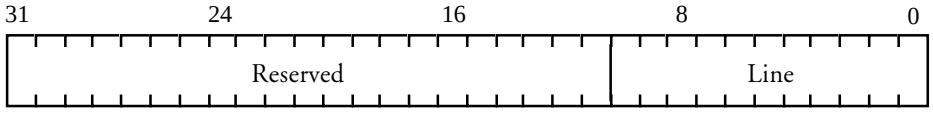
Bits 11-31 Reserved

3.5.42 VSBVideoEndLine

VSBVideoEndLine

Region: Region 0 Read/Write

Offset: 5A50h Reset Value: undefined



Bits 0-10 First scanline after video data

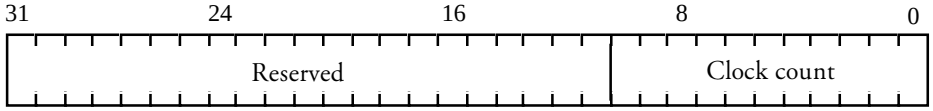
Bits 11-31 Reserved

3.5.43 VSBVideoStartData

VSBVideoStartData

Region: Region 0 Read/Write

Offset: 5A58h Reset Value: undefined



Bits 0-10 First clock of valid data.

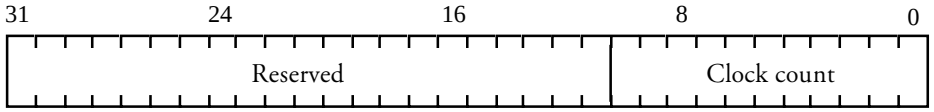
Bits 11-31 Reserved

3.5.44 VSBVideoEndData

VSBVideoEndData

Region: Region 0 Read/Write

Offset: 5A60h Reset Value: undefined



Bits 0-10 First clock after active video.

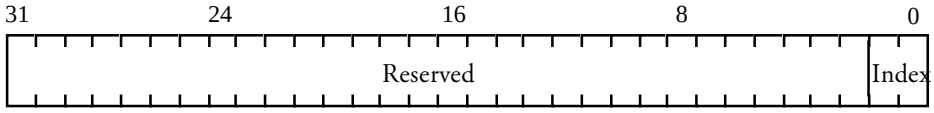
Bits 11-31 Reserved

3.5.45 VSBVBIAddressHost

VSBVBIAddressHost

Region: Region 0 Read/Write

Offset: 5A68h Reset Value: undefined



Bits 0-1 Host base address register index.

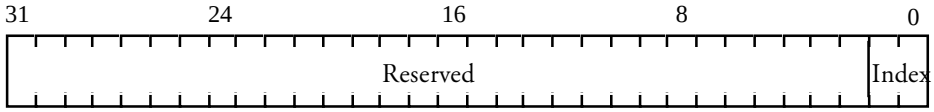
Bits 2-31 Reserved

3.5.46 VSBVBIAddressIndex

VSBVBIAddressIndex

Region: Region 0 Read/Write

Offset: 5A70h Reset Value: 0000.0000h



Bits 0-1 Base address register index.

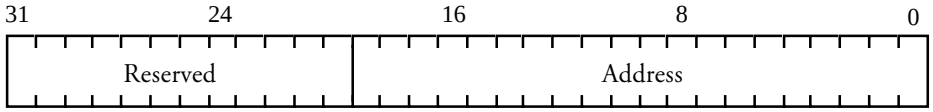
Bits 2-31 Reserved

3.5.47 VSBVBIAddress0

VSBVBIAddress0

Region: Region 0 Read/Write

Offset: 5A78h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

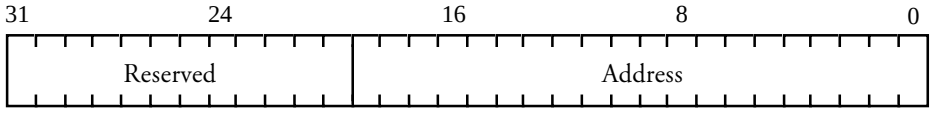
Bits 20-31 Reserved

3.5.48 VSBVideoAddress1

VSBVBIAddress1

Region: Region 0 Read/Write

Offset: 5A80h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

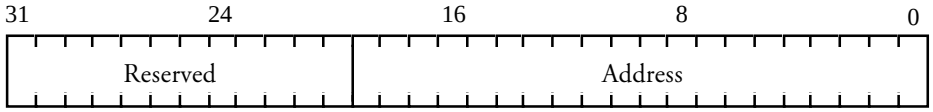
Bits 20-31 Reserved

3.5.49 VSAVBIAddress2

VSBVBIAddress2

Region: Region 0 Read/Write

Offset: 5A88h Reset Value: undefined



Bits 0-19 Base address (64 bit aligned)

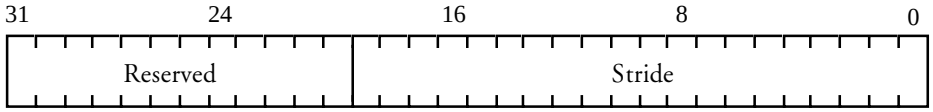
Bits 20-31 Reserved

3.5.50 VSBVBIStride

VSBVBIStride

Region: Region 0 Read/Write

Offset: 5A90h Reset Value: undefined



Bits 0-19 Stride between scanlines (64 bit aligned)

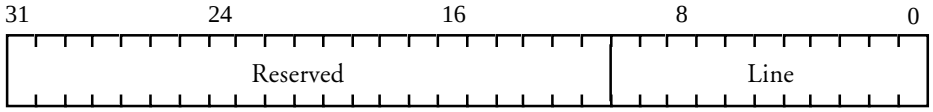
Bits 20-31 Reserved

3.5.51 VSBVBIStartLine

VSBVBIStartLine

Region: Region 0 Read/Write

Offset: 5A98h Reset Value: undefined



Bits 0-10 First scanline of VBI data

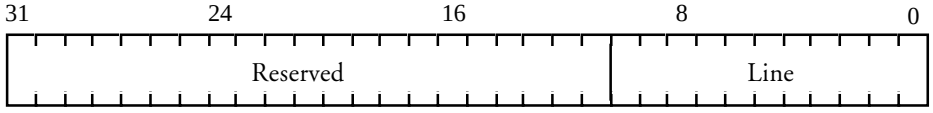
Bits 11-31 Reserved

3.5.52 VSBVBIEndLine

VSBVBIEndLine

Region: Region 0 Read/Write

Offset: 5AA0h Reset Value: undefined



Bits 0-10 First scanline after VBI data

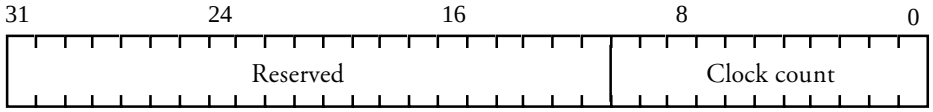
Bits 11-31 Reserved

3.5.53 VSBVBIStartData

VSBVBIStartData

Region: Region 0 Read/Write

Offset: 5AA8h Reset Value: undefined



Bits 0-10 First clock of valid data in VBI line.

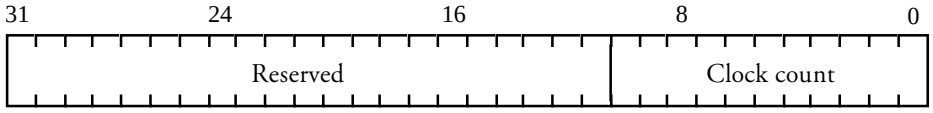
Bits 11-31 Reserved

3.5.54 VSBVBIEndData

VSBVBIEndData

Region: Region 0 Read/Write

Offset: 5AB0h Reset Value: undefined



Bits 0-10 First clock after active VBI data.

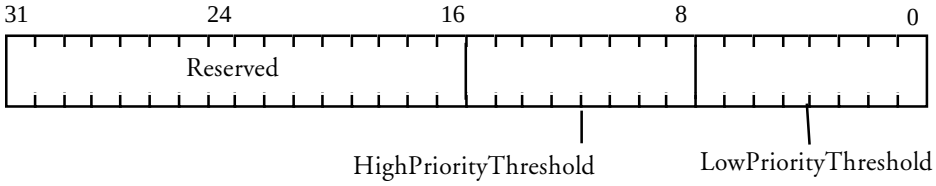
Bits 11-31 Reserved

3.5.55 VSBFifoControl

VSBFifoControl

Region: Region 0 Read/Write

Offset: 5AB8h Reset Value: undefined



Bits 0-7 Low priority threshold

Bits 8-15 High priority threshold

Bits 16-31 Reserved

3.6 RAMDAC Registers

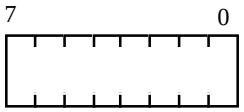
Refer to the RAMDAC section for details on programming these registers.

3.6.1 Palette/Cursor RAM Write Address Register

RDPaletteWriteAddress

Region: Region 0 Read/Write

Offset: 4000h Reset Value: undefined



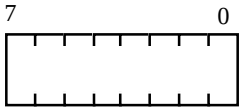
Bits 0-7 Address

3.6.2 Palette/Cursor RAM Data Register

RDPaletteData

Region: Region 0 Read/Write

Offset: 4008h Reset Value: undefined



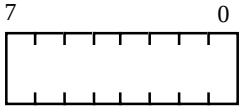
Bits 0-7 Data

3.6.3 Pixel Read Mask Register

RDPixelMask

Region: Region 0 Read/Write

Offset: 4010h Reset Value: FFh



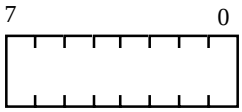
Bits 0-7 Mask

3.6.4 Palette/Cursor Read Address Register

RDPaletteReadAddress

Region: Region 0 Read/Write

Offset: 4018h Reset Value: undefined



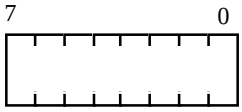
Bits 0-7 Address

3.6.5 Cursor Color Address Register

RDCursorColorAddress

Region: Region 0 Read/Write

Offset: 4020h Reset Value: undefined



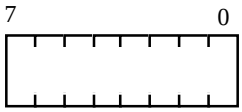
Bits 0-7 Address

3.6.6 Cursor Color Data Register

RDCursorColorData

Region: Region 0 Read/Write

Offset: 4028h Reset Value: undefined



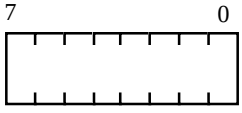
Bits 0-7 Data

3.6.7 Indexed Data Register

RDIndexedData

Region: Region 0 Read/Write

Offset: 4050h Reset Value: undefined



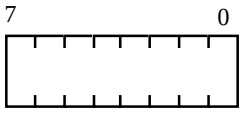
Bits 0-7 Data

3.6.8 Cursor RAM Data Register

RDCursorRAMData

Region: Region 0 Read/Write

Offset: 4058h Reset Value: undefined



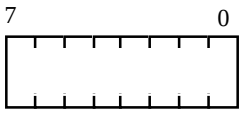
Bits 0-7 Data

3.6.9 Cursor Position X Low Register

RDCursorXLow

Region: Region 0 Read/Write

Offset: 4060h Reset Value: undefined



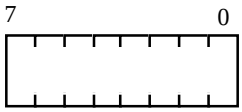
Bits 0-7 Data

3.6.10 Cursor Position X High Register

RDCursorXHigh

Region: Region 0 Read/Write

Offset: 4068h Reset Value: undefined



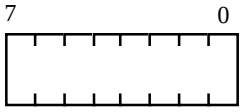
Bits 0-7 Data

3.6.11 Cursor Position Y Low Register

RDCursorYLow

Region: Region 0 Read/Write

Offset: 4070h Reset Value: undefined

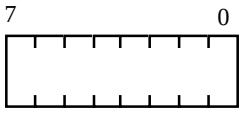


Bits 0-7 Data

3.6.12 Cursor Position Y High Register

RDCursorYHigh

Region:	Region 0	Read/Write
Offset:	4070h	Reset Value: undefined

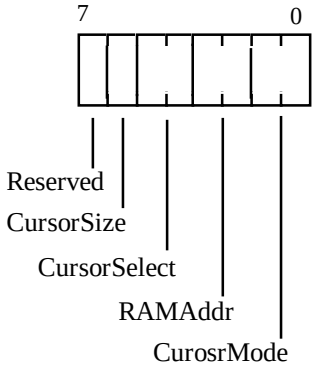


Bits 0-7 Data

3.6.13 Cursor Control Register

RDCursorControl

Region:	Region 0	Read/Write
Indirect index:	06h	Reset Value: 00h



Bits 0-1 CursorMode
0 = Cursor disabled
1 = 3 color cursor
2 = XGA cursor
3 = X Windows cursor

Bits 2-3 RAMAddr

Bits 4-5 CursorSelect
0 = Cursor 0
1 = Cursor 1
2 = Cursor 2
3 = Cursor 3

Bit 6 CursorSize
0 = 32x32
1 = 64x64

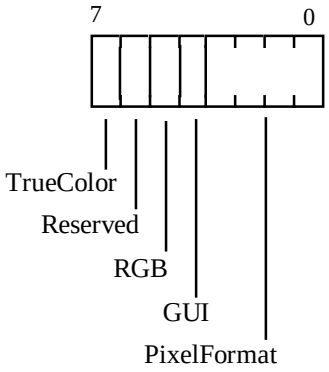
Bit 7 Reserved

3.6.14 Color Mode Register

RDColorMode

Region: Region 0 Read/Write

Indirect index: 018h Reset Value: 00h



Bits 0-3 PixelFormat
0 = CI8/SVGA
1 = RGB332
2 = RGB232Offset
3 = RGBA2321
4 = RGBA5551
5 = RGBA4444
6 = RGB565
8 = RGBA8888
9 = RGB888

Bit 4 GUI
0 = Disabled (SVGA)
1 = Enabled (GUI)

Bit 5 RGB
0 = Disabled (BGR)
1 = Enabled (RGB)

Bit 6 Reserved

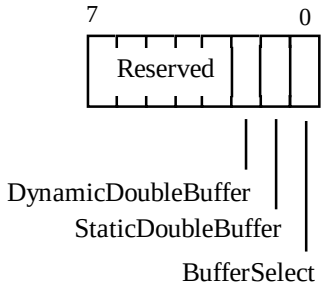
Bit 7 TrueColor
0 = Disabled (does not use palette)
1 = Enabled (uses palette)

3.6.15 Mode Control Register

RDModeControl

Region: Region 0 Read/Write

Indirect index: 019h Reset Value: 00h



Bit 0 BufferSelect
0 = Front
1 = Back
Selects buffer to display when static double buffering is enabled.

Bit 1 StaticDoubleBuffer
0 = Disabled
1 = Enabled

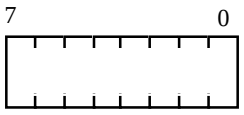
Bit 2 DynamicDoubleBuffer
0 = Disabled
1 = Enabled
Enables per-pixel double buffering, buffer displays depends on state of bit 31 of the pixel.

3.6.16 Palette Page Register

RDPalettePage

Region: Region 0 Read/Write

Indirect index: 01Ch Reset Value: 00h



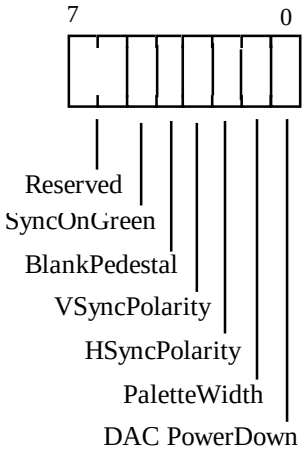
Bits 0-7 Page
Specifies the additional bit planes when the overlay pixel has less than 8 bits per component. The extra bits are used to pad the low order bits of the overlay value.

3.6.17 Miscellaneous Control Register

RDMiscControl

Region: Region 0 Read/Write

Indirect index: 01Eh Reset Value: 00h



Bit 0 DACPowerDown
0 = Disable
1 = Enable

Bit 1 PaletteWidth
0 = 6 bits
1 = 8 bits

Bit 2 HSyncPolarity
0 = Non-inverted
1 = Inverted

Bit 3 VSyncPolarity
0 = Non-inverted
1 = Inverted

Bit 4 BlankPedestal
0 = Disabled
1 = Enabled

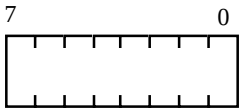
Bit 5 SyncOnGreen
0 = Disabled
1 = Enabled

Bits 6-7 Reserved

3.6.18 Pixel Clock Register A1

RDPixelClockA1

Region:	Region 0	Write
Indirect index:	020h	Reset Value: 1Ch

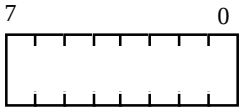


Bits 0-7	Parameter M

3.6.19 Pixel Clock Register A2

RDPixelClockA2

Region:	Region 0	Write
Indirect index:	021h	Reset Value: 02h

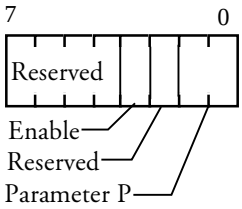


Bits 0-7	Parameter N

3.6.20 Pixel Clock Register A3

RDPixelClockA3

Region:	Region 0	Write
Indirect index:	022h	Reset Value: 0Bh

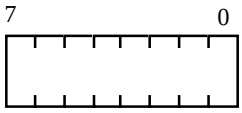


Bits 0-1	Parameter P
Bit 2	Reserved
Bit 3	Enable
Bits 4-7	Reserved

3.6.21 Pixel Clock Register B1

RDPixelClockB1

Region:	Region 0	Write
Indirect index:	023h	Reset Value: 10h

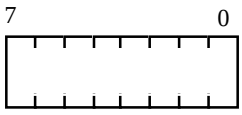


Bits 0-7	Parameter M

3.6.22 Pixel Clock Register B2

RDPixelClockB2

Region:	Region 0	Write
Indirect index:	024h	Reset Value: 02h

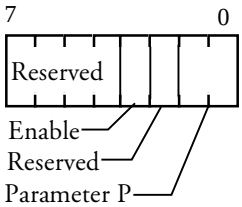


Bits 0-7	Parameter N

3.6.23 Pixel Clock Register B3

RDPixelClockB3

Region:	Region 0	Write
Indirect index:	025h	Reset Value: 0Bh

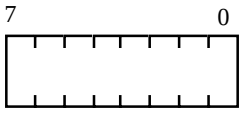


Bits 0-1	Parameter P
Bit 2	Reserved
Bit 3	Enable
Bits 4-7	Reserved

3.6.24 Pixel Clock Register C1

RDPixelClockC1

Region:	Region 0	Write
Indirect index:	026h	Reset Value: 00h

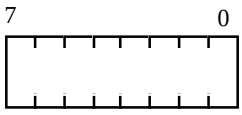


Bits 0-7	Parameter M

3.6.25 Pixel Clock Register C2

RDPixelClockC2

Region:	Region 0	Write
Indirect index:	027h	Reset Value: 00h

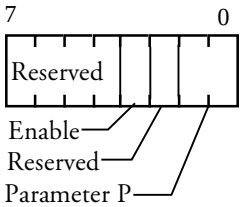


Bits 0-7	Parameter N

3.6.26 Pixel Clock Register C3

RDPixelClockC3

Region:	Region 0	Write
Indirect index:	028h	Reset Value: 00h



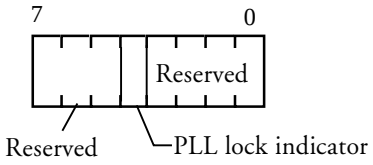
Bits 0-1	Parameter P
Bit 2	Reserved
Bit 3	Enable
Bits 4-7	Reserved

3.6.27 Pixel Clock Status Register

RDPixelClockStatus

Region: Region 0 Read/Write

Indirect index: 029h Reset Value: undefined



Bits 0-3 Reserved

Bit 4 PLL lock indicator.
0 = out-of-lock
1 = locked

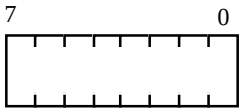
Bits 5-7 Reserved

3.6.28 Memory Clock Register 1

RDMemoryClock1

Region: Region 0 Write

Indirect index: 030h Reset Value: 1Ch



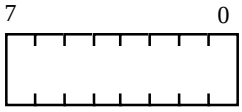
Bits 0-7 Parameter M

3.6.29 Memory Clock Register 2

RDMemoryClock2

Region: Region 0 Write

Indirect index: 031h Reset Value: 02h

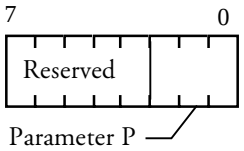


Bits 0-7 Parameter N

3.6.30 Memory Clock Register 3

RDMemoryClock3

Region:	Region 0	Write	
Indirect index:	032h	Reset Value:	02h

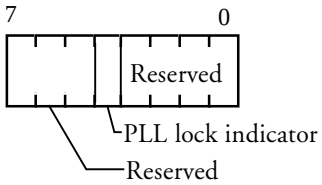


Bits 0-1	Parameter P
Bits 2-7	Reserved

3.6.31 Memory Clock Status Register

RDMemoryClockStatus

Region:	Region 0	Read/Write	
Indirect index:	033h	Reset Value:	undefined



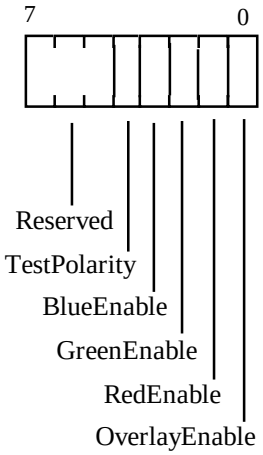
Bits 0-3	Reserved
Bit 4	PLL lock indicator. 0 = out-of-lock 1 = locked
Bits 5-7	Reserved

3.6.32 Color Key Control Register

RDColorKeyControl

Region: Region 0 Read/Write

Indirect index: 040h Reset Value: 00h



Bit 0 OverlayEnable
0 = Disabled (do not test overlay)
1 = Enabled (test overlay)

Bit 1 RedEnable
0 = Disabled (do not test red component of pixel)
1 = Enabled (test red component of pixel)

Bit 2 GreenEnable
0 = Disabled (do not test green component of pixel)
1 = Enabled (test green component of pixel)

Bit 3 BlueEnable
0 = Disabled (do not test blue component of pixel)
1 = Enabled (test blue component of pixel)

Bit 4 TestPolarity
0 = True (display overlay if tests pass)
1 = False (do not display overlay if tests pass)

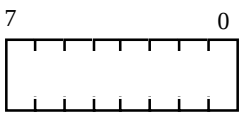
Bits 5-7 Reserved

3.6.33 Overlay Key Register

RDOverlayKey

Region: Region 0 Read/Write

Indirect index: 041h Reset Value: 00h



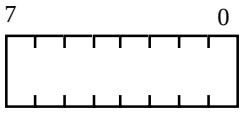
Bits 0-7 Data

3.6.34 Red Key Register

RDRedKey

Region: Region 0 Read/Write

Indirect index: 042h Reset Value: 00h



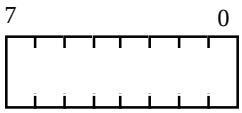
Bits 0-7 Data

3.6.35 Green Key Register

RDGreenKey

Region: Region 0 Read/Write

Indirect index: 043h Reset Value: 00h



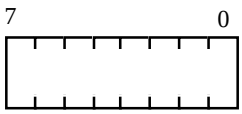
Bits 0-7 Data

3.6.36 Blue Key Register

RDBlueKey

Region: Region 0 Read/Write

Indirect index: 044h Reset Value: 00h



Bits 0-7 Data

4. Memory System

The PERMEDIA memory system is intended for use with SGRAM or SDRAM memory devices. A typical organization is shown below.

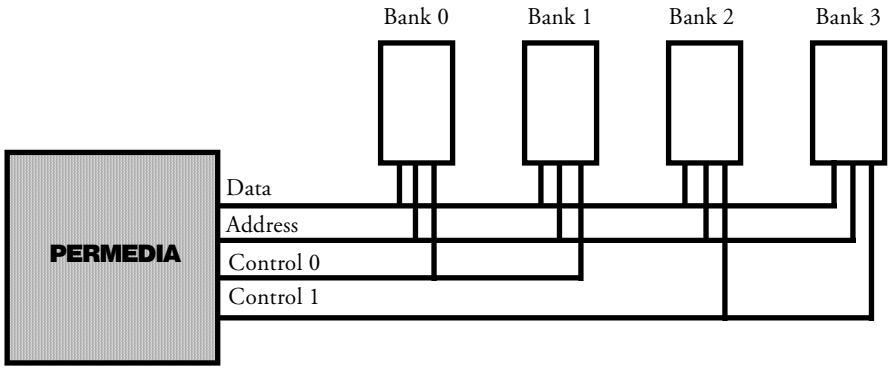


Figure 4.1 Organization of memory devices

Each bank is made up of two 32 bit wide devices. The data and address lines are common to all the memory devices. There are two sets of control lines which are provided to reduce loading: they are driven identically. The example above shows one set of control lines driving bank 0 and bank 1, with the second set driving bank 2 and bank 3. This organization is preferable if the second two banks are on a mezzanine connector as it eases routing and termination of clock lines. Alternatively the control lines could be split along the upper and lower devices in each bank.

4.1 System Parameters

The various timing parameters used to control synchronous memories can be adjusted to allow for optimum performance depending on memory speed grade and the PERMEDIA system clock frequency (MClk).

The following parameters are used to control accesses to the memory. These values are usually set at reset from configuration resistors. Each field defines the operation of the memory controller, with the exception of the boot address field which is loaded directly into the memory device. It is very important that the boot address is consistent with the memory controller parameters, failing to do this may damage the memory devices.

The reset configuration may be over-written by software following reset (see 3.3.4).

4.1.1 RowCharge

This flag determines the method used to load the Special Mode Register in the SGRAM. If set high, the memory controller assumes that the Special Mode Register may be accessed when one internal bank is IDLE and the other is ROW-ACTIVE.

4.1.2 TimeRCD

This parameter defines, in MClk cycles, the time from issuing a RAS to the SGRAM before CAS is active. This is usually referred to in the SGRAM data sheets as tRCD and will be assigned by the memory controller as TimeRCD + 2.

4.1.3 TimeRC

This parameter defines, in MClk cycles, the time from issuing an AUTO-REFRESH command to the SGRAM being able to accept another command. This is usually referred to in the SGRAM data sheets as tRC and will be assigned by the memory controller as TimeRC + 2.

4.1.4 TimeRP

This parameter defines, in MClk cycles, the time from issuing a PRECHARGE command to the SGRAM being able to accept an ACTIVATE command (RAS). This is usually referred to in the SGRAM data sheets as tRP, and will be assigned by the memory controller as TimeRP + 1.

4.1.5 CASLatency3

This flag determines the CAS latency expected by the memory controller. If set high the controller expects the SGRAM to be operating with a CAS latency of 3. If set low the controller expects a CAS latency of 2.

4.1.6 BootAddress

This parameter defines the value of the Mode Register loaded into the SGRAM at the end of the boot sequence (see data sheet). Items to note : Burst type should be sequential, burst length should be consistent with the Burst1Cycle flag and CAS latency should be consistent with the CAS3Latency flag. All other bits in the BootAddress field should be set low.

4.1.7 NumberBanks

This field defines the size of SGRAM array being used. Values are "00" = 2Mbytes, "01" = 4Mbytes, "10" = 6Mbytes, "11" = 8Mbytes.

4.1.8 RefreshCount

This parameter defines the period between AUTO-REFRESH commands being issued to the SGRAM. The count is in MClks/16 i.e. if RefreshCount = 1, the SGRAM will be refreshed every 16 MClk cycles. For the required refresh rate, see the SGRAM data sheet.

4.1.9 TimeRASMin

"This parameter defines, in MClk cycles, the minimum Row Active time. This is sometimes referred to as the Active to Precharge period. It is assigned by the memory controller as tRAS (in MClk cycles) + 3. However, loading the MemConfig register field or setting the config resistors to a value of 0 is invalid."

4.1.10 DeadCycleEnable

If this flag is set high the memory controller will insert a turnaround cycle when changing from a READ to a WRITE command. Some SGRAM speed grades may require this.

4.1.11 BankDelay

This parameter defines the READ burst length of the SGRAM being used. It should always be consistent with the BootAddress parameter and should be set to burst length - 1.

4.1.12 Burst1Cycle

This flag, if set high, allows the memory controller to assume a burst length of 1 for the SGRAM. It should always be consistent with the SGRAM BootAddress parameter.

4.1.13 SDRAM

If this flag is set high, the memory controller will assume that SDRAM is fitted as the memory array. It will disable all block fills and bit-masked writes. Any block fill operations will be ignored, any masked writes will be converted to non-masked writes.

4.2 Recommended Parameter Values

The following values are recommended for a PERMEDIA system running with an MClk of 50MHz and using Samsung SGRAM (-12) parts. The total SGRAM size is 4MB.

RowCharge	0
TimeRCD	000
TimeRC	0100
TimeRP	001
CAS3Latency	0
BootAddress	000010000
NumberBanks	01
RefreshCount	00110000
TimeRASMin	001
DeadCycle	0
BankDelay	000
Burst1Cycle	1
SDRAM	0

Table 4.1 50MHz/Samsung SGRAM (-12)/total SGRAM 4MB.

The following values are recommended for a PERMEDIA system running with an MClk of 66MHz and using NEC SGRAM (-12) parts. The total SGRAM size is 6MB. The CAS3Latency flag is set high. This is due to the access time of the (-12) part relative to the bus speed. If a CAS latency of 2 was used, the part may not drive data onto the bus in time for the read cycle to complete.

RowCharge	1
TimeRCD	001
TimeRC	0110
TimeRP	010
CAS3Latency	1
BootAddress	000110000
NumberBanks	10
RefreshCount	01000001
TimeRASMin	001
DeadCycle	1
BankDelay	000
Burst1Cycle	1
SDRAM	0

Table 4.2 66MHz/NEC SGRAM (-12)/total SGRAM 6MB

The following values are recommended for a PERMEDIA system running with an MClk of 66MHz and using NEC SGRAM (-10) parts. The total SGRAM size is 8MB.

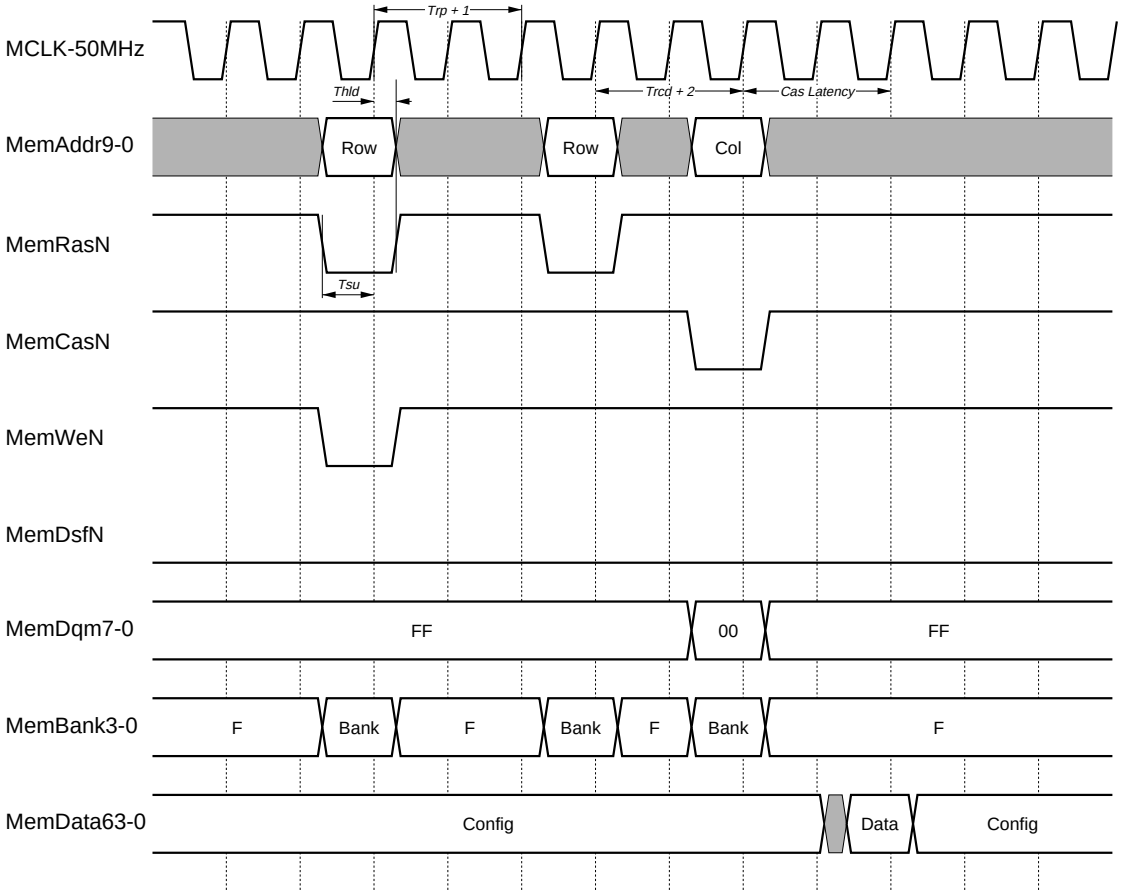
RowCharge	1
TimeRCD	001
TimeRC	0110
TimeRP	001
CAS3Latency	0
BootAddress	000010000
NumberBanks	11
RefreshCount	01000001
TimeRASMin	001
DeadCycle	1
BankDelay	000
Burst1Cycle	1
SDRAM	0

Table 4.3 66MHz/NEC SGRAM (-10)/total SGRAM 8MB

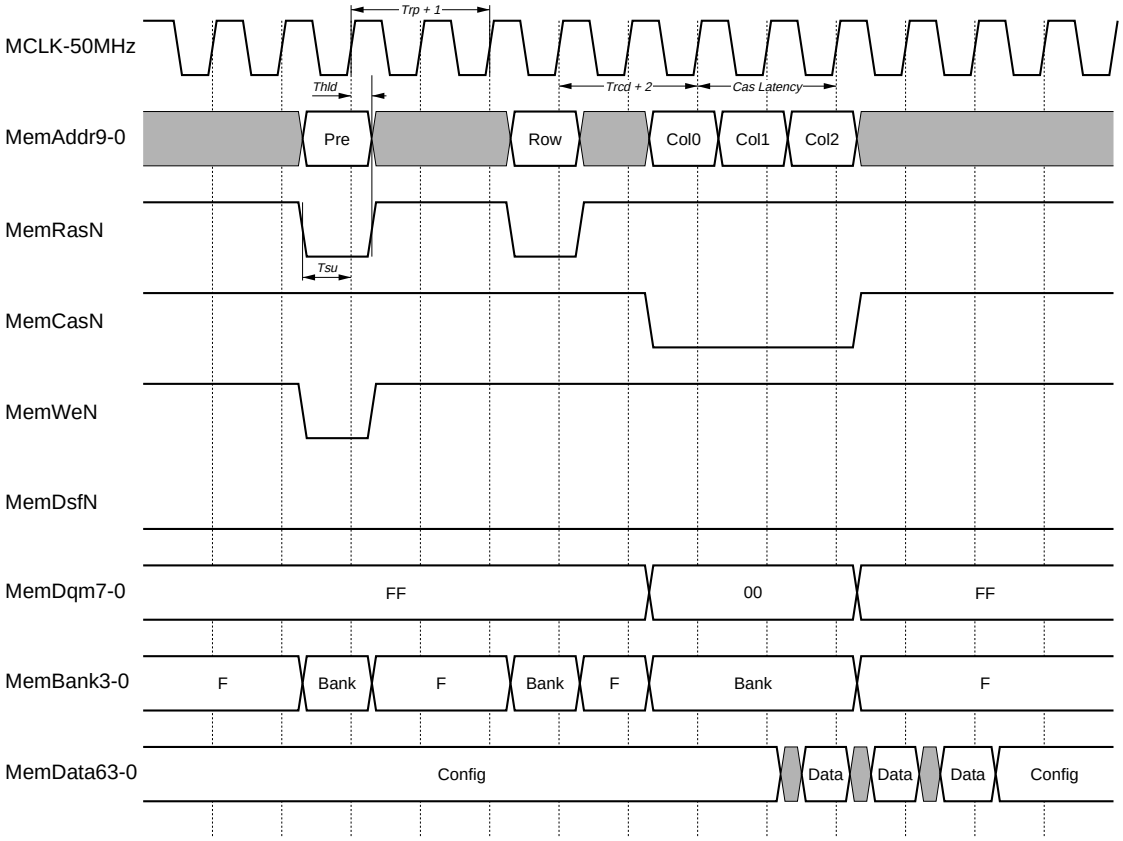
4.3 Timing Diagrams

The following timing diagrams show specific operations of the memory controller.

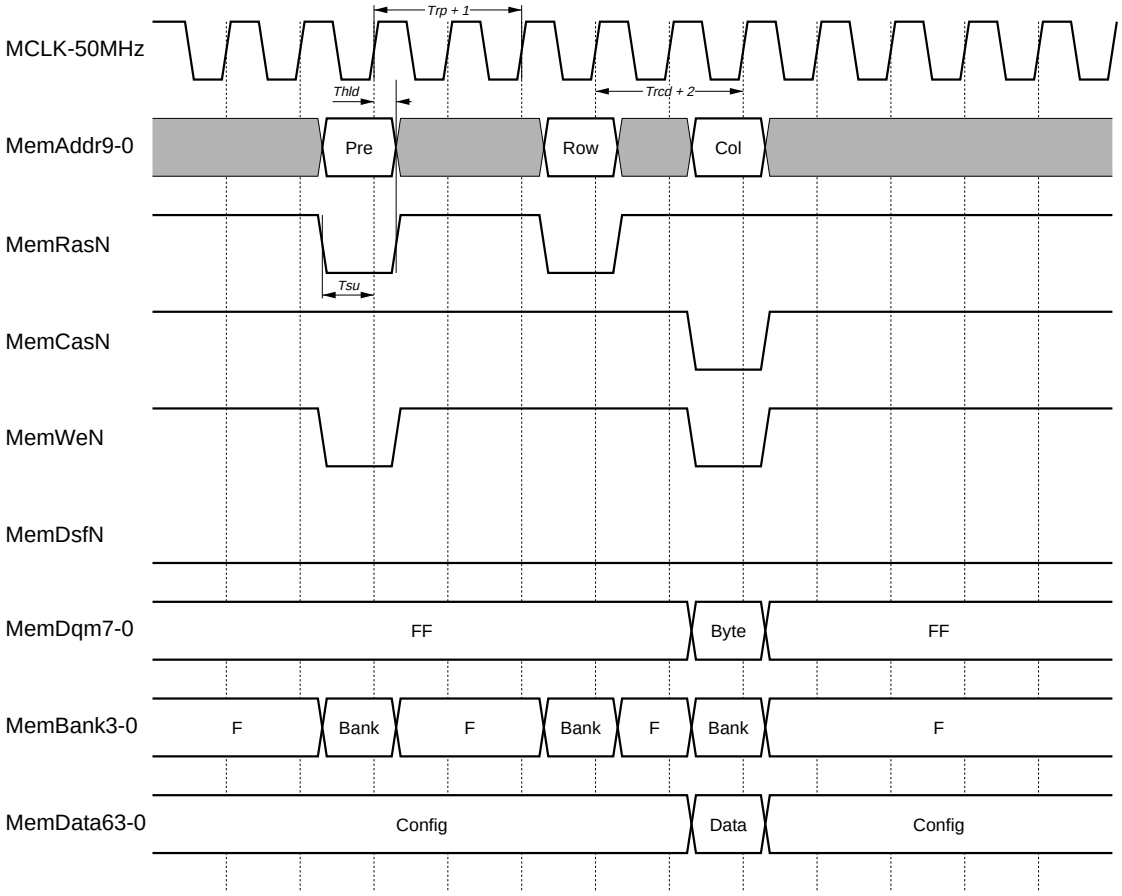
4.3.1 Single Read with Precharge



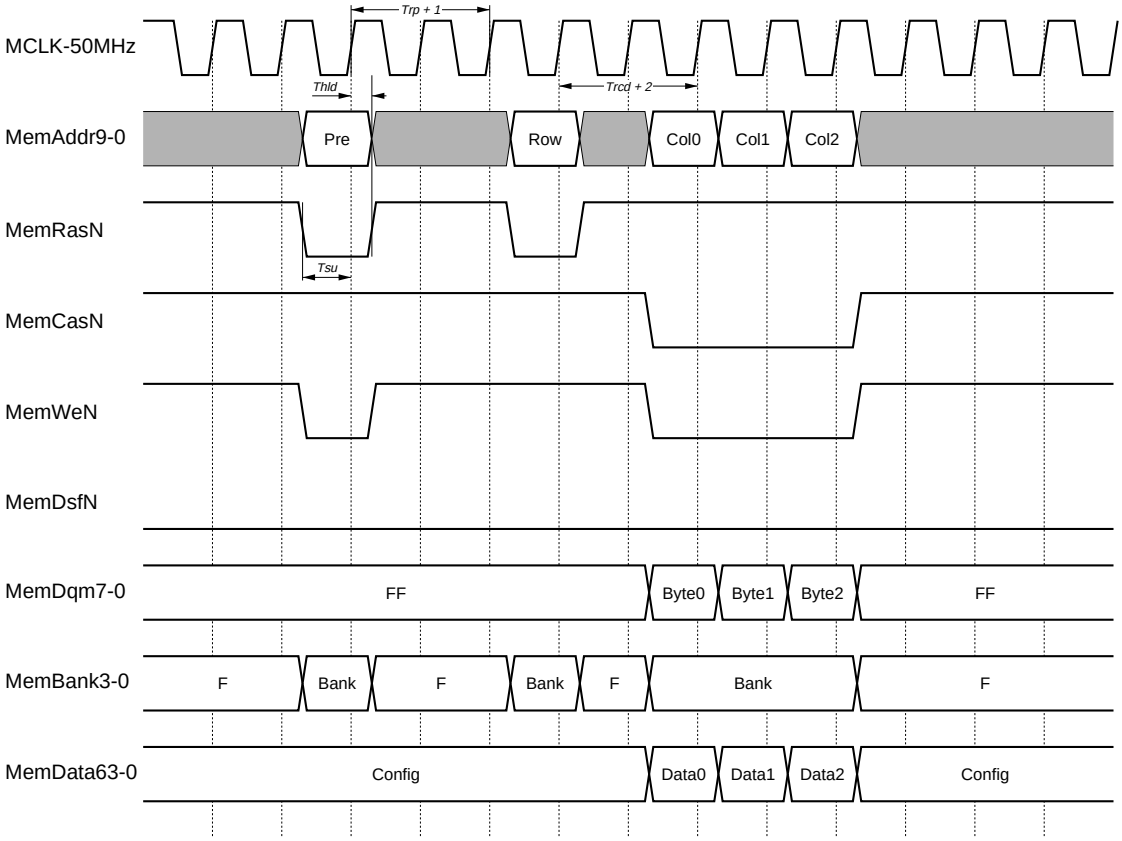
4.3.2 Multiple Reads to Same Bank



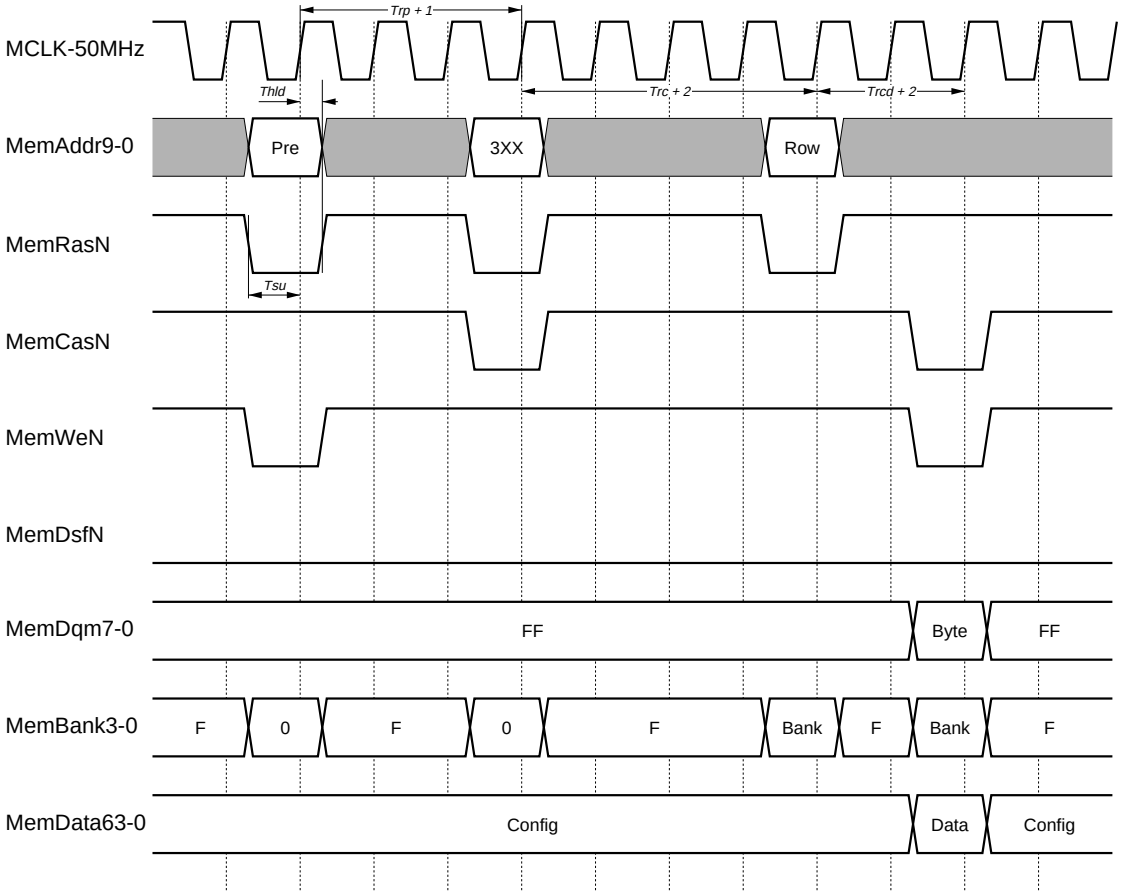
4.3.3 Single Write with Precharge



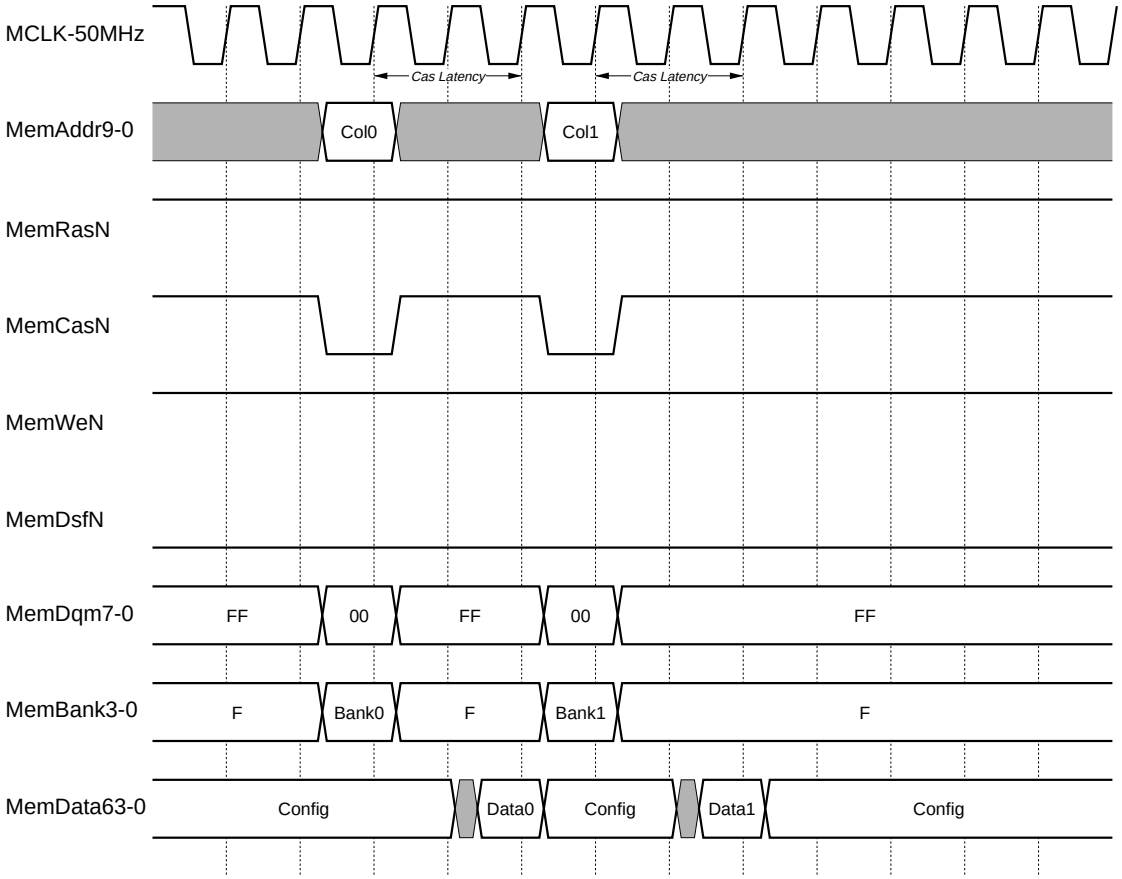
4.3.4 Multiple Writes to Same Bank



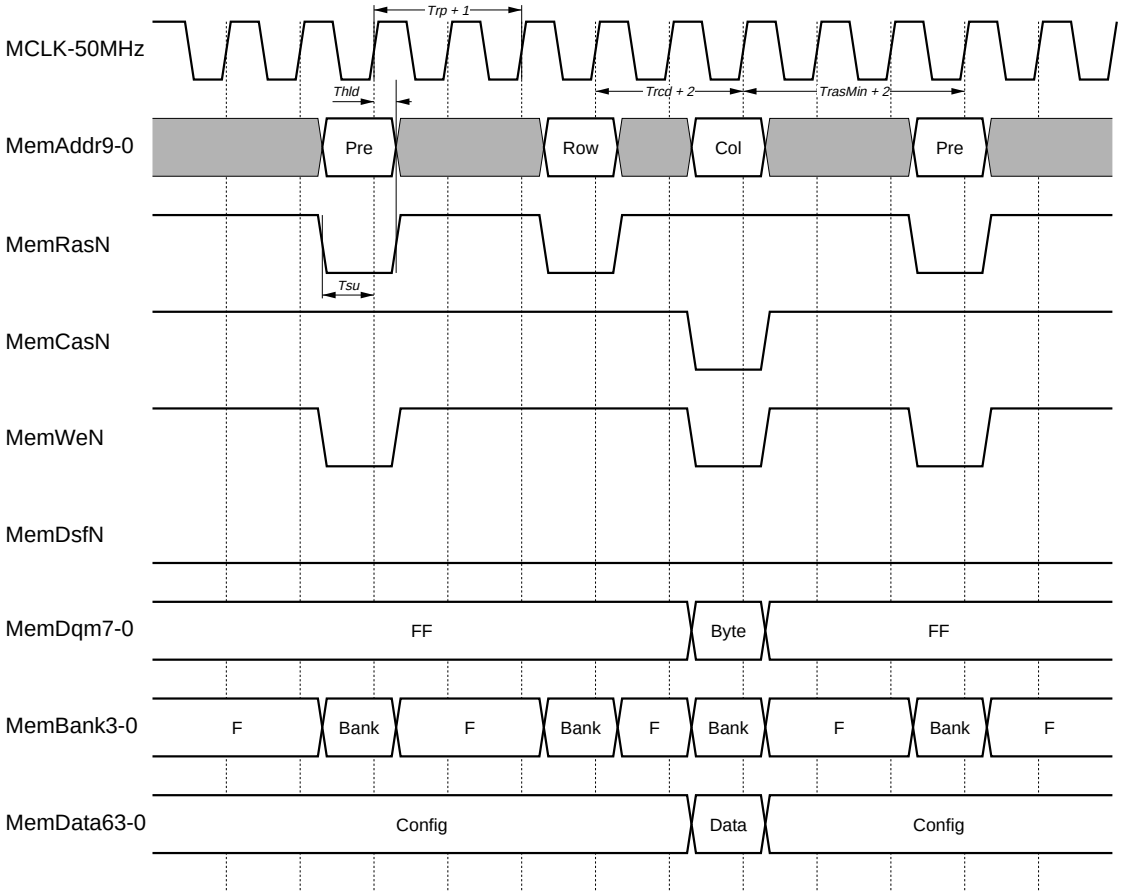
4.3.5 Refresh Followed by Access



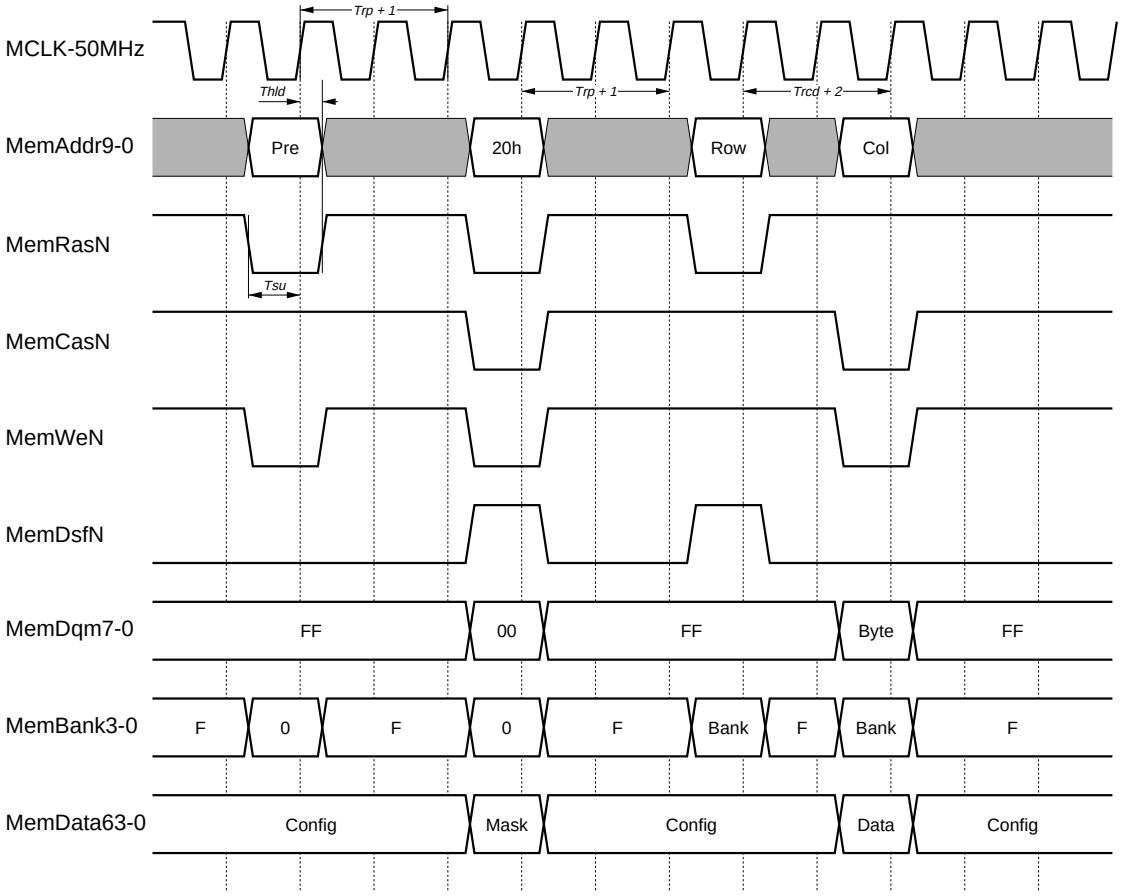
4.3.6 Multiple Reads From Different Banks



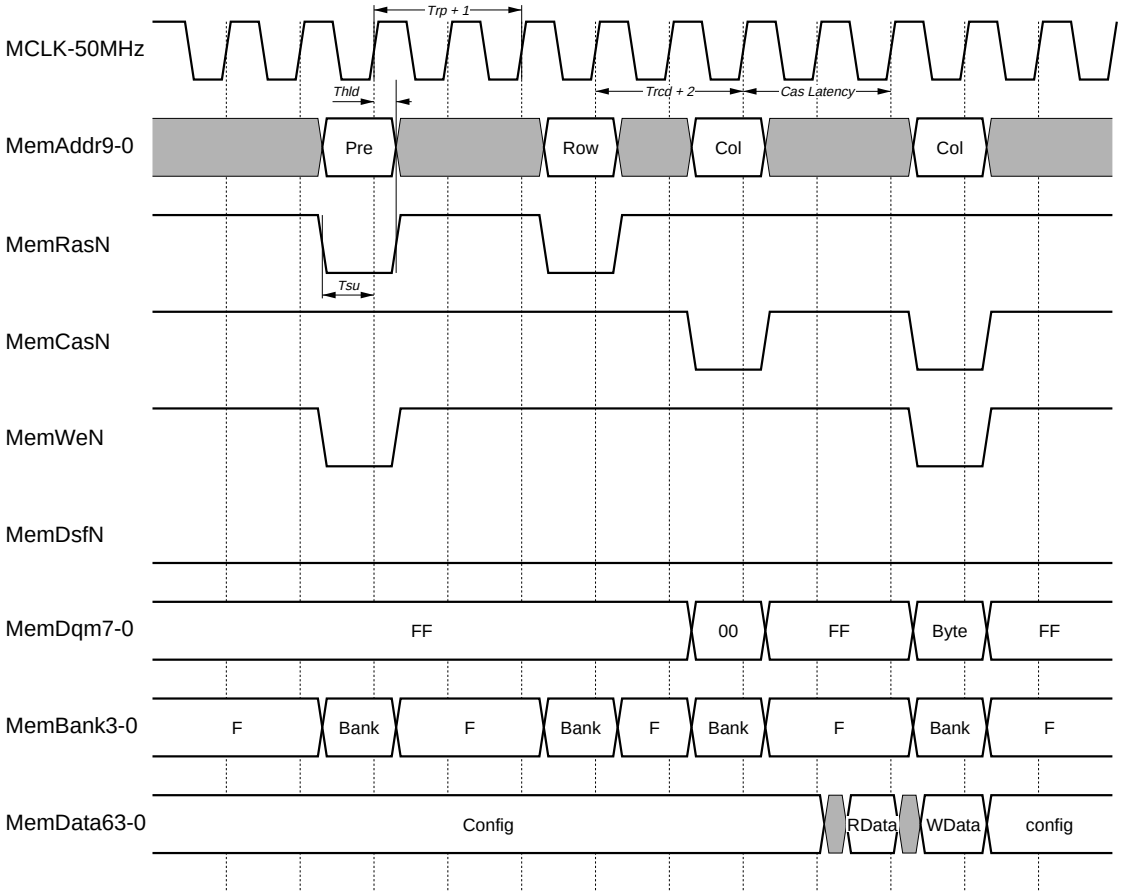
4.3.7 RAS Minimum Access Timing



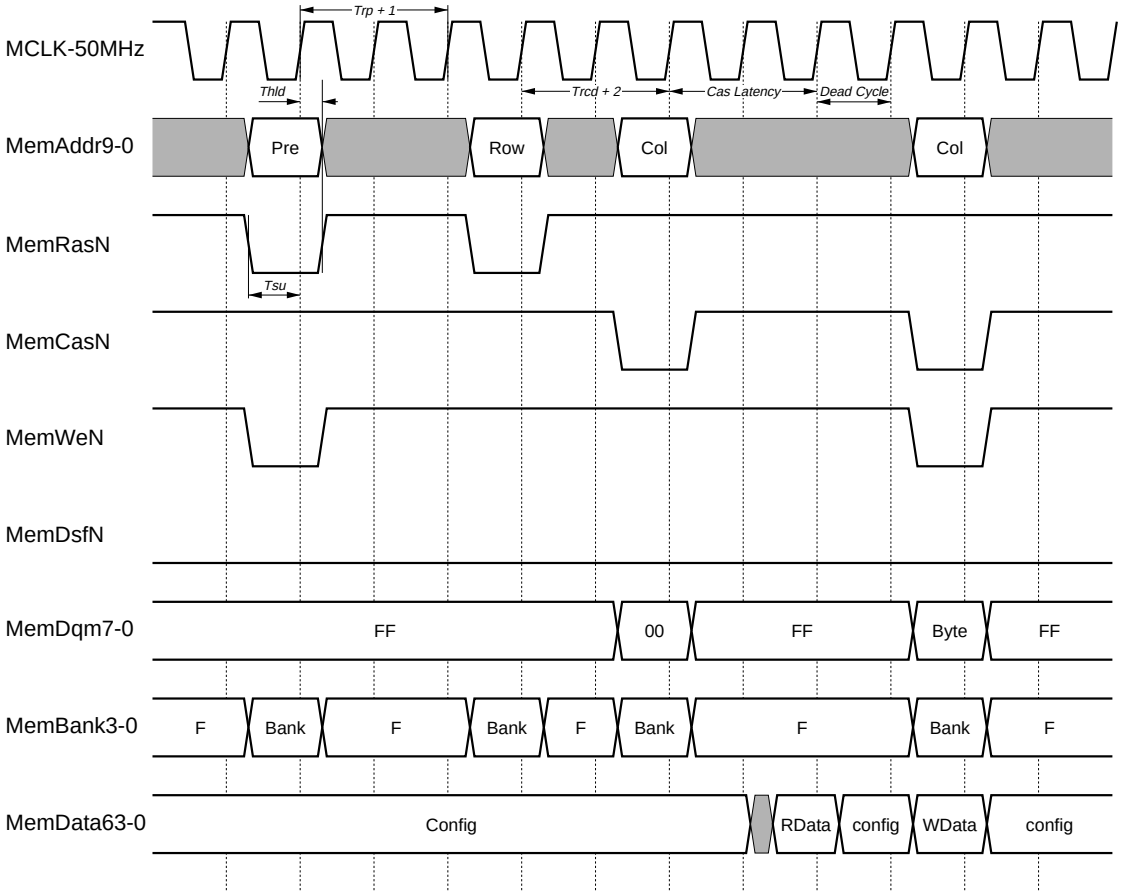
4.3.8 Mask Load Followed by Masked Write



4.3.9 Read Followed by Write - No Dead Cycle



4.3.10 Read Followed by Write - With Dead Cycle



5. Video Unit and RAMDAC

The video unit and RAMDAC should be configured to display the framebuffer data with the format, resolution, and refresh frequency required.

5.1 Using the Video Unit

The diagram below shows the parameters that are used to control the display of images generated by the graphics processor. Any images generated by the SVGA unit are displayed by the SVGA which should be programmed in accordance with normal SVGA practice.

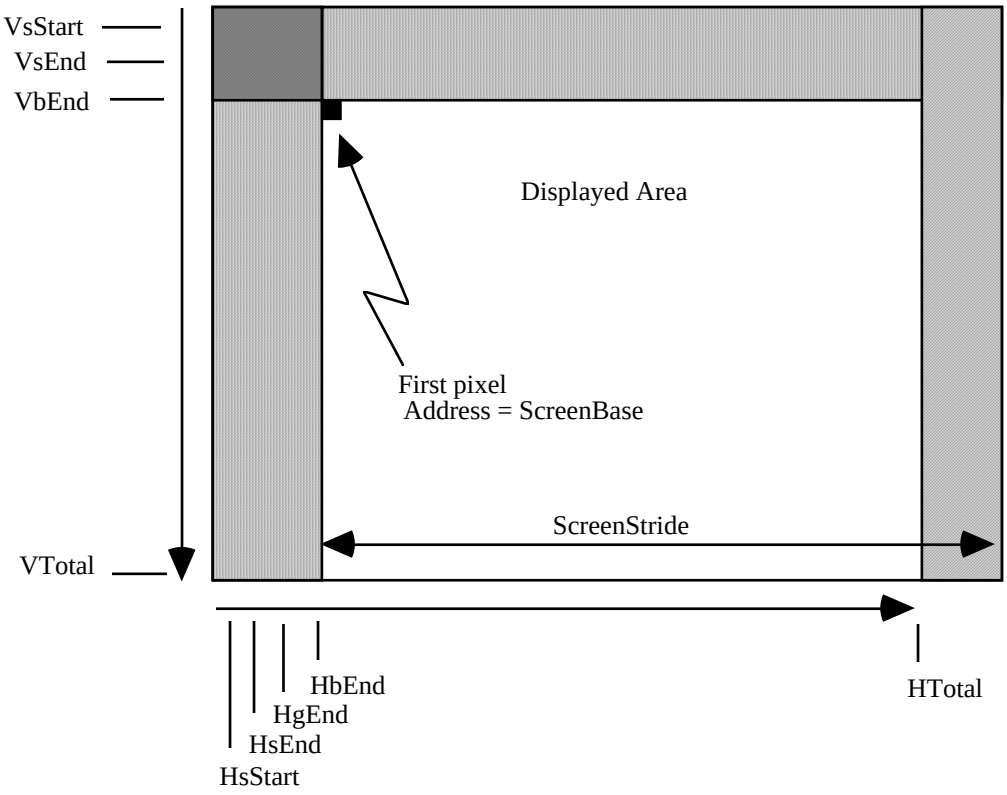


Figure 5.1 Video Timing Parameters

5.2 Example Timing Values

5.2.1 Timing Values for 640x480 16 BPP 75Hz

Parameter	Hex	Decimal
HTotal	000001A3	419
HsStart	00000008	8
HsEnd	00000028	40
HbEnd	00000064	100
HgEnd	00000064	100
VTotal	000001F3	499
VsStart	00000000	0
VsEnd	00000003	3
VbEnd	00000014	20
ScreenStride	000000A0	160
ScreenBase	00000000	0
VideoControl	00000029	41

5.2.2 Timing Values for 800x600 32 BPP 75Hz

Parameter	Hex	Decimal
HTotal	0000041F	1055
HsStart	00000010	16
HsEnd	00000060	96
HbEnd	00000100	256
HgEnd	00000100	256
VTotal	00000270	624
VsStart	00000000	0
VsEnd	00000003	3
VbEnd	00000019	25
ScreenStride	00000190	400
ScreenBase	00000000	0
VideoControl	00000029	41

5.3 Display Data Channel

The DDC interface allows PERMEDIA to read timing information from a compatible monitor. Both DDC1 and DDC2 protocols are supported.

For DDC1, the data is read one bit at a time, and is clocked from the monitor by the vertical sync signal. The vertical sync should be controlled directly from software using the video configuration register.

Vertical sync should be driven high and the data will become valid 30 microseconds later; when the data has been read the vertical sync should be driven low for at least 20 microseconds before it is driven high again. Accurate timing can be derived from the counter in the memory controller register group.

For DDC2, an I2C bus interface is used. For details on using this see the appropriate section in this manual.

5.4 RAMDAC

The RAMDAC registers are either accessed directly or indirectly. The tables below show the direct and the indirect registers. The direct registers are accessed by reading or writing the appropriate offset. The indirect registers are accessed by writing their respective index to the Index Register (offset 00h) and, then, either reading or writing from the Indexed Data Register (offset 0Ah). For example, if Color Mode register is to be written value A5h, first write 18h at offset 00h, then write A5h at offset 0Ah.

Offset (hex)	Mode	
00	R/W	Palette/Cursor RAM Write Address Register
01	R/W	Palette RAM Data Register
02	R/W	Pixel Read Mask
03	R/W	Palette/Cursor RAM Read Address Register
04	R/W	Cursor Color Address Register
05	R/W	Cursor Color Data Register
06		Reserved
07		Reserved
08		Reserved
09		Reserved
0A	R/W	Indexed Data Register
0B	R/W	Cursor RAM Data Register
0C	R/W	Cursor Position X LSB
0D	R/W	Cursor Position X MSB
0E	R/W	Cursor Position Y LSB
0F	R/W	Cursor Position Y MSB

Table 5.1 Direct Register Map

Index (hex)	Mode	
00 - 05		Reserved
06	R/W	Cursor Control Register (CR)
07 - 17		Reserved
18	R/W	Color Mode Register (CMR)
19	R/W	Mode Control Register (MDCR)
1A - 1B		Reserved
1C	R/W	Palette Page Register
1D		Reserved
1E	R/W	Miscellaneous Control Register (MCR)
1F		Reserved
20	W	Pixel Clock Program Register A1
21	W	Pixel Clock Program Register A2
22	W	Pixel Clock Program Register A3
23	W	Pixel Clock Program Register B1
24	W	Pixel Clock Program Register B2
25	W	Pixel Clock Program Register B3
26	W	Pixel Clock Program Register C1
27	W	Pixel Clock Program Register C2
28	W	Pixel Clock Program Register C3
29	R	Pixel Clock Status Register
2A - 2F		Reserved
30	W	Memory Clock Program Register 1
31	W	Memory Clock Program Register 2
32	W	Memory Clock Program Register 3
33	R	Memory Clock Status Register
34 - 3F		Reserved
40	R/W	Color Key Control Register
41	R/W	Color Key Overlay Value
42	R/W	Color Key Red Value
43	R/W	Color Key Green Value
44	R/W	Color Key Blue Value
45 - FF	R/W	Reserved

Table 5.2 Indirect Register Map

5.5 Color Palette RAM

The color palette RAM is addressed by an internal 8-bit address register for writing and reading the RAM. This register is automatically incremented following a RAM transfer, allowing the entire palette to be accessed with one write to the address register. When the address register increments beyond the last location in the RAM it is reset to the first location.

The color palette RAM is 8 bits wide for each color component even when 6-bit mode is chosen. If 6-bit mode is chosen and the color data is written into the palette, the 6 LSB bits will be shifted to the 6 MSB positions and the 2 LSBs filled with 0's. In addition, if they are read back in the 6 bit mode, the 6 MSB bits will be shifted to the 6 LSB positions and the 2 MSBs filled with 0's.

To load the color palette, the CPU first writes to the palette RAM write address register (Direct register: 00h) with the index of the first entry to be modified. The selected palette RAM location is loaded a byte at a time by writing a sequence of three bytes (red, green and blue) to the palette RAM data register (Direct register: 01h). After the blue write cycle, the palette RAM address register increments to the next location.

To read from the color palette, the CPU first writes to the palette read address register (Direct register: 03h) with the index of the entry to be read. Three successive reads from the palette RAM data register supplies red, green, and blue color data for the specified location. Following the blue read cycle, the address register is incremented.

The read-mask register (Direct register: 02h) is an 8-bit register used to enable or disable a bit plane from addressing the color-palette RAM in the SVGA mode. Each palette address bit is logically ANDed with the corresponding bit from the read-mask register before going to the palette page register and addressing the palette RAM

5.6 Cursor Color Registers

The registers for the three cursor colors are accessed through the direct register map. See table below for the use of the cursor colors.

The cursor color address register (Direct register: 04h) must be initialized before writing to the cursor color data register. The lower two bits of this register select one of the four cursor color registers according to the table below. The selected 24-bit cursor color register is loaded a byte at a time by writing a sequence of three bytes (red, green and blue) to the cursor color data register (Direct register: 05h). After the blue byte is written, the cursor color address register increments to the next color. All three colors (plus the unused color) may be loaded with a single write to the cursor color address followed by 12 consecutive writes to the cursor color data register.

Reading is handled in a similar manner by first loading the address register, then reading from the data register. After the blue byte is read, the cursor color address register is incremented to the next color. All four colors may be read with a single write to the cursor color address register followed by 12 consecutive reads of the cursor color data register.

Bit 1	Bit 0	REGISTER
0	0	Unused color
0	1	Cursor color 0
1	0	Cursor color 1
1	1	Cursor color 2

Table 5.3 Cursor Color Register Address Format

5.6.1 Three-Color Cursor

Cursor Control Register, bits 1 and 0 specify whether the XGA mode, or X-Windows mode, or 3-color mode, is used to interpret the cursor information stored in planes 0 and 1 of the cursor RAM. The relationship of cursor color modes and cursor RAM contents to the cursor appearance is shown in the table below.

Cursor RAM				
Plane 1	Plane 0	3 Color	XGA Mode	X Windows Mode
0	0	Transparent	Cursor color 0	Transparent
0	1	Cursor color 0	Cursor color 1	Transparent
1	0	Cursor color 1	Transparent	Cursor color 0
1	1	Cursor color 2	Complement	Cursor color 1

Table 5.4 Cursor Color Selection Modes

Notes: 1. Cursor color 0, 1, and 2 are set by writing to the cursor color registers.
2. Transparent: The underlying pixel color is displayed.
3. Complement: The ones complement of the underlying pixel color is displayed.

5.7 Cursor RAM

The cursor RAM can be written to using the following procedure. First, Cursor Control Register (CR) bits 3 and 2 are set to the top two bits of the RAM address. Next, Cursor RAM Write Address Register (Direct register 00h) is loaded with the bottom eight bits of the address. Then the cursor data can be written to the Cursor RAM Data Register (Direct register 0Bh). Each time the Cursor RAM Data Register is written, the Cursor RAM Write Address Register is automatically incremented. The increment is also carried through to CR[3:4] so that the whole cursor RAM can be written to with one write to the Cursor RAM Write Address Register followed by multiple writes to the Cursor RAM Data Register.

The cursor RAM can be read from using the same procedure, but reading from the data register instead of writing to it.

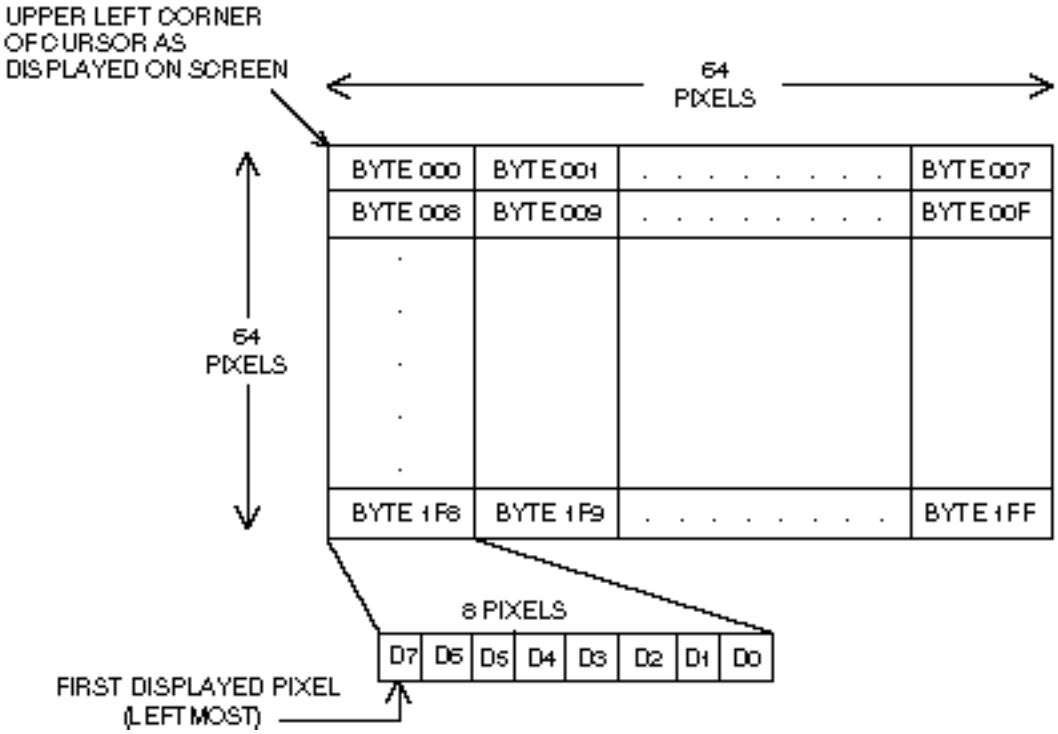


Figure 5.2 Cursor Plane 0

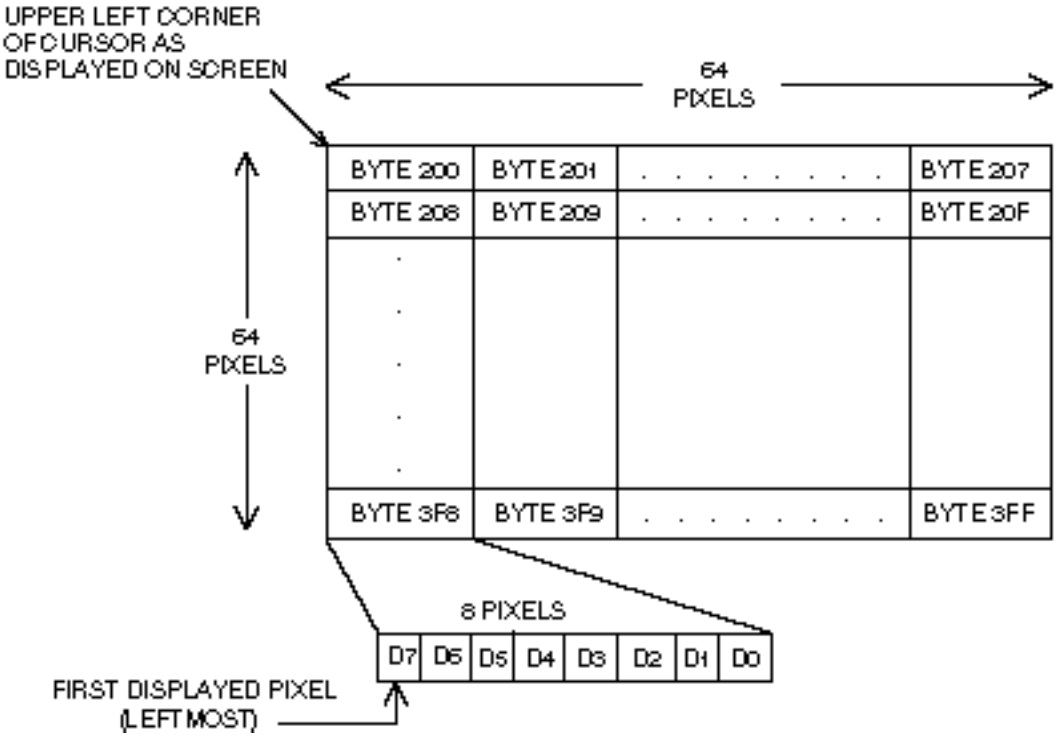


Figure 5.3 Cursor Plane 1

The cursor may be configured to hold four 32x32 cursors, in which case the cursor RAM is divided as follows:

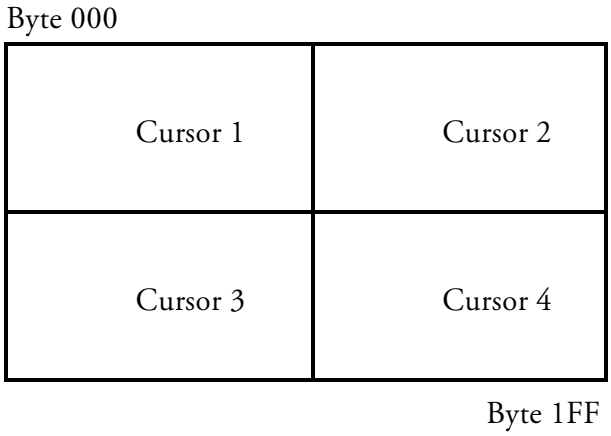


Figure 5.4 Cursor RAM Division Plane 0

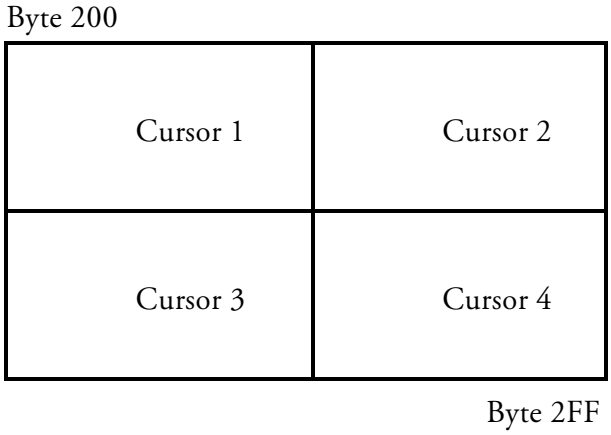


Figure 5.5 Cursor RAM Division Plane 1

5.8 Cursor Positioning

The cursor position (x,y) registers are used to position the cursor on the display screen. The cursor position (x,y) registers specify the location of the bottom right corner on the display screen relative to the end of the internal blank signal. The diagram below shows the orientation of the x,y coordinates for positioning the cursor.

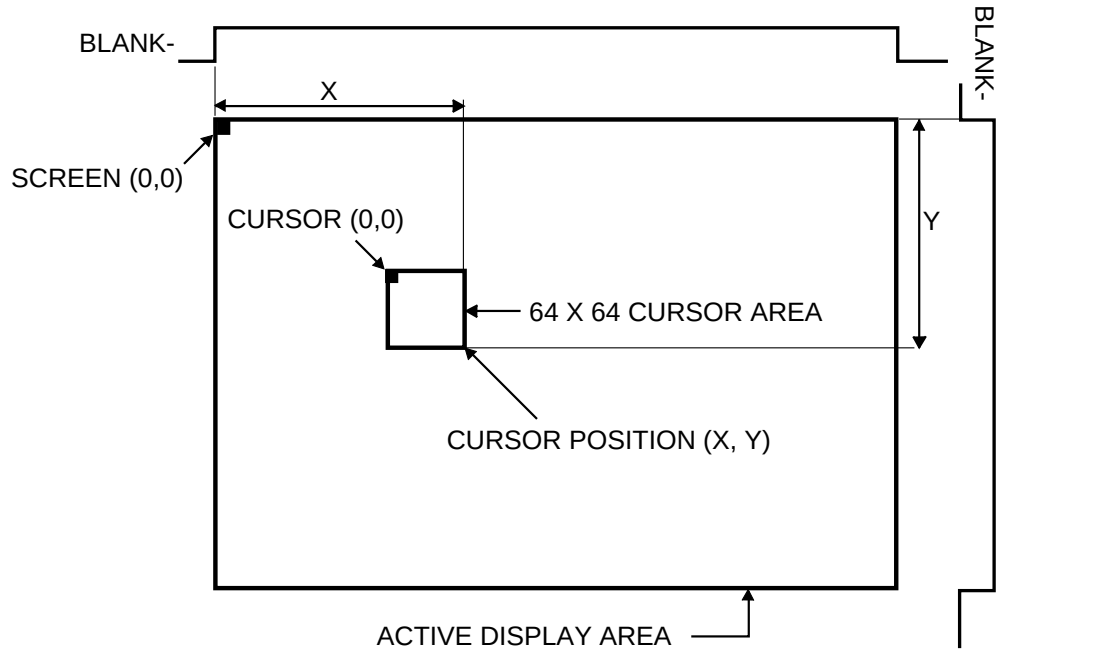


Figure 5.6 Cursor-Positioning

5.9 PLL Programming

Detailed PLL programming information for PERMEDIA can be found in 3Dlabs Application Note, 147.9.1 PLL Programming.

5.9.1 DClk Programming

DClk (or Dot Clock) is used to control the operation of the video output from the RAMDAC, and must be set to a frequency suitable for the display resolution and refresh rate. There are three sets of registers used to control its frequency: A, B, and C. Only one of them can be selected at a time. The selection is controlled by the VClkCtl register (bits 1 and 0) located at offset 0000.0040h of Region Zero.

Note: There is a maximum DClk frequency defined in section 8.3.1.

The output frequency is defined by:

DClk frequency = M[7:0] / N[7:0] / (2^{P}[2:0]) * Frequency of reference clock

M, N, and P are fields in the PLL control registers described in section 3. The power-on default values program the PLL as follows assuming the external crystal frequency is 14.31818MHz:

Register A	-	25.06 MHz
Register B	-	28.64 MHz
Register C	-	PLL Off

5.9.2 MClk Programming

The MClk (or Memory Clock) is used to control the operation of the graphics processor and the SGRAM memory. It is usually set to the highest frequency compatible with the timings of the memory parts fitted

Note: There is a maximum MClk frequency defined in section 8.3.1.

The output frequency is defined by:

$$\text{MClk frequency} = M[7:0] / N[7:0] / (2^{P}[2:0]) * \text{Frequency of reference clock}$$

The power-on default frequency is 50.11 MHz with a 14.31818MHz crystal.

6. Video Streams Unit

The Video Streams Unit transfers digital video data to and from the local memory. There are two separate streams, Stream A for input and Stream B for output. The unit also supports a General Purpose parallel bus, a serial bus, and an external ROM. These functions share the same pins, so only certain combinations are possible.

Mode	Video Stream A	Video Stream B	GPBus	ROM	Notes
0	Disabled	Disabled	Disabled	Enabled	ROM access
1	Enabled	Disabled	Enabled	Disabled	MPEG data to decoder via GP bus, decoded video into input port.
2	Disabled	Enabled	Disabled	Disabled	Wide output 16 bit.
3	Enabled	Enabled	Disabled	Disabled	Simultaneous input and output, program decoder and encoder through I2C.
4	Enabled	Disabled	Disabled	Disabled	Wide input 16 bit.
5..7	Disabled	Disabled	Disabled	Enabled	Default to mode 0.

Table 6.1 Possible Function Combinations

The pins change their meaning depending on the mode that the unit is in.

Pin Name	Mode0	Mode1	Mode2	Mode3	Mode4
VSAData[7]	ROMAddr[15]	VSAData[7]	VSBData[15]	VSAData[7]	VSAData[7]
VSAData[6]	ROMAddr[14]	VSAData[6]	VSBData[14]	VSAData[6]	VSAData[6]
VSAData[5]	ROMAddr[13]	VSAData[5]	VSBData[13]	VSAData[5]	VSAData[5]
VSAData[4]	ROMAddr[12]	VSAData[4]	VSBData[12]	VSAData[4]	VSAData[4]
VSAData[3]	ROMAddr[11]	VSAData[3]	VSBData[11]	VSAData[3]	VSAData[3]
VSAData[2]	ROMAddr[10]	VSAData[2]	VSBData[10]	VSAData[2]	VSAData[2]
VSAData[1]	ROMAddr[9]	VSAData[1]	VSBData[9]	VSAData[1]	VSAData[1]
VSAData[0]	ROMAddr[8]	VSAData[0]	VSBData[8]	VSAData[0]	VSAData[0]
VSBData[7]	ROMData[7]	GPData[7]	VSBData[7]	VSBData[7]	VSAData[15]
VSBData[6]	ROMData[6]	GPData[6]	VSBData[6]	VSBData[6]	VSAData[14]
VSBData[5]	ROMData[5]	GPData[5]	VSBData[5]	VSBData[5]	VSAData[13]
VSBData[4]	ROMData[4]	GPData[4]	VSBData[4]	VSBData[4]	VSAData[12]
VSBData[3]	ROMData[3]	GPData[3]	VSBData[3]	VSBData[3]	VSAData[11]
VSBData[2]	ROMData[2]	GPData[2]	VSBData[2]	VSBData[2]	VSAData[10]
VSBData[1]	ROMData[1]	GPData[1]	VSBData[1]	VSBData[1]	VSAData[9]
VSBData[0]	ROMData[0]	GPData[0]	VSBData[0]	VSBData[0]	VSAData[8]
VSCtl[7]	ROMAddr[7]	VSAHRef	VSAHRef	VSAHRef	VSAHRef
VSCtl[6]	ROMAddr[6]	VSAVRef	VSAVRef	VSAVRef	VSAVRef
VSCtl[5]	ROMAddr[5]	VSAVActive	VSAVActive	VSAVActive	VSAVActive
VSCtl[4]	ROMAddr[4]	GPInt	VSBHRef	VSBHRef	VSBHRef
VSCtl[3]	ROMAddr[3]	GPAddr[3]	VSBVRef	VSBVRef	VSBVRef

VSCtl[2]	ROMAddr[2]	GPAddr[2]	VSBVActive	VSBVActive	VSBVActive
VSCtl[1]	ROMAddr[1]	GPAddr[1]	VSAField	VSAField	VSAField
VSCtl[0]	ROMAddr[0]	GPAddr[0]	VSBField	VSBField	VSBField
VSAClk	VSAClk	VSAClk	VSAClk	VSAClk	VSAClk
VSBClk	VSBClk	VSBClk	VSBClk	VSBClk	VSBClk
VSAResetN	VSAReset	VSAReset	VSAReset	VSAReset	VSAReset
VSBResetN	VSBReset	VSBReset	VSBReset	VSBReset	VSBReset
VSGPChipSelectN	GPChipSelect	GPChipSelect	GPChipSelect	GPChipSelect	GPChipSelect
VSGPDataStrobeN	GPDataStrobe	GPDataStrobe	GPDataStrobe	GPDataStrobe	GPDataStrobe
VSGPReadWriteN	GPReadWrite	GPReadWrite	GPReadWrite	GPReadWrite	GPReadWrite
VSGPDataAckN	GPDataAck	GPDataAck	GPDataAck	GPDataAck	GPDataAck
ROMSelectN	ROMSelect	ROMSelect	ROMSelect	ROMSelect	ROMSelect
ROMWEN	ROMWE	ROMWE	ROMWE	ROMWE	ROMWE
SBClk	SBClk	SBClk	SBClk	SBClk	SBClk
SBData	SBData	SBData	SBData	SBData	SBData

Table 6.2 Pin Mode Name

The different modes are controlled through the VSConfiguration register Unit Mode field. The ROM mode is also enabled by an access to the ROM address space and overrides the current setting; when the mode is changed like this it does not revert to its original state after the ROM access completes. When either video stream A or video stream B is disabled, the corresponding reset is asserted.

Stream A and Stream B use externally generated horizontal and vertical timing signals to control the transfer of data. The video streams unit is always a slave and does not generate it's own timing controls or clocks.

6.1 Stream A

Video Stream A (VSA) is for input only. It accepts YUV422 data, down sizes as required, and writes it to memory. Downsizing is always by a power of 2 (1:1, 1:2, 1:4, 1:8), with independent control in X and Y. Scaling in X includes an averaging filter, scaling in Y discards data. There is also independent control for mirroring in X and Y. Input data may in the order YUYV or UYVY.

The video input is double or triple buffered through 2 or 3 address registers. The index which selects the address register to use is automatically updated after every field (or frame if the data is not interlaced or the fields are being combined). Synchronization is handled by comparing the automatically generated index against the value set in the VSAVideoAddressHost register. This is controlled by the CPU, and the value in it represents the buffer being read from or written to by the CPU. The corresponding register VSAVideoAddressIndex shows which buffer the VSA interface is using.

The VSA interface increments the index after each frame, but will not overtake the host register, so it will not increment to become equal to the register. The host register is reset to 2 and VSA register to zero. There are two index registers, one for video and the other for VBI data. Alternatively, the VSA unit can be locked to the VSB unit for playing video

directly without host intervention. This is done by setting the LockToStreamB bit in the VSAControl register.

The input video control is a slave to the video source. The HRef, VRef, and VActive signals are generated by the source and used to control the operation of the interface. The VRef signal is used to mark the start of a frame, which involves moving to the next base address and resetting the line counter. HRef marks the start of a line, which moving the address generator to the next line.

Vertical Blank Interval (VBI) data may be extracted and stored to a different part of memory. The current line count is compared against start and end values for the VBI data, and if the test passes the video stream is treated as VBI data which will not be scaled or mirrored.

Both VBI and video data are restricted to specified active regions of the field. The active regions are defined by a set of registers. The video data is also qualified by the VActive signal; to input data the video must be within the active region and have VActive asserted. VBI data may optionally use VActive to condition its input. If the VActive signal is not used, it should be tied permanently active.

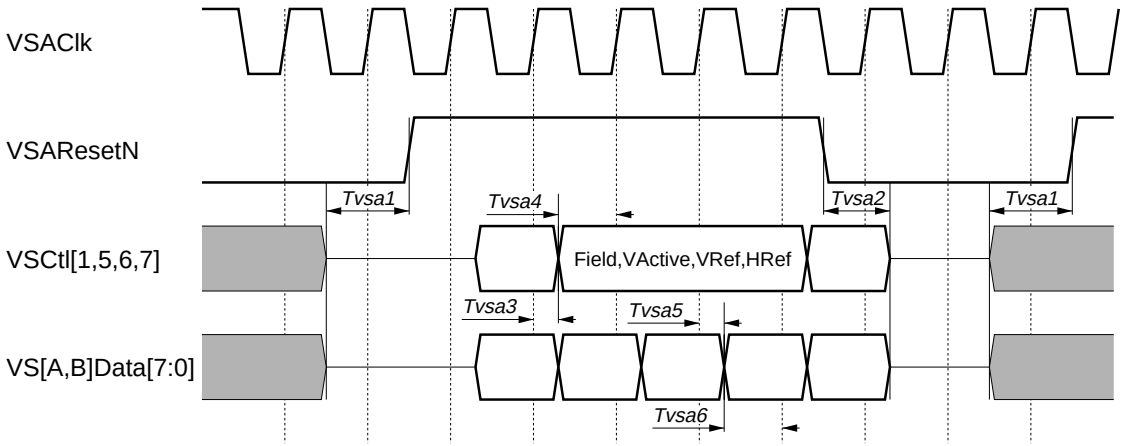


Figure 6.1 Video Stream A Timing

Tvsa1	Delay VSCtl[1,5,6,7],VSAData, VSBData driven to VSAResetN deasserted
Tvsa2	Delay VSAResetN asserted to VSCtl[1,5,6,7],VSAData, VSBData high-Z
Tvsa3	Input hold time for VSCtl[1,5,6,7]
Tvsa4	Input setup time for VSCtl[1,5,6,7]
Tvsa5	Input hold time for VSAData, VSBData
Tvsa6	Input setup time for VSAData, VSBData

Table 6.3 Video Stream A Timing

6.1.1 Programming Stream A

The way the Stream A is programmed depends on the type of external hardware supplying the data, and the format of the data. The first step in programming the unit is to set the hardware specific controls in the VSConfiguration register.

The polarity of the timing control signals should be set. If the field signal is available it should be enabled; if there is no field signal and the video is interlaced the VRef timing is used to distinguish between odd and even fields. There is a separate control that specifies that the VRef timing has correct interlace timings (InterlaceA). If the VBI data is to be gated by the VActive signal the bit VActiveVBIA should be enabled. Finally, if the order of the data is UYVY (instead of YUYV) the ReverseData bit should be enabled.

When VSConfiguration has been set, VSAControl should be set. In particular, the video and VBI data should enabled as required; if either of these are not enabled, the corresponding data is discarded. The BufferCtl bit specifies how many buffers will be available for stream. Double buffering saves memory, but triple buffering is required if the input video needs to be genlocked to the monitor refresh rate. Note that single buffering is possible by setting all the address registers to the same value.

Scaling and mirroring of the data may be enabled, and there is also the option to discard either of the fields in an interlaced frame. If the video is interlaced then two fields will be combined into one if the CombineFields bit is set. In this situation the unit will move to the next buffer after two fields have been received. If this bit is not set the unit will move to the next buffer after every field.

There are a set of registers which define the size and layout of the data in memory. These consist of three address registers and a stride which specifies how to move through memory between scanlines. A further set of registers defines the valid period of the data stream. Separate registers are available for video and VBI data. When allocating buffers in memory care should be taken to make sure they are large enough. If the incoming video is interlaced and being combined enough memory for a full frame must be allocated; if interlaced video is not combined each buffer only needs half the space.

The status register holds information about the state of Stream A. If the unit has been configured to expect interlace video and it receives an invalid sequence of fields it will set the InvalidInterlace bit in the VSStatus register and optionally generate an interrupt. The status register also reports the type of the last three fields so that the current field type can be detected and the cause of an invalid status determined.

The status register may also report a FIFO overflow, which may also result in an error interrupt. The FIFO parameters are set by the VSAFifoControl register, and should be adjusted for optimum performance. If the setting is incorrect and does not allow a sufficiently high bandwidth into local memory the FIFO may overflow and the error reported.

6.2 Stream B

Video Stream B (VSB) is for output only. It outputs YUV422 or RGB data, and can convert from RGB to YUV with gamma correction. Data may be either YUYV or UYVY

(or RGB or BGR). The RGB data is 16 bits, with 5 bits of red and blue, and 6 bits of green. The data read from memory may be in any of the color formats supported by the graphics core.

The video output is double or triple buffered through 2 or 3 address registers. The index which selects the address register to use is automatically updated after every field (or frame if the data is not interlaced or the fields are being combined). Synchronization is handled by comparing the automatically generated index against the value set in the VSBVideoAddressHost register. This is controlled by the CPU, and the value in it represents the buffer being read from or written to by the CPU. The corresponding register VSBVideoAddressIndex shows which buffer the VSB interface is using.

The VSB interface increments the index after each frame, but will not overtake the host register, so it will not increment to become equal to the register. The host register is reset to 0 and VSA register to two. There are two index registers, one for video and the other for VBI data. Alternatively, the VSB unit can be locked to the VSA unit for playing video directly without host intervention. This is done by setting the LockToStreamA bit in the VSBControl register.

The input video control is a slave to the video source. The HRef, VRef, and VActive signals are generated by the source and used to control the operation of the interface. The VRef signal is used to mark the start of a frame, which involves moving to the next base address and resetting the line counter. HRef marks the start of a line, which moving the address generator to the next line.

Vertical Blank Interval (VBI) data may be taken from a different part of memory. The current line count is compared against start and end values for the VBI data, and if the test passes the video stream is treated as VBI data which will not be formatted or gamma corrected or converted to YUV.

Both VBI and video data are restricted to specified active regions of the field. The active regions are defined by a set of registers. The video data is also qualified by the VActive signal; to input data the video must be within the active region and have VActive asserted. VBI data may optionally use VActive to condition its input. If the VActive signal is not used, it should be tied permanently active.

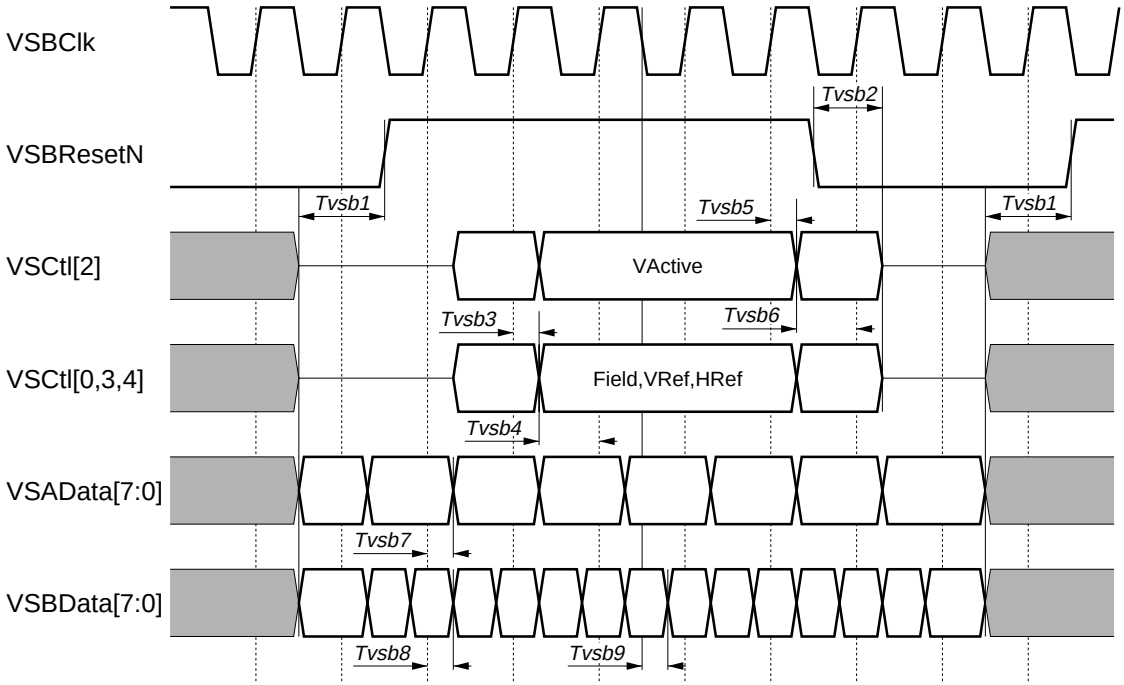


Figure 6.2 Video Stream B Timing

Tvsb1	Delay VSCtl[0,2,3,4] driven to VSBResetN deasserted
Tvsb2	Delay VSBResetN asserted to VSCtl[0,2,3,4] high-Z
Tvsb3	Input hold time for VSCtl[0,3,4]
Tvsb4	Input setup time for VSCtl[0,3,4]
Tvsb5	Input hold time for VSCtl[2]
Tvsb6	Input setup time for VSCtl[2]
Tvsb7	Output valid time from rising VSBClk to VSAData
Tvsb8	Output valid time from rising VSBClk to VSBData
Tvsb9	Output valid time from falling VSBClk to VSBData

Table 6.4 Video Stream B Status

6.2.1 Programming Stream B

The way the Stream B is programmed depends on the type of external hardware using the data, and the format of the data. The first step in programming the unit is to set the hardware specific controls in the VSConfiguration register.

The polarity of the timing control signals should be set. If the field signal is available it should be enabled; if there is no field signal and the video is interlaced the VRef timing is used to distinguish between odd and even fields. There is a separate control that specifies that the VRef timing has correct interlace timings (InterlaceB). If the VBI data is to be gated by the VActive signal the bit VActiveVBIA should be enabled.

If the output data should be YUV422 the color space conversion should be enabled. If data is required to be transferred on both clock edges DoubleEdgeB should be enabled. Finally, if the order of the data is UYVY (instead of YUYV) the ReverseData bit should be enabled.

When VSConfiguration has been set, VSBControl should be set. In particular, the video and VBI data should enabled as required; if either of these are not enabled, the corresponding data is discarded. The BufferCtl bit specifies how many buffers will be available for stream. Double buffering saves memory, but triple buffering is required if the input video needs to be genlocked to the monitor refresh rate. Note that single buffering is possible by setting all the address registers to the same value. If CombineFields is set both fields of an interlaced frame are assumed to come from the same buffer, and both will be output before moving on to the next buffer.

The color format must be set to match the format of the data in the local memory.

			Internal Color Channels			
Color Format	RGB	Name	R/Y	G/U	B/V	A
0	0	8:8:8:8	8@0	8@8	8@16	8@24
1	0	5:5:5:1Front	5@0	5@5	5@10	1@15
2	0	4:4:4:4	4@0	4@4	4@8	4@12
5	0	3:3:2Front	3@0	3@3	2@6	0
6	0	3:3:2Back	3@8	3@11	2@14	0
9	0	2:3:2:1Front	2@0	3@2	2@5	1@7
10	0	2:3:2:1Back	2@8	3@10	2@13	1@15
11	0	2:3:2FrontOff	2@0	3@2	2@5	0
12	0	2:3:2BackOff	2@8	3@10	2@13	0
13	0	5:5:5:1Back	5@16	5@21	5@26	1@31
16	0	5:6:5Front	5@0	6@5	5@11	0
17	0	5:6:5Back	5@16	6@21	5@27	0
19	0	YUV422	8@0	8@8	8@8	0
0	1	8:8:8:8	8@16	8@8	8@0	8@24
1	1	5:5:5:1Front	5@10	5@5	5@0	1@15
2	1	4:4:4:4	4@8	4@4	4@0	4@12
5	1	3:3:2Front	3@5	3@2	2@0	0
6	1	3:3:2Back	3@13	3@10	2@8	0
9	1	2:3:2:1Front	2@5	3@2	2@0	1@7
10	1	2:3:2:1Back	2@13	3@10	2@8	1@15
11	1	2:3:2FrontOff	2@5	3@2	2@0	0
12	1	2:3:2BackOff	2@13	3@10	2@8	0
13	1	5:5:5:1Back	5@26	5@21	5@16	1@31
16	1	5:6:5Front	5@11	6@5	5@0	0
17	1	5:6:5Back	5@27	6@21	5@16	0
19	1	YUV422	8@8	8@0	8@0	0

Table 6.5 Component Bit Position Formats

The offset modes have 64 subtracted from them before the components are separated. The individual color components (R, G and B) are formed, optionally gamma corrected and converted to YUV.

There are a set of registers which define the size and layout of the data in memory. These consist of three address registers and a stride which specifies how to move through memory between scanlines. A further set of registers defines the valid period of the data stream.

Separate registers are available for video and VBI data. When allocating buffers in memory care should be taken to make sure they are large enough. If the incoming video is interlaced and being combined enough memory for a full frame must be allocated; if interlaced video is not combined each buffer only needs half the space.

The status register holds information about the state of Stream B. If the unit has been configured to expect interlace video and it receives an invalid sequence of fields it will set the InvalidInterlace bit in the VSStatus register and optionally generate an interrupt. The status register also reports the type of the last three fields so that the current field type can be detected and the cause of an invalid status determined.

The status register may also report a FIFO underflow, which may also result in an error interrupt. The FIFO parameters are set by the VSBFifoControl register, and should be adjusted for optimum performance. If the setting is incorrect and does not allow a sufficiently high bandwidth into local memory the FIFO may underflow and the error reported.

6.3 General Purpose Bus

The GPBus is available in mode 1. Two different bus protocols are supported, Mode A and Mode B. Which mode is used is set in the VSConfiguration register. The different modes have different timings and different signals.

External Name	Mode A Name	Mode B Name
VSBData[7..0]	GPData	GPData
VSCtl[3..0]	GPAddr	GPAddr
VSGPChipSelectN	GPChipSelect	GPChipSelect
VSGPDataStrobeN	GPDataStrobe	GPRead
VSGPReadWriteN	GPReadWrite	GPWrite
VSGPDataAckN	GPDataAck	GPReady

Table 6.6 Mode A and B Pin Sharing

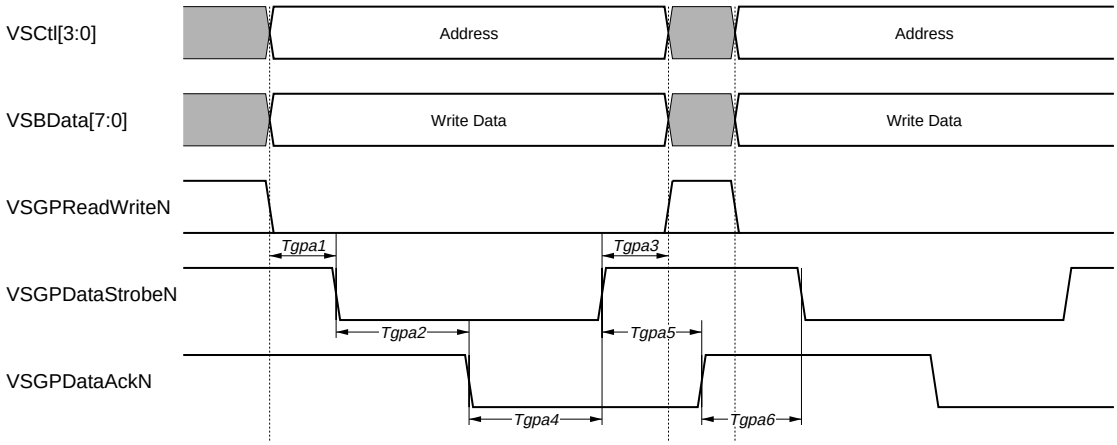


Figure 6.3 General Purpose Bus Mode A Write

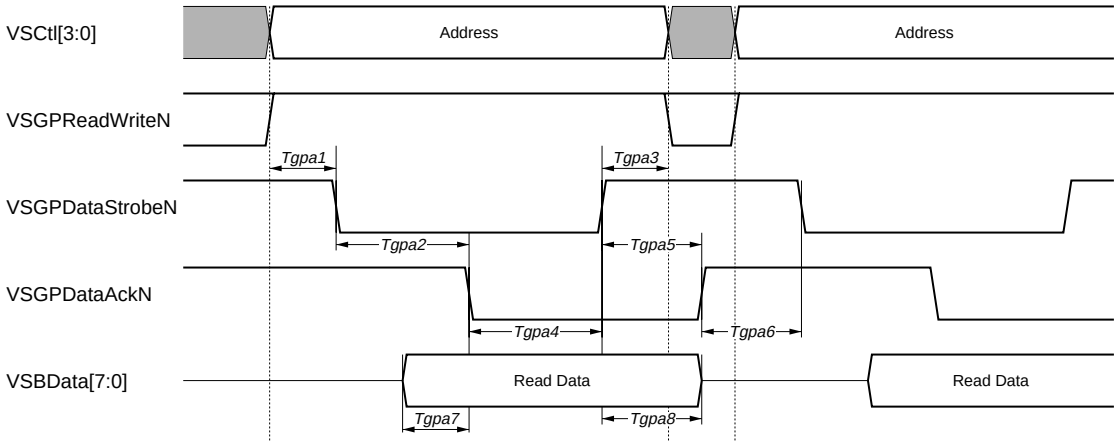


Figure 6.4 General Purpose Bus Mode A Read

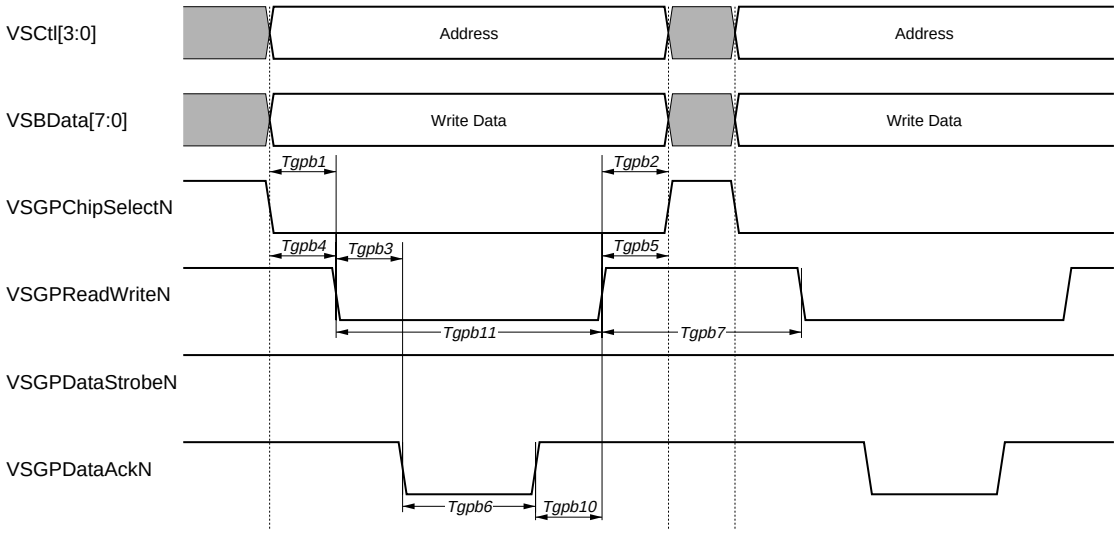


Figure 6.5 General Purpose Bus Mode B Write

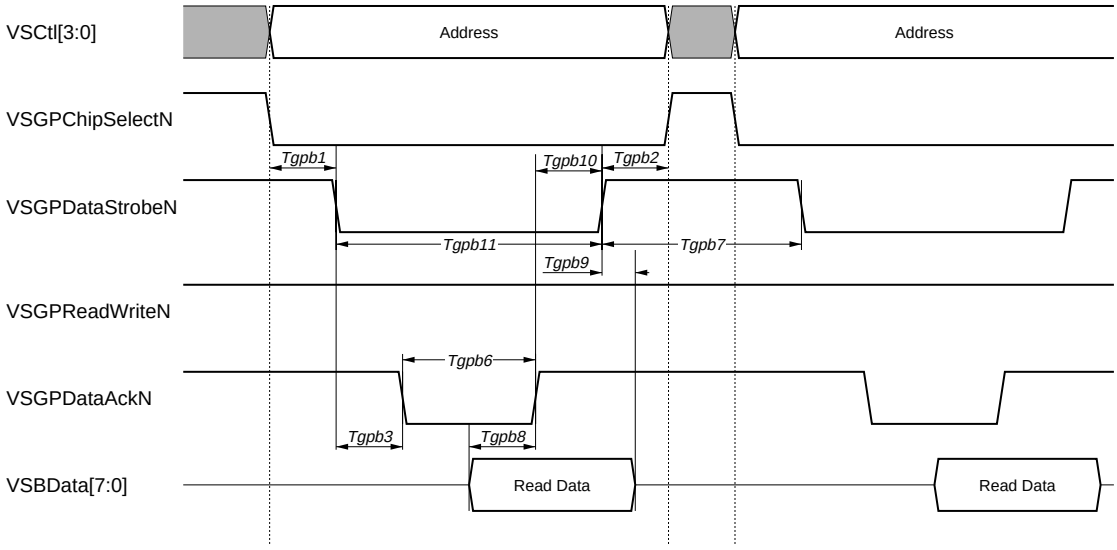


Figure 6.6 General Purpose Bus Mode B Read

Tgpa1	VSGPReadWriteN,VSCtl[3:0],VSBData setup to VSGPDataStrobeN asserted.
Tgpa2	Delay from VSGPDataStrobeN asserted to VSGPDataAckN asserted.
Tgpa3	VSGPReadWriteN,VSCtl[3:0],VSBData hold from VSGPDataStrobeN deasserted.
Tgpa4	Delay from VSGPDataAckN asserted to VSGPDataStrobeN deasserted.
Tgpa5	Delay from VSGPDataStrobeN deasserted to VSGPDataAckN deasserted.
Tgpa6	Delay from VSGPDataAckN deasserted to VSGPDataStrobeN asserted.
Tgpa7	Read data setup to VSGPDataAckN asserted.
Tgpa8	Read data hold from VSGPDataAckN deasserted.

Table 6.7 General Purpose Bus Mode Status

6.4 Serial Bus

The serial bus in the Video Streams unit follows the I2C bus protocol.

PERMEDIA has two serial bus interfaces, one in the video unit for connecting to a monitor, and one in the video streams unit for connecting to an external device such as a TV encoder or decoder. Both buses are controlled in the same way, with the I2C bus protocol being driven by the CPU. This section assumes an understanding of the I2C protocols.

In situations where PERMEDIA is the only master device on the I2C bus, only the DataIn, ClkIn, DataOut, and ClkOut bits in the registers need to be used. The data level is set by the DataOut bit, and the clock line strobed by the ClockOut bit. The ClockIn bit should be checked in case a slave device inserts wait states by holding the clock line low. Data returned from the slave is available from the DataIn bit.

If there are multiple masters on the bus a start condition can be detected and reported by an interrupt. When data is transferred it is latched in the registers as LatchedData, and the DataValid bit set. When receiving data from another master it is likely that wait states will need to be inserted. This is done by setting the Wait bit to enable which causes wait states to be inserted until the register is next read. If the register is read with the DataValid bit set, the data should be used as the bus will have been released to move onto the next transfer. When the bus passes through a stop condition, the Stop bit is set.

6.5 ROM

A ROM access takes over the video stream unit, forces Stream A and Stream B into reset, and disables the GP bus. When the access has completed the video streams remain in reset and must be explicitly released before they can be used.

The ROM is accessed immediately after reset to load chip configuration data which is passed directly to the PCI bus interface.

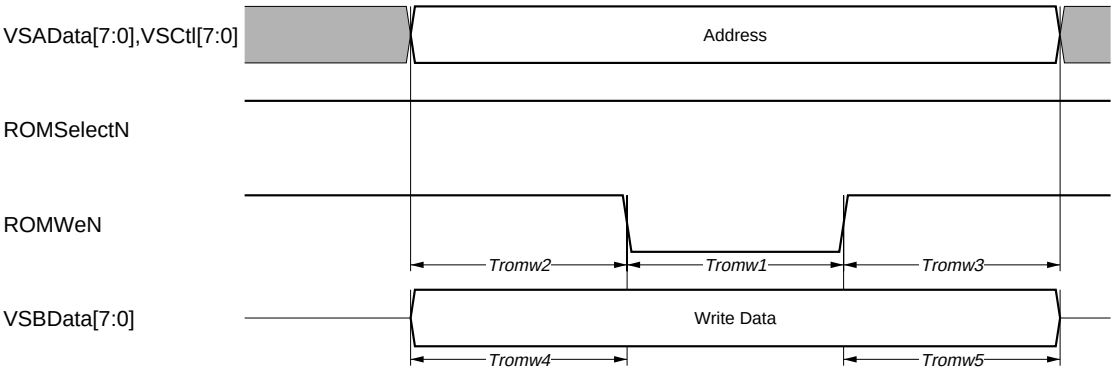


Figure 6.7 ROM Write

Tromw1	ROMWeN pulse width
Tromw2	Address setup to ROMWeN asserted
Tromw3	Address hold from ROMWeN deasserted
Tromw4	Write data setup to ROMWeN asserted
Tromw5	Write data hold from ROMWeN deasserted

Table 6.8 ROM Write Status

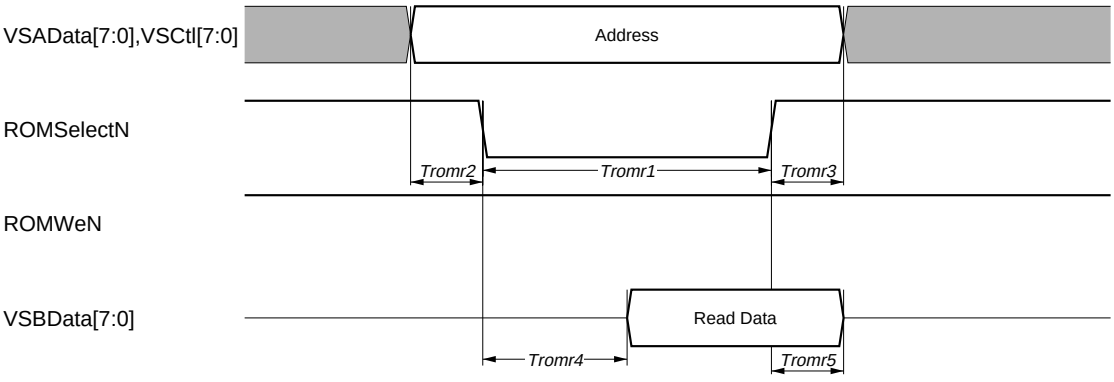


Figure 6.8 ROM Read

Tromr1	ROMSelectN pulse width
Tromr2	Address setup to ROMSelectN asserted
Tromr3	Address hold from ROMSelectN deasserted
Tromr4	Delay ROMSelectN asserted to read data valid
Tromr5	Delay ROMSelectN deasserted to read data high-Z

Table 6.9 RO M Read Status

7. Reset Control

7.1 Reset Control

At reset certain signal pins are read and the values present (due to pull-ups or pull-downs) are used to initialize bits of particular registers.

Name	Pin	Description
ShortReset	VSAData[0]	1 = use short reset
PCIMaxLat1	VSAData[1]	bit 7 of PCI max latency register
PCIMaxLat0	VSAData[2]	bit 6 of PCI max latency register
PCIMinGnt1	VSAData[3]	bit 7 of PCI min grant register
PCIMinGnt0	VSAData[4]	bit 6 of PCI min grant register
BaseClassZero	VSAData[5]	1 = base class is zero
VgaEnable	VSAData[6]	1 = SVGA present
VgaFixed	VSAData[7]	1 = enable SVGA fixed address
AGPCapable	VSBData[0]	1 = AGP Capable
SBACapable	VSBData[1]	1 = Sideband Address Capable
RetryDisable	VSBData[2]	1 = PCIRetrys disabled
SClkSel0	VSBData[3]	Clock source for Delta setup unit
SClkSel1	VSBData[4]	Clock source for Delta setup unit
SubsystemFromRom	VSBData[5]	1 = load subsystem data from at reset

The SClk select pins are encoded as follows:

SClk1	SClk 0	Frequency
0	0	PClk
0	1	PClk/2
1	0	MClk
1	1	MClk/2

Additionally there is a hard configuration pin:

Name	Pin	Description
PCIClk66	PCIClkSel	0 = upto 33MHz 1 = 66MHz

8. Electrical Characteristics

8.1 Absolute Maximum Ratings

Junction Temperature	125°C
Storage Temperature	-65°C to 150°C
VDD DC Supply Voltage	3.8V
VCC DC Supply Voltage	5V
I/O Pin Voltage with respect to GND	-0.5V to VDD + 0.3V
I/O Pin Voltage with respect to GND	-0.5V to VCC + 0.5V

8.2 DC Specifications

Symbol	Parameter	Min	Max	Unit
VDD	Supply Voltage	3.15	3.45	V
VCC	Supply Voltage	4.75	5.25	V
L_{PIN}	Pin Inductance		18.4	nH
I_{CC} (3V)	Power Supply Current		1	A
I_{CC} (5V)	Power Supply Current			mA

8.2.1 PCI Signal DC Specifications

Symbol	Parameter	Min	Max	Unit
V_{PIL}	Input Low Voltage		0.8	V
V_{PIH}	Input High Voltage	2.0		V
V_{POL}	Output Low Voltage		0.5	V
V_{POH}	Output High Voltage	2.4		V
I_{PIL}	Input Low Current		-20	uA
I_{PIH}	Input High Current		+20	uA
C_{PIN}	Input Capacitance		10	pF
C_{CLK}	PCI Clock Input Capacitance		10	pF
C_{IDSEL}	PCI Idsel Input Capacitance		8	pF

8.2.2 Non-PCI Signal DC Specifications

Symbol	Parameter	Min	Max	Unit
V_{IL}	Input Low Voltage		0.8	V
V_{IH}	Input High Voltage	2.0		V
V_{OL}	Output Low Voltage		0.5	V
V_{OH}	Output High Voltage	2.4		V
I_{IL}	Input Low Current		+10	uA
I_{IH}	Input High Current		-10	uA
I_{IHPD}	Pulldown Input High Current		250	uA
I_{ILPU}	Pullup Input Low Current		250	uA
C_{IN}	Input Capacitance		10	pF

8.3 AC Specifications

Pin Name	Capacitive Load
MADD[9:0].	80pF
PCIAD[31:0], PCICBEN[3:0], PCIPar, PCIFrameN,PCIIRdyN, PCITRdyN, PCIStopN, PCIIdsel, PCIDevselN, PCIReqN, PCIGntN, PCIIntAN ,AGPPipeN, AGPRbfN, AGPSBA[7:0],	50pF in PCI 33 system 10pF in AGP system
MBANK[3:0], MBYTE[7:0], MCAS[1:0], MDSF[1:0], MEMCKE, MEMCKOUT[1:0], MRAS[1:0], MWE[1:0]. VidDDCClk, VidDDCData, VidRightEye, VidHSYNC, VidVSYNC,VSAResetN,VSBResetN	50pF
MDAT[63:0].	40pF
ROMSelectN, ROMWEN, SBClk, SBData, VSAData[7:0], VSBData[7:0], VSCtl[7:0], VSGPChipSelectN, VSGPDataAckN, VSGPDataStrobeN, VSGPReadWriteN.	30pF

8.3.1 Clock Timing

Symbol	Parameter	Min	Max	Units	Notes
T_{PCyc}	PCIClk Cycle Time	15	-	ns	
T_{PHigh}	PCIClk High Time	-	-	ns	
T_{SLow}	PCIClk Low Time	-	-	ns	
T_{MCyc}	MClk Cycle Time	12	-	ns	
T_{MHigh}	MClk High Time	-	-	ns	
T_{MLow}	MClk Low Time	-	-	ns	
T_{SCyc}	SClkin Cycle Time	24	-	ns	
T_{SHigh}	SClkin High Time	8	-	ns	
T_{SLow}	SClkin Low Time	8	-	ns	
T_{DCyc}	DClk Cycle Time	4.3	-	ns	
T_{DHigh}	DClk High Time	-	-	ns	
T_{DLow}	DClk Low Time	-	-	ns	

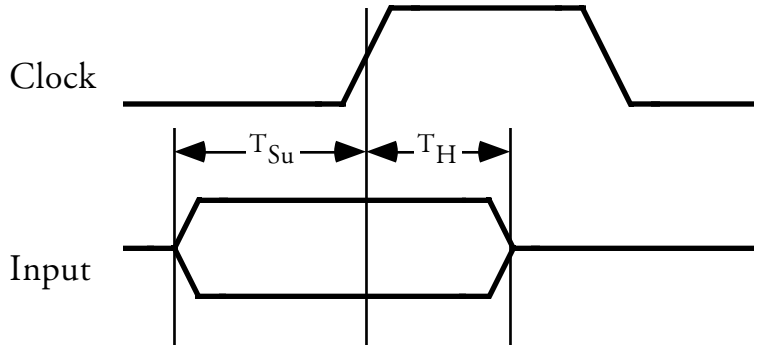


Figure 8.1 Input Timing Parameters

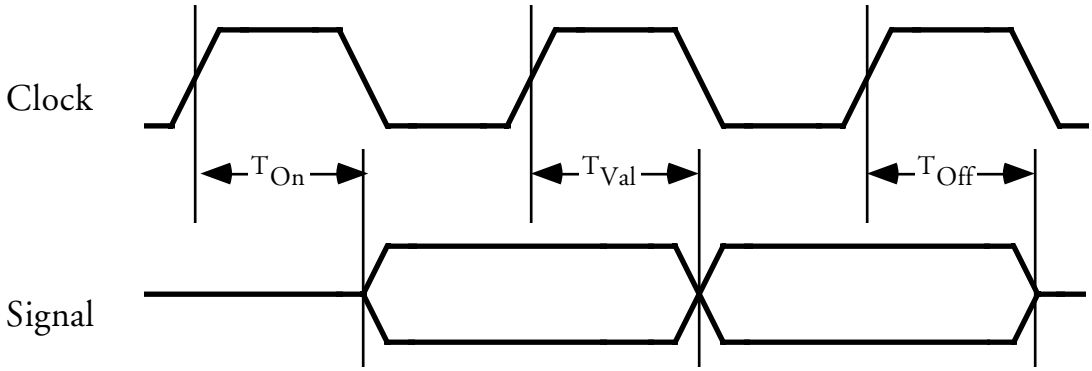


Figure 8.2 Output Timing Parameters

8.3.2 PCI Clock Referenced Input Timing

Parameter	T_{Su} Min	T_H Min	Units	Notes
PCIAD(31:0), PCICBEN(3:0), PCIPar, PCIFrameN, PCIIRdyN, PCITRdyN, PCIStopN, PCIIdsel, PCIDevselN, AGPSt0-2	5	0	ns	
PCIGntN	5	0	ns	
PCIRstN	7	0	ns	1

Note 1: PCIRstN is resynchronised internally. The timings given, when met, ensure that the reset is detected in the current cycle.

8.3.3 PCI -Referenced Output Timing

	T_{Val}		T_{On}		T_{Off}			
Parameter	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**	**Notes**
PCIAD(31:0), PCICBEN(3:0), PCIPar, PCIFrameN, PCIIRdyN, PCITRdyN, PCIStopN, PCIIdsel, PCIDevselN	2	11	2	11	2	11	ns	
PCIReqN	2	12					ns	
PCIIntAN	2	11					ns	1

Note 1: Timings given are for falling edges of the open drain signal. Rise times are dependent on the external pull-up resistor.

8.3.4 AGP Referenced Output Timing

	T_{Val}		T_{On}		T_{Off}			
Parameter	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**	**Notes**
PCIAD(31:0), PCICBEN(3:0), PCIPar, PCIFrameN, PCIIRdyN, PCITRdyN, PCIStopN, PCIIdsel, PCIDevselN	1.5	6	1.5	6	1.0	14	ns	
PCIReqN	1.5	6					ns	
PCIIntAN	1.5	6					ns	1

Note 1: Timings given are for falling edges of the open drain signal. Rise times are dependent on the external pull-up resistor.

8.3.5 MEMCKOUT Referenced Input Timing

All timings below are with respect to MEMCKOUT, which is a delayed version of MClk.

Parameter	**T_{Su} Min**	**T_{H} Min**	**Units**	**Notes**
MDAT[63:0]	1	3	ns	

8.3.6 MEMCKOUT Referenced Output Timing

All timings below are with respect to MEMCKOUT, which is a delayed version of MClk.

	T_{Val}		T_{On}		T_{Off}			
Parameter	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**	**Notes**
All memory control, data and address lines		12.5					ns	

8.3.7 Video Stream A Timing

	Min	Max	Unit	Notes
Tvsa1	1		PClk	
Tvsa2		RomPulse	PClk	1
Tvsa3	2		ns	
Tvsa4	1		ns	
Tvsa5	2		ns	
Tvsa6	1		ns	

Note 1: RomPulse refers to field of VSConfiguration register.

8.3.8 Video Stream B Timing

	Min	Max	Unit	Notes
Tvsb1	1		PClk	
Tvsb2		RomPulse	PClk	1
Tvsb3	2		ns	
Tvsb4	1		ns	
Tvsb5	0		ns	
Tvsb6	7		ns	
Tvsb7	3	13	ns	
Tvsb8	3	10	ns	
Tvsb9	3	10.5	ns	

Note 1: RomPulse refers to field of VSConfiguration register.

8.3.9 General Purpose Bus A Timing

	Min	Max	Unit	Note
Tgpa1	1		PClk	
Tgpa2	0	13020	ns	
Tgpa3	1		PClk	
Tgpa4	60	103	ns	
Tgpa5	0	0	ns	
Tgpa6	1		PClk	
Tgpa7	0		ns	
Tgpa8	0		ns	

8.3.10 General Purpose Bus B Timing

	MIN	MAX	Unit	Notes
Tgpb1	1		PClk	
Tgpb2	1		PClk	
Tgpb3		30	ns	
Tgpb4	1		PClk	
Tgpb5	1		PClk	
Tgpb6	0		ns	
Tgpb7	45		ns	
Tgpb8	0		ns	
Tgpb9	0	3	PClk	
Tgpb10	60	103	ns	
Tgpb11	75		ns	

8.3.11 ROM Write Timing

	Nominal	Unit	Note
Tromw1	RomPulse + 1	PClk	1
Tromw2	RomPulse + 1	PClk	1
Tromw3	RomPulse + 1	PClk	1
Tromw4	RomPulse + 1	PClk	1
Tromw5	RomPulse + 1	PClk	1

Note 1: RomPulse refers to field of VSConfiguration register.

8.3.12 ROM Read Timing

	Min	Nominal	Max	Unit	Note
Tromr1		RomPulse + 1		PClk	1
Tromr2		RomPulse + 1		PClk	1
Tromr3		RomPulse + 1		PClk	1
Tromr4			RomPulse - 2	PClk	1
Tromr5	0		3	PClk	

Note 1: RomPulse refers to field of VSConfiguration register.

9. Pin Assignment

This section shows the pin assignment for PERMEDIA and lists the signal types.

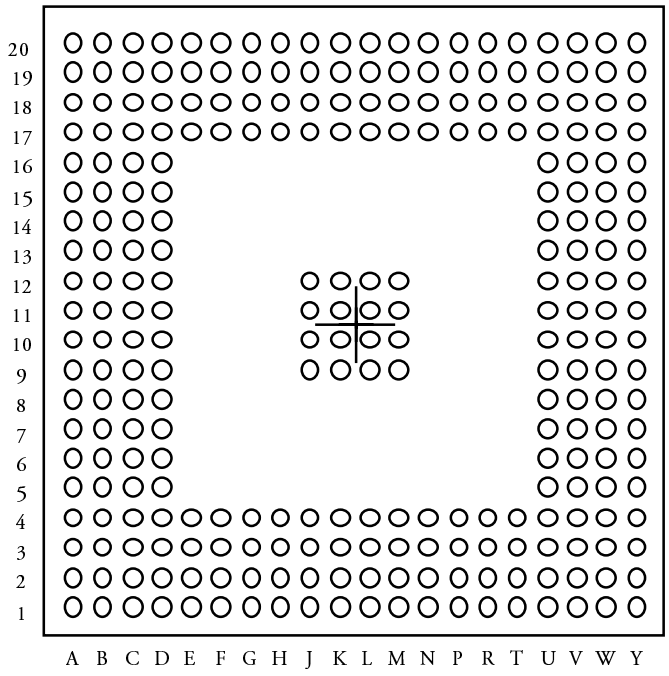


Figure 9.1 PERMEDIA Pin Numbering (from top)

9.1 Notes to Pin Listing

1. All VSS pins must be connected to ground, including the central pin grid H8:N13.
2. All VDD pins must be connected to 3.3V.
3. All VDDQ pins must be connect to VDDQ for AGP, or 3.3V for PCI.
4. VSTol should be connected to 5V.
5. All AGPTol pins should be connected to 3.3V for AGP, or Vio for PCI.
6. Unused pins should be left No Connect. However depending on the design some unused pins should be connected e.g. some AGP pins in a non-AGP design require pull ups. Refer to the relevant 3Dlabs PERMEDIA 2 Reference Board Design Schematics for further information.
7. Signal names ending in “N” are active low.

9.2 Pinlist by Name

The table below provides a brief description of each pin, and the following pin type definitions are used.

I = Input Signal

O = Output Signal

I/O = Bi-directional signal

OD = Open drain

Output power ratings are marked as 8 or 12 for the milliamp current rating, or P indicating a PCI compatible output.

Name		#	type	Description
AGPADSTB0	V	2	I	AGP AD 2X strobe
AGPADSTB1	J	1	I	AGP AD 2X strobe
AGPPipeN	D	3	O	AGP Pipelined address
AGPRbfN	C	2	O	AGP Read data buffer full
AGPSBA0	D	2	O	AGP Sideband Address 0
AGPSBA1	D	1	O	AGP Sideband Address 1
AGPSBA2	E	4	O	AGP Sideband Addr 2
AGPSBA3	E	3	O	AGP Sideband Addr 3
AGPSBA4	E	1	O	AGP Sideband Addr 4
AGPSBA5	F	3	O	AGP Sideband Addr 5
AGPSBA6	F	2	O	AGP Sideband Addr 6
AGPSBA7	G	4	O	AGP Sideband Addr 7
AGPSBSTB	E	2	O	AGP Sideband Addr 2X strobe
AGPSt0	C	4	I	AGP status 0
AGPSt1	C	3	I	AGP status 1
AGPSt2	D	5	I	AGP status 2
AGPTol1	F	1	I	AGP/PCI tolerance supply
AGPTol2	P	1	I	AGP/PCI tolerance supply
Dac AGND	B	12	I	DAC Power/Gnd pin
DacComp	A	14	I/O	Compensation pin
DacComp2	B	14	I/O	Compensation pin
Maddr0	V	16	O	Memory address line 0
Maddr1	W	16	O	Memory address line 1
Maddr2	Y	16	O	Memory address line 2
Maddr3	V	15	O	Memory address line 3
Maddr4	W	15	O	Memory address line 4
Maddr5	Y	15	O	Memory address line 5
Maddr6	V	14	O	Memory address line 6
Maddr7	W	14	O	Memory address line 7
Maddr8	Y	14	O	Memory address line 8
Maddr9	V	13	O	Memory address line 9
Mbank0	F	18	O	Memory bank select 0
Mbank1	F	19	O	Memory bank select 1
Mbank2	F	20	O	Memory bank select 2
Mbank3	G	17	O	Memory bank select 3
Mbyte0	G	18	O	Memory byte select 0

Name		#	type	Description
Mbyte1	G	19	O	Memory byte select 1
Mbyte2	G	20	O	Memory byte select 2
Mbyte3	H	18	O	Memory byte select 3
Mbyte4	H	19	O	Memory byte select 4
Mbyte5	H	20	O	Memory byte select 5
Mbyte6	J	17	O	Memory byte select 6
Mbyte7	J	18	O	Memory byte select 7
MCAS0	M	17	O	Memory CAS line 0
MCAS1	M	18	O	Memory CAS line 1
Mdat0	Y	5	I/O	Memory data line 0
Mdat1	Y	4	I/O	Memory data line 1
Mdat10	W	11	I/O	Memory data line 10
Mdat11	U	11	I/O	Memory data line 11
Mdat12	W	12	I/O	Memory data line 12
Mdat13	U	12	I/O	Memory data line 13
Mdat14	W	13	I/O	Memory data line 14
Mdat15	U	14	I/O	Memory data line 15
Mdat16	Y	17	I/O	Memory data line 16
Mdat17	V	17	I/O	Memory data line 17
Mdat18	W	18	I/O	Memory data line 18
Mdat19	Y	19	I/O	Memory data line 19
Mdat2	Y	6	I/O	Memory data line 2
Mdat20	W	20	I/O	Memory data line 20
Mdat21	V	19	I/O	Memory data line 21
Mdat22	U	19	I/O	Memory data line 22
Mdat23	T	19	I/O	Memory data line 23
Mdat24	T	17	I/O	Memory data line 24
Mdat25	R	18	I/O	Memory data line 25
Mdat26	R	20	I/O	Memory data line 26
Mdat27	P	20	I/O	Memory data line 27
Mdat28	P	18	I/O	Memory data line 28
Mdat29	N	18	I/O	Memory data line 29
Mdat3	V	6	I/O	Memory data line 3
Mdat30	N	20	I/O	Memory data line 30
Mdat31	M	20	I/O	Memory data line 31
Mdat32	W	5	I/O	Memory data line 32
Mdat33	V	5	I/O	Memory data line 33
Mdat34	W	6	I/O	Memory data line 34
Mdat35	Y	7	I/O	Memory data line 35
Mdat36	V	7	I/O	Memory data line 36
Mdat37	Y	8	I/O	Memory data line 37
Mdat38	V	8	I/O	Memory data line 38
Mdat39	Y	9	I/O	Memory data line 39
Mdat4	W	7	I/O	Memory data line 4
Mdat40	V	9	I/O	Memory data line 40
Mdat41	Y	10	I/O	Memory data line 41
Mdat42	V	10	I/O	Memory data line 42
Mdat43	Y	11	I/O	Memory data line 43
Mdat44	V	11	I/O	Memory data line 44

Name		#	type	Description
Mdat45	Y	12	I/O	Memory data line 45
Mdat46	V	12	I/O	Memory data line 46
Mdat47	Y	13	I/O	Memory data line 47
Mdat48	U	16	I/O	Memory data line 48
Mdat49	W	17	I/O	Memory data line 49
Mdat5	U	7	I/O	Memory data line 5
Mdat50	Y	18	I/O	Memory data line 50
Mdat51	Y	20	I/O	Memory data line 51
Mdat52	W	19	I/O	Memory data line 52
Mdat53	V	20	I/O	Memory data line 53
Mdat54	V	18	I/O	Memory data line 54
Mdat55	U	18	I/O	Memory data line 55
Mdat56	U	20	I/O	Memory data line 56
Mdat57	T	18	I/O	Memory data line 57
Mdat58	T	20	I/O	Memory data line 58
Mdat59	R	19	I/O	Memory data line 59
Mdat6	W	8	I/O	Memory data line 6
Mdat60	P	19	I/O	Memory data line 60
Mdat61	P	17	I/O	Memory data line 61
Mdat62	N	19	I/O	Memory data line 62
Mdat63	M	19	I/O	Memory data line 63
Mdat7	W	9	I/O	Memory data line 7
Mdat8	U	9	I/O	Memory data line 8
Mdat9	W	10	I/O	Memory data line 9
MDSF0	J	19	O	Memory DSF line 0
MDSF1	J	20	O	Memory DSF line 1
MemClkE	K	18	O	Memory clock enable
MemClkOut0	K	19	O	Memory clock 0
MemClkOut1	K	20	O	Memory clock 1
MonitorID2	C	14	I/O	Monitor ID 2
MRAS0	L	19	O	Memory RAS line 0
MRAS1	L	20	O	Memory RAS line 1
MWE0	K	17	O	Memory write enable 0
MWE1	L	18	O	Memory write enable 1
PCIAD0	Y	1	I/O	PCI address/data line 0
PCIAD1	Y	2	I/O	PCI address/data line 1
PCIAD10	U	2	I/O	PCI address/data line 10
PCIAD11	U	3	I/O	PCI address/data line 11
PCIAD12	T	1	I/O	PCI address/data line 12
PCIAD13	T	2	I/O	PCI address/data line 13
PCIAD14	T	3	I/O	PCI address/data line 14
PCIAD15	T	4	I/O	PCI address/data line 15
PCIAD16	N	3	I/O	PCI address/data line 16
PCIAD17	M	3	I/O	PCI address/data line 17
PCIAD18	M	2	I/O	PCI address/data line 18
PCIAD19	L	2	I/O	PCI address/data line 19
PCIAD2	Y	3	I/O	PCI address/data line 2
PCIAD20	M	4	I/O	PCI address/data line 20
PCIAD21	K	1	I/O	PCI address/data line 21

Name		#	type	Description
PCIAD22	L	4	I/O	PCI address/data line 22
PCIAD23	K	3	I/O	PCI address/data line 23
PCIAD24	J	2	I/O	PCI address/data line 24
PCIAD25	J	3	I/O	PCI address/data line 25
PCIAD26	J	4	I/O	PCI address/data line 26
PCIAD27	H	1	I/O	PCI address/data line 27
PCIAD28	H	2	I/O	PCI address/data line 28
PCIAD29	H	3	I/O	PCI address/data line 29
PCIAD3	W	1	I/O	PCI address/data line 3
PCIAD30	G	2	I/O	PCI address/data line 30
PCIAD31	G	3	I/O	PCI address/data line 31
PCIAD4	W	2	I/O	PCI address/data line 4
PCIAD5	W	3	I/O	PCI address/data line 5
PCIAD6	W	4	I/O	PCI address/data line 6
PCIAD7	V	1	I/O	PCI address/data line 7
PCIAD8	V	4	I/O	PCI address/data line 8
PCIAD9	U	1	I/O	PCI address/data line 9
PCICBEN0	V	3	I/O	PCI byte enable 0
PCICBEN1	R	2	I/O	PCI byte enable 1
PCICBEN2	M	1	I/O	PCI byte enable 2
PCICBEN3	K	2	I/O	PCI byte enable 3
PCIClk	B	4	I	PCI clock
PCIClkSel	A	3	I	33 / 66 MHz PCI select
PCIDevSelN	P	4	I/O	PCI device select
PCIFIFOInDis	A	4	O	Delta control
PCIFIFOOutDis	B	5	O	Delta control
PCIFrameN	N	1	I/O	PCI frame signal
PCIGntN	B	1	I	PCI grant signal
PCIIdSel	L	3	I	PCI ID select
PCIIntAN	A	2	OD	PCI interrupt
PCIPar	R	3	I/O	PCI parity
PCIRdyN	N	2	I/O	PCI ready
PCIReqN	B	2	O	PCI request
PCIRstN	B	3	I	PCI reset
PCIStopN	P	2	I/O	PCI stop
PCITRdyN	P	3	I/O	PCI T ready
PLL GND	B	10	I	PLL Power/Gnd pin
PLL Power	C	10	I	PLL Power/Gnd pin
PllDisable	C	11	I	
ROMSelectN	A	9	O	ROM select signal
ROMWeN	B	9	O	
SBClk	D	18	I/O	VideoStream B clock
SBData	D	10	I/O	serial bus data
SClkIn	C	6	I	Delta unit external clock
SclkOut	B	6	I	Delta clock out (from PCI clock)
TestMode	D	20	I	Testmode
VDD	D	6		
VDD	D	11		
VDD	D	15		

Name		#	type	Description
VDD	F	4		
VDD	F	17		
VDD	K	4		
VDD	L	17		
VDD	R	4		
VDD	R	17		
VDD	U	6		
VDD	U	10		
VDD	U	15		
VDDQ	C	1		
VDDQ	C	5		
VDDQ	G	1		
VDDQ	L	1		
VDDQ	R	1		
VDDQ	U	5		
VidBlue	A	13	O	Analogue blue signal
VidDDCClk	E	20	I/O	Clock line for DDC
VidDDCData	E	19	I/O	Data line for DDC
VidGreen	B	13	O	Analogue green signal
VidHSync	A	12	O	Horizontal sync
VidRed	C	13	O	Analogue red signal
VidResRef	C	12	I	Reference resistor
VidRightEye	E	18	O	Right signal for stereo
VidVRef	D	12	I	Voltage reference
VidVSync	B	11	O	Vertical sync
VSAClk	D	9	I	VideoStream A clock
VSAData0	A	6	I/O	VideoStream A data line 0
VSAData1	A	7	I/O	VideoStream A data line 1
VSAData2	B	7	I/O	VideoStream A data line 2
VSAData3	C	7	I/O	VideoStream A data line 3
VSAData4	D	7	I/O	VideoStream A data line 4
VSAData5	A	8	I/O	VideoStream A data line 5
VSAData6	B	8	I/O	VideoStream A data line 6
VSAData7	C	8	I/O	VideoStream A data line 7
VSAResetN	A	5	O	Video Stream reset
VSBClk	C	9	I	VideoStream B clock
VSBData0	B	18	I/O	VideoStream B data line 0
VSBData1	C	18	I/O	VideoStream B data line 1
VSBData2	A	19	I/O	VideoStream B data line 2
VSBData3	B	19	I/O	VideoStream B data line 3
VSBData4	C	19	I/O	VideoStream B data line 4
VSBData5	A	20	I/O	VideoStream B data line 5
VSBData6	B	20	I/O	VideoStream B data line 6
VSBData7	C	20	I/O	VideoStream B data line 7
VSBResetN	E	17	O	Video Stream B Reset Out
VSCtl0	D	14	O	VideoStreams Control line 0
VSCtl1	C	15	O	VideoStreams Control line 1
VSCtl2	B	15	O	VideoStreams Control line 2
VSCtl3	A	15	O	VideoStreams Control line 3

Name		#	type	Description
VSCtl4	D	16	O	VideoStreams Control line 4
VSCtl5	C	16	O	VideoStreams Control line 5
VSCtl6	B	16	O	VideoStreams Control line 6
VSCtl7	A	16	O	VideoStreams Control line 7
VSGPChipSelectN	C	17	O	VS GP bus chip select
VSGPDataAckN	A	18	I	VS GP bus data ack
VSGPDataStrobeN	B	17	O	VS GP bus data strobe
VSGPReadWriteN	A	17	O	VS GP bus read/write signal
VSS	A	1		
VSS	D	4		
VSS	D	8		
VSS	D	13		
VSS	D	17		
VSS	H	4		
VSS	H	17		
VSS	N	4		
VSS	N	17		
VSS	U	4		
VSS	U	8		
VSS	U	13		
VSS	U	17		
VSTol	D	19	I	VideoStreams tolerance supply
Xtal1	A	11	I	Crystal i/p 1
Xtal2	A	10	I	Crystal i/p 2

Table 9.1 Pinlist by Name

9.3 Pinlist by Number

Name		#	Description
VSS	A	1	
PCIIntAN	A	2	PCI interrupt
PCIClkSel	A	3	33 / 66 MHz PCI select
PCIFIFOInDis	A	4	Delta control
VSAResetN	A	5	Video Stream reset
VSAData0	A	6	VideoStream A data line 0
VSAData1	A	7	VideoStream A data line 1
VSAData5	A	8	VideoStream A data line 5
ROMSelectN	A	9	ROM select signal
Xtal2	A	10	Crystal i/p 2
Xtal1	A	11	Crystal i/p 1
VidHSync	A	12	Horizontal sync
VidBlue	A	13	Analogue red signal
DacComp	A	14	Compensation pin
VSCtl3	A	15	VideoStreams Control line 3
VSCtl7	A	16	VideoStreams Control line 7
VSGPReadWriteN	A	17	VS GP bus read/write signal
VSGPDataAckN	A	18	VS GP bus data ack
VSBData2	A	19	VideoStream B data line 2
VSBData5	A	20	VideoStream B data line 5
PCIGntN	B	1	PCI grant signal
PCIReqN	B	2	PCI request
PCIRstN	B	3	PCI reset
PCIClk	B	4	PCI clock
PCIFIFOOutDis	B	5	Delta control
SclkOut	B	6	Delta clock out (from PCI clock)
VSAData2	B	7	VideoStream A data line 2
VSAData6	B	8	VideoStream A data line 6
ROMWeN	B	9	
PLL GND	B	10	PLL Power/Gnd pin
VidVSync	B	11	Vertical sync
Dac AGND	B	12	DAC Power/Gnd pin
VidGreen	B	13	Analogue green signal
DacComp2	B	14	Compensation pin
VSCtl2	B	15	VideoStreams Control line 2
VSCtl6	B	16	VideoStreams Control line 6
VSGPDataStrobeN	B	17	VS GP bus data strobe
VSBData0	B	18	VideoStream B data line 0
VSBData3	B	19	VideoStream B data line 3
VSBData6	B	20	VideoStream B data line 6
VDDQ	C	1	
AGPRbfN	C	2	AGP Read data buffer full
AGPSt1	C	3	AGP status 1
AGPSt0	C	4	AGP status 0
VDDQ	C	5	
SClkIn	C	6	Delta unit external clock
VSAData3	C	7	VideoStream A data line 3

Name		#	Description
VSAData7	C	8	VideoStream A data line 7
VSBClk	C	9	serial bus clock
PLL Power	C	10	PLL Power/Gnd pin
PllDisable	C	11	
VidResRef	C	12	Reference resistor
VidRed	C	13	Analogue blue signal
MonitorID2	C	14	Monitor ID 2
VSCtl1	C	15	VideoStreams Control line 1
VSCtl5	C	16	VideoStreams Control line 5
VSGPChipSelectN	C	17	VS GP bus chip select
VSBData1	C	18	VideoStream B data line 1
VSBData4	C	19	VideoStream B data line 4
VSBData7	C	20	VideoStream B data line 7
AGPSBA1	D	1	AGP Sideband Address 1
AGPSBA0	D	2	AGP Sideband Address 0
AGPPipeN	D	3	AGP Pipelined address
VSS	D	4	
AGPSt2	D	5	AGP status 2
VDD	D	6	
VSAData4	D	7	VideoStream A data line 4
VSS	D	8	
VSAClk	D	9	VideoStream A clock
SBData	D	10	serial bus data
VDD	D	11	
VidVRef	D	12	Voltage reference
VSS	D	13	
VSCtl0	D	14	VideoStreams Control line 0
VDD	D	15	
VSCtl4	D	16	VideoStreams Control line 4
VSS	D	17	
SBClk	D	18	VideoStream B clock
VSTol	D	19	VideoStreams tolerance supply
TestMode	D	20	Testmode
AGPSBA4	E	1	AGP Sideband Addr 4
AGPSBSTB	E	2	AGP Sideband Addr 2X strobe
AGPSBA3	E	3	AGP Sideband Addr 3
AGPSBA2	E	4	AGP Sideband Addr 2
VSBResetN	E	17	Video Stream B Reset Out
VidRightEye	E	18	Right signal for stereo
VidDDCData	E	19	Data line for DDC
VidDDCClk	E	20	Clock line for DDC
AGPTol	F	1	AGP/PCI tolerance supply
AGPSBA6	F	2	AGP Sideband Addr 6
AGPSBA5	F	3	AGP Sideband Addr 5
VDD	F	4	
VDD	F	17	
Mbank0	F	18	Memory bank select 0
Mbank1	F	19	Memory bank select 1
Mbank2	F	20	Memory bank select 2

Name		#	Description
VDDQ	G	1	
PCIAD30	G	2	PCI address/data line 30
PCIAD31	G	3	PCI address/data line 31
AGPSBA7	G	4	AGP Sideband Addr 7
Mbank3	G	17	Memory bank select 3
Mbyte0	G	18	Memory byte select 0
Mbyte1	G	19	Memory byte select 1
Mbyte2	G	20	Memory byte select 2
PCIAD27	H	1	PCI address/data line 27
PCIAD28	H	2	PCI address/data line 28
PCIAD29	H	3	PCI address/data line 29
VSS	H	4	
VSS	H	17	
Mbyte3	H	18	Memory byte select 3
Mbyte4	H	19	Memory byte select 4
MByte5	H	20	Memory byte select 5
AGPADSTB1	J	1	AGP AD 2X strobe
PCIAD24	J	2	PCI address/data line 24
PCIAD25	J	3	PCI address/data line 25
PCIAD26	J	4	PCI address/data line 26
MByte6	J	17	Memory byte select 6
MByte7	J	18	Memory byte select 7
MDSF0	J	19	Memory DSF line 0
MDSF1	J	20	Memory DSF line 1
PCIAD21	K	1	PCI address/data line 21
PCICBEN3	K	2	PCI byte enable 3
PCIAD23	K	3	PCI address/data line 23
VDD	K	4	
MWE0	K	17	Memory write enable 0
MemClkE	K	18	Memory clock enable
MemClkOut0	K	19	Memory clock 0
MemClkOut1	K	20	Memory clock 1
VDDQ	L	1	
PCIAD19	L	2	PCI address/data line 19
PCIIdSel	L	3	PCI ID select
PCIAD22	L	4	PCI address/data line 22
VDD	L	17	
MWE1	L	18	Memory write enable 1
MRAS0	L	19	Memory RAS line 0
MRAS1	L	20	Memory RAS line 1
PCICBEN2	M	1	PCI byte enable 2
PCIAD18	M	2	PCI address/data line 18
PCIAD17	M	3	PCI address/data line 17
PCIAD20	M	4	PCI address/data line 20
MCAS0	M	17	Memory CAS line 0
MCAS1	M	18	Memory CAS line 1
MDat63	M	19	Memory data line 63
MDat31	M	20	Memory data line 31
PCIFrameN	N	1	PCI frame signal

Name		#	Description
PCIRdyN	N	2	PCI ready
PCIAD16	N	3	PCI address/data line 16
VSS	N	4	
VSS	N	17	
MDat29	N	18	Memory data line 29
MDat62	N	19	Memory data line 62
MDat30	N	20	Memory data line 30
AGPTol	P	1	AGP/PCI tolerance supply
PCIStopN	P	2	PCI stop
PCITRdyN	P	3	PCI T ready
PCIDevSelN	P	4	PCI device select
MDat61	P	17	Memory data line 61
MDat28	P	18	Memory data line 28
MDat60	P	19	Memory data line 60
MDat27	P	20	Memory data line 27
VDDQ	R	1	
PCICBEN1	R	2	PCI byte enable 1
PCIPar	R	3	PCI parity
VDD	R	4	
VDD	R	17	
MDat25	R	18	Memory data line 25
MDat59	R	19	Memory data line 59
MDat26	R	20	Memory data line 26
PCIAD12	T	1	PCI address/data line 12
PCIAD13	T	2	PCI address/data line 13
PCIAD14	T	3	PCI address/data line 14
PCIAD15	T	4	PCI address/data line 15
MDat24	T	17	Memory data line 24
MDat57	T	18	Memory data line 57
MDat23	T	19	Memory data line 23
MDat58	T	20	Memory data line 58
PCIAD9	U	1	PCI address/data line 9
PCIAD10	U	2	PCI address/data line 10
PCIAD11	U	3	PCI address/data line 11
VSS	U	4	
VDDQ	U	5	
VDD	U	6	
MDat5	U	7	Memory data line 5
VSS	U	8	
MDat8	U	9	Memory data line 8
VDD	U	10	
MDat11	U	11	Memory data line 11
MDat13	U	12	Memory data line 13
VSS	U	13	
MDat15	U	14	Memory data line 15
VDD	U	15	
MDat48	U	16	Memory data line 48
VSS	U	17	
MDat55	U	18	Memory data line 55

Name		#	Description
MDat22	U	19	Memory data line 22
MDat56	U	20	Memory data line 56
PCIAD7	V	1	PCI address/data line 7
AGPADSTB0	V	2	AGP AD 2X strobe
PCICBEN0	V	3	PCI byte enable 0
PCIAD8	V	4	PCI address/data line 8
MDat33	V	5	Memory data line 33
MDat3	V	6	Memory data line 3
MDat36	V	7	Memory data line 36
MDat38	V	8	Memory data line 38
MDat40	V	9	Memory data line 40
MDat42	V	10	Memory data line 42
MDat44	V	11	Memory data line 44
MDat46	V	12	Memory data line 46
Maddr9	V	13	Memory address line 9
Maddr6	V	14	Memory address line 6
Maddr3	V	15	Memory address line 3
Maddr0	V	16	Memory address line 0
MDat17	V	17	Memory data line 17
MDat54	V	18	Memory data line 54
MDat21	V	19	Memory data line 21
MDat53	V	20	Memory data line 53
PCIAD3	W	1	PCI address/data line 3
PCIAD4	W	2	PCI address/data line 4
PCIAD5	W	3	PCI address/data line 5
PCIAD6	W	4	PCI address/data line 6
MDat32	W	5	Memory data line 32
MDat34	W	6	Memory data line 34
MDat4	W	7	Memory data line 4
MDat6	W	8	Memory data line 6
MDat7	W	9	Memory data line 7
MDat9	W	10	Memory data line 9
MDat10	W	11	Memory data line 10
MDat12	W	12	Memory data line 12
MDat14	W	13	Memory data line 14
MAddr7	W	14	Memory address line 7
MAddr4	W	15	Memory address line 4
MAddr1	W	16	Memory address line 1
MDat49	W	17	Memory data line 49
MDat18	W	18	Memory data line 18
MDat52	W	19	Memory data line 52
MDat20	W	20	Memory data line 20
PCIAD0	Y	1	PCI address/data line 0
PCIAD1	Y	2	PCI address/data line 1
PCIAD2	Y	3	PCI address/data line 2
MDat1	Y	4	Memory data line 1
MDat0	Y	5	Memory data line 0
MDat2	Y	6	Memory data line 2
MDat35	Y	7	Memory data line 35

Name		#	Description
MDat37	Y	8	Memory data line 37
MDat39	Y	9	Memory data line 39
MDat41	Y	10	Memory data line 41
MDat43	Y	11	Memory data line 43
MDat45	Y	12	Memory data line 45
MDat47	Y	13	Memory data line 47
MAddr8	Y	14	Memory address line 8
MAddr5	Y	15	Memory address line 5
MAddr2	Y	16	Memory address line 2
MDat16	Y	17	Memory data line 16
MDat50	Y	18	Memory data line 50
MDat19	Y	19	Memory data line 19
MDat51	Y	20	Memory data line 51

Table 9.2 Pinlist by Number

10. Package Drawings

10.1 BGA Side

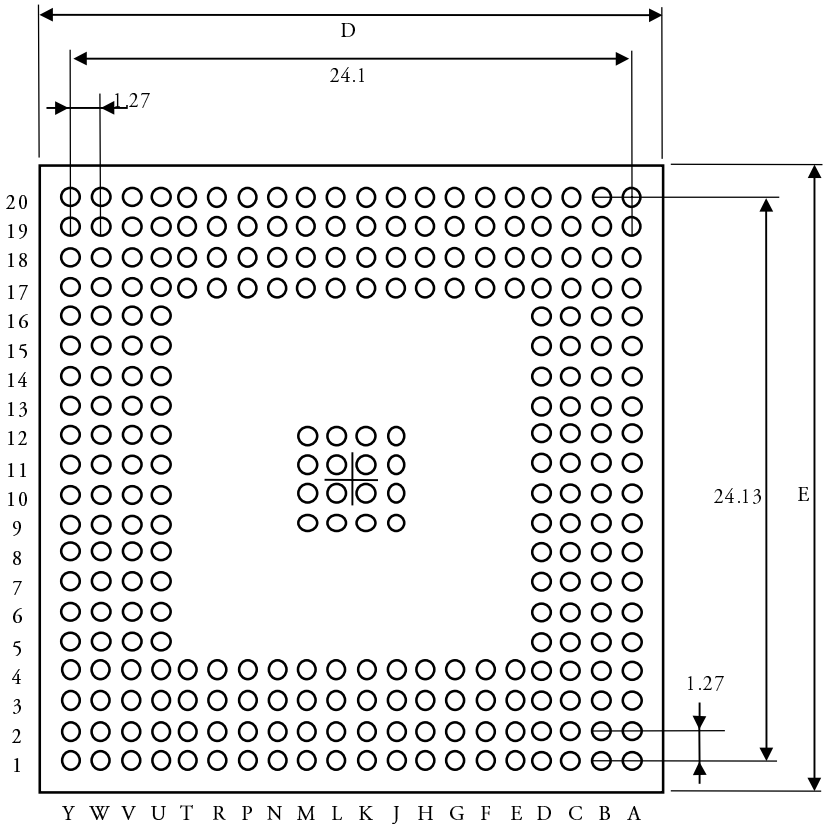


Figure 10.1 BGA Side

JEDEC Code: MO-151

Lead Balls: 272

Thermal Balls: 16 (J9:M12)

Dimension	mm
D length	27 ± 0.2
E width	27 ± 0.2
Height	2.13

For solder ball tolerances refer to the JEDEC specification MO-151 Issue D.

10.2 Overmold Side

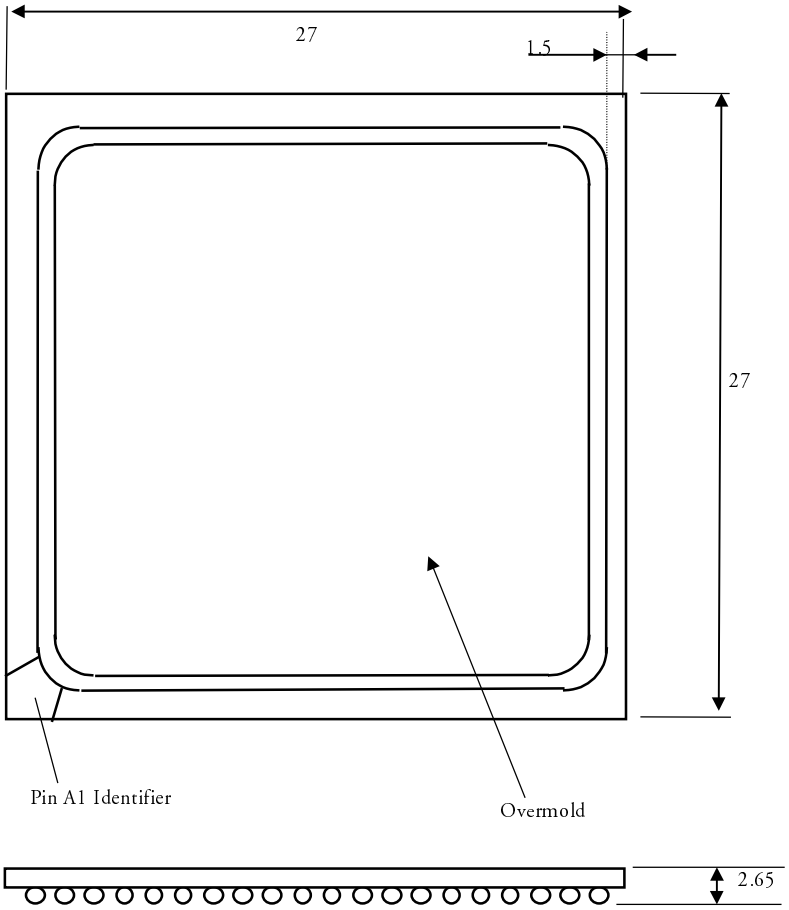


Figure 10.2 Overmold Side

11. Thermal Characteristics

The maximum junction temperature must be kept below T_j(max)and this can only be guaranteed by proper analysis of the operating environment and the thermal path between the die and the air surrounding it.

11.1 Device Characteristics

These are fixed characteristics of the device and are independent of the operating environment or the characteristics of any heatsink:-

$T_{j(max)}$ = 125 °C.
Pd(max) = 3.0 Watts @ Vdd(max), f_{MClk} = 80MHz.
θ_{jt} = 5.5 °C/Watt.

11.2 Thermal Model

The formula used to calculate the junction temperature (T_j) is

$$T_j = T_a + Pd(\theta_{jt} + \theta_{cs} + \theta_{sa})$$
$$= T_a + Pd\theta_{ja}$$

Where:

T_j = Junction temperature (°C)
T_a = Ambient temperature (°C)
Pd = Power dissipation (Watts)
θ_{jt} = Junction to top of case thermal resistance (°C/Watt)
θ_{cs} = Case to Heatsink thermal resistance (°C/Watt)
θ_{sa} = Heatsink to Air thermal resistance (°C/Watt)
θ_{ja} = Total Junction to Air thermal resistance (°C/Watt)

The θ_{ja} form of the equation is more appropriate when there is no heatsink attached to the device (see below).

11.3 Operation Without Heatsink

The 256 PBGA package with no attached heatsink has the following θ_{ja} characteristic as a function of airflow.

Airflow lfpm		**θ_{ja} ° C/W**
0	(Convection Cooling)	26.5
100	(0.5m/sec)	23
400	(2.0m/sec)`	19

Table 11.1 Operation Without Heatsink

Example:-

Ta = 40°C
Airflow = 0 lfpm
T_j = 40 + 3.0 x 26.5
= 119.5 ° C

11.4 Operation With Heatsink

With a heatsink attached to the device the junction temperature will depend on θ_{cs} and θ_{sa}. θ_{cs} is the thermal resistance of the join between the heatsink and the case. θ_{sa} is the thermal resistance of the heatsink and will be a function of system airflow.

Example:-

Ta = 40°C

θ_{cs} = 0.6°C/Watt (EG 7655 epoxy - see below)

$\theta_{sa} \leq (125 - 40)/3.0 - 5.5 - 0.6$

$\leq$ 22.2°C/Watt.

In this example a heatsink must be chosen which has a thermal resistance figure of no greater than 22.2°C/Watt at an airflow matching the expected airflow in the system.

11.4.1 Heatsink Attachment

Two methods have been approved for the purpose of attaching a heatsink directly onto the exposed die surface and the Urethane conformal coating material covering the top of the CQFP package.

11.4.2 Preferred Attachment Method

Thermally conductive epoxy. Either Loctite Output 315 with Loctite 7386 or type EG 7655 from A.I. Technology Inc. The thickness of the epoxy layer should be between 0.05mm and 0.15mm with 100% coverage of the exposed die area, and a maximum voiding in the bond area of 3%. This epoxy should not be used outside the die area.

Typical achievable θ_{cs} using this method is 1.0 ° C/Watt.

11.4.3 Alternative Attachment Method

Chomerics Thermattach 405 thermally conductive tape when used with an additional adhesive, such as Loctite Output 315, for structural support outside the die area.

Typical achievable θ_{cs} using this method is 2.2 ° C/Watt.

Index