RAGE™128
Register Reference Guide

Technical Reference Manual
P/N: RRG-G04100-C Rev 0.01

The information contained in this manual has been carefully checked and is believed to be entirely reliable. No responsibility is assumed for inaccuracies. ATI reserves the right to make changes at any time to improve design and supply the best product possible.

Record of Revisions

Release	Date	Description of Changes
0.01	Nov 1998	Preliminary Release

Related Manuals

RAGE 128 series

- RAGE^{TM} 128 Register Reference Guide (RRG-R04100)
- RAGE^{TM} 128 Graphics Controller Specifications (GCS-C04100)

Table of Contents

Chapter 1: Introduction

1.1 About this Manual 1-1

1.2 Nomenclature and Conventions 1-2
- 1.2.1 Register and Field Names 1-2
- 1.2.2 Numeric Representations 1-2
- 1.2.3 Register Description 1-2
- 1.2.4 Acronyms 1-3

Chapter 2: Overview and Memory Mapping

2.1 General Classification 2-1
- 2.1.1 Setup and Configuration Registers 2-1
- 2.1.2 Host Interface Registers 2-2
- 2.1.3 VGA Registers 2-2
- 2.1.4 Accelerator CRTC and DAC Registers 2-2
- 2.1.5 2D Engine Registers 2-2
- 2.1.6 3D Engine Registers 2-3
- 2.1.7 Concurrent Command Engine Registers 2-4
- 2.1.8 Multimedia Registers 2-4
- 2.1.9 Miscellaneous Registers 2-4

2.2 Memory Mapping 2-5
- 2.2.1 Description of Mapped Memory Apertures 2-6
- 2.2.2 Accessing Bytes, Words, and Dwords 2-8
- 2.2.3 Non-Intel Based Memory Mapping 2-9

Chapter 3: Configuration

3.1 General I/O Control Register 3-1

3.2 Host Path Interface Registers 3-4

3.3 Test and Debug Registers 3-6

3.4 Clock Control and PLL Registers 3-12

3.5 Pixel Cache Registers 3-29

3.6 Power Management Interface Registers 3-42

Chapter 4: Host Interface

4.1 PCI Configuration Registers 4-1
4.2 AGP Registers 4-22
4.3 Bus Control Register 4-25

Chapter 5: VGA Registers

5.1 General VGA Status and Configuration Registers 5-1
5.2 VGA DAC Registers 5-7
5.3 VGA Sequencer Registers 5-9
5.4 VGA CRT Registers 5-13
5.5 VGA Graphics Registers 5-37
5.6 VGA Attribute Registers 5-43

Chapter 6: CRTC and DAC

6.1 CRTC Registers 6-1
6.2 Overscan Registers 6-16
6.3 Hardware Cursor Registers 6-18
6.4 GenLocking Registers 6-22
6.5 Memory Control Registers 6-25
6.6 DAC Control Registers 6-36

Chapter 7: 2D Engine Registers

7.1 Destination Registers 7-1
7.2 GIU Source Registers 7-14
7.3 GUI Host Data Registers 7-17
7.4 Pattern Registers 7-18
7.5 Datapath Registers 7-32
7.6 Scissor Registers 7-46

7.7 Color Compare Registers 7-53
7.8 2D Engine Control Registers 7-55
7.9 2D Engine Status Registers 7-57

Chapter 8: Miscellaneous Registers

8.1 Miscellaneous Registers 8-1
8.2 Scratch Pad Registers 8-14

Appendix A: Reference Table

Chapter 1
Introduction

1.1 About this Manual

This manual serves as a register reference guide to the RAGE 128 graphics controller. It is organized into 8 chapters and 1 appendix.

Chapter 1 outlines the contents of this document and explains the notations and conventions used throughout.

Chapter 2 gives an overview of the registers and describes the memory mapping architecture.

Chapter 3 describes the setup and system configuration registers.

Chapter 4 describes the PCI configuration registers and the AGP registers.

Chapter 5 describes the VGA compatible registers.

Chapter 6 describes the CRTC and DAC registers.

Chapter 7 describes the 2D Accelerator registers which include:

Chaper 8 describes a number of miscellaneous registers.

Finally, for ease of reference, a linked register index is included in Appendix A.

1.2 Nomenclature and Conventions

1.2.1 Register and Field Names

Mnemonics in upper-case are used throughout this document to represent hardware register names and field names. The naming conventions for registers and bit fields are as indicated below:

- REGISTER_MNEMONIC

  For example, CONFIG_CHIP_ID is the mnemonic for the Configuration Chip ID register.

- REGISTER_MNEMONIC[Bit_Numbers] or FIELD_NAME@REGISTER_MNEMONIC

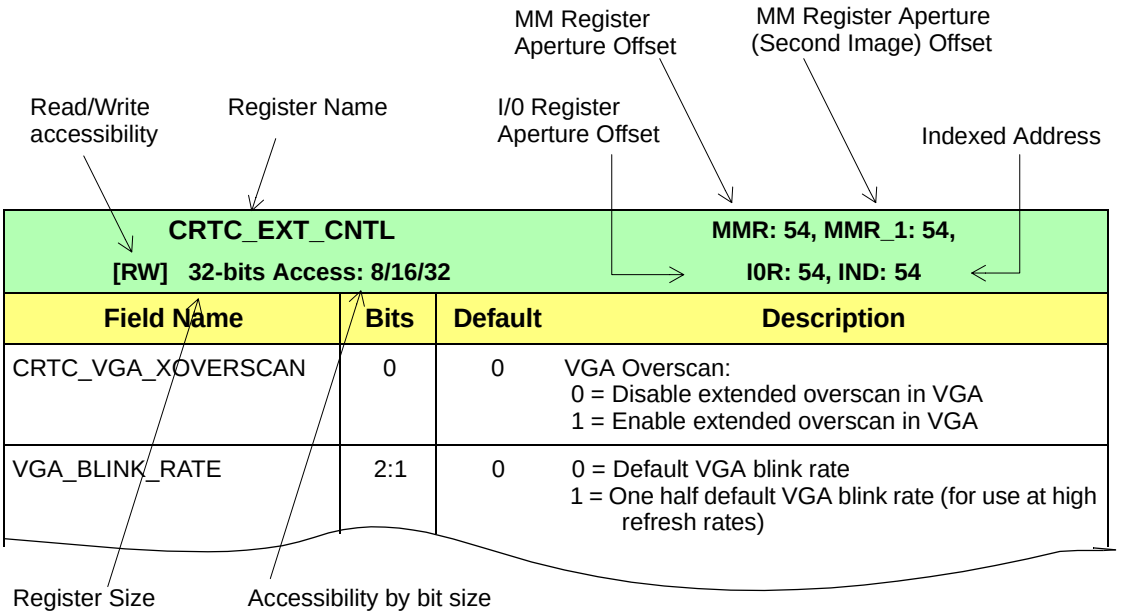
  For example, CONFIG_CHIP_ID[15:0] refers to the bit field that occupies bit positions 0 through 15 within this register, whereas CFG_CHIP_TYPE@CONFIG_CHIP_ID gives the field name CFG_CHIP_TYPE (Product Type Code) instead of the bits position.

1.2.2 Numeric Representations

- Hexadecimal numbers are appended with “h” whenever there is a risk of ambiguity. Other numbers are assumed to be in decimal.
- Registers (or fields) of identical function are sometimes indicated by a single expression in which the part of the signal name that differs is enclosed in [ ] brackets. For example, the eight Host Data registers — HOST_DATA0 through to HOST_DATA7 — are represented by the single expression HOST_DATA[7:0].

1.2.3 Register Description

All registers in this document are described with the format of the self-explained sample table below. All offsets are in hexadecimal notation, while programmed bits are in either binomial or hexadecimal notation. (Note: sometimes not shown are the indirect type of byte offsets, e.g., CFG, PLL, VGA, etc., which will be indicated on the appropriate registers).



1.2.4 Acronyms

Standard acronyms or abbreviations used in the literature are presumed known and therefore freely used without any explanation. When in doubt, refer to Table 1-1 below for a quick check. Less frequently used or ATI-specific acronyms will be accompanied by the full expression when appearing for the first time in the document.

Table 1-1 Acronyms

Acronym	Full Expression
AGP	Accelerated Graphics Port
AMC	ATI Multimedia Channel
BIOS	basic input/output system
bpp	bits per pixel
CCE	Concurrent Command Engine
DAC	digital-to-analog converter
EDO RAM	Extended Data Output RAM
FIFO	first in first out
GUI	graphical user interface

Table 1-1 Acronyms (Continued)

Acronym	Full Expression
I^2C	inter IC's communication
I/O	input/output
MPEG	Motion Picture Experts Group
MPP	Multimedia Peripheral Port
PCI	Peripheral Component Interconnect
PLL	phase-locked loop
POST	power-on self-test
RAMDAC	RAM digital-to-analog converter
RGB	red-green-blue (may refer to a color encoding scheme or a video signal)
R/W	read/write
SDRAM	Synchronous DRAM
SGRAM	Synchronous Graphics RAM
VGA	Video Graphics Array
VIP	Video Interface Port
WRAM	Windows RAM
YUV	A color encoding scheme, no direct correspondence to the letters

Chapter 2
Overview and Memory Mapping

2.1 General Classification

For ease of discussion and reference, the registers are grouped into the following main classes according to their functionality. Note that these are general register classes only. There are instances when specific bit fields of the same register may belong to a different class register. In such cases, it is noted in the register description.

- Setup and Configuration registers
- Host Interface (PCI Configuration Space and AGP) registers
- VGA registers
- Accelerator CRTC and DAC registers
- 2D Engine registers
- 3D Engine registers (**omitted**)
- Multimedia registers (**omitted**)
- Concurrent Command Engine registers (**omitted**)
- Miscellaneous registers

NOTE: As indicated, descriptions of some of the registers are omitted in this document for specific reasons.

The following is an overview of all the registers. As can be seen, some classes are further divided into sub-groups.

2.1.1 Setup and Configuration Registers

Setup and configuration registers are memory mapped and aliased at an I/O address. Most of these registers are initialized only once at boot time. They are further divided into:

- **General I/O Control register** — used to configure the General Purpose I/O pins on the accelerator chip.
- **Bus Control register** — used to configure the on-chip bus interface unit.
- **Memory registers** — used to configure the memory interfaces.
- **Test and Debug registers** — used for chip diagnostics and hardware debugging.

- **Configuration registers** — used to configure the memory aperture and to read the current board configuration.

2.1.2 Host Interface Registers

- **PCI Configuration Space Registers** — used to determine the host bus configuration during system reset. For the RAGE 128, the internal host bus interface has been optimized to support the PCI Version 2.1 bus configuration, providing full 32-bit memory and I/O operations.
- **AGP Registers** — used to configure the Accelerated Graphic Port.

2.1.3 VGA Registers

The VGA registers provide register-level compatibility with the IBM VGA display adapter. They and the accelerator registers are completely segregated from each other, and their functions are mutually exclusive.

2.1.4 Accelerator CRTC and DAC Registers

Accelerator CRTC and DAC Registers are memory mapped and aliased at an I/O address. (Note that accelerator CRTC registers are not the same as the VGA CRTC registers.) They are further divided into the following groups:

- **Accelerator CRTC registers** — used to configure the CRT controller.
- **Clock Control register** — used to configure the pixel clock.
- **PLL registers** — accessed indirectly through the Clock Control register.
- **DAC Control registers** — used to configure the DAC.
- **Overscan registers** — used to configure overscan borders.
- **Hardware Cursor registers** — used to define and move the hardware cursor.

2.1.5 2D Engine Registers

These are divided into two main groups: Trajectory registers and Draw Engine Control registers:

Trajectory Registers

Trajectory registers are memory mapped. They set up the source and destination trajectories and initiate draw operations. They are further divided into two groups:

- **Destination Trajectory registers** — used to define the region in which pixels are drawn. The region may be a a line, a rectangular, or a trapezoidal area.
- **Source Trajectory registers** — used to define a rectangular region from which pixel data is taken. The pixel data may be used as a monochrome or color pixel source, or a polygon fill mask.

Draw Engine Control Registers

Draw Engine Control Registers are memory mapped. They set up the source pixel data, the draw engine data path, and the destination mixing logic. They are divided into the following groups:

- **Host Data registers** — used for transferring data from the host to the draw engine.
- **Pattern registers** — used to enable and define fixed patterns.
- **Scissor registers** — used to define a draw region.
- **Data Path registers** — used to configure the data path and ALU.
- **Color Compare registers** — used to configure the source and destination color compare.
- **Draw Engine Composite Control register** — abbreviated composites of other draw engine control registers.
- **Draw Engine Status register** — used to report the current state of the draw engine.

2.1.6 3D Engine Registers

The 3D Engine registers are memory mapped and are further divided into the following groups:

- **Front-End Scaler Pipe registers** — used to configure the front-end scaler source data and to control any subsequent blending, color conversion, and dithering. Most of the scaler registers are aliased with certain 3D and Texture Mapping registers.
- **Texture Mapping registers** — used to hold the ‘S’ and ‘T’ sample address offsets to the start of the available mipmaps, and to configure the associated quadratic interpolators.
- **Specular, Color, Z, and Alpha Interpolator registers** — used to configure the specular interpolation, the Z buffering and interpolation, the RGB and alpha interpolation, alpha blending, and fogging.
- **Setup Engine registers** — used to setup the draw and color/texture functions.

2.1.7 Concurrent Command Engine Registers

- **Vertex Controller and Floating Point Unit Registers** —
- **Status/Data/Address registers** —
- **Miscellaneous registers** —

2.1.8 Multimedia Registers

These are registers used for multimedia operations such as video capture and playback. They are divided into the following groups:

- **Overlay Window registers** — used to specify the overlayed scaling window dimensions and coordinates to be displayed.
- **Overlay Scaler registers** — used to set up the scaling factors.
- **Video Capture registers** — used to initialize, set the video configuration, define the capture buffer requirements, and trigger the capture.
- **Multimedia Peripheral Port (MPP) registers** — used to configure and access the MPP.
- **Subpicture registers** —
- **VIP Port registers** —
- **Hardware Assisted I2C registers** — used to control a 16-entry deep buffer for storing out-going or in-coming data.
- **iDCT registers** —

2.1.9 Miscellaneous Registers

These are registers that do not quite fit into any of the groups above. Amongst them are:

- **Scratch Pad registers —** used for general purpose storage for the adapter ROM and for communicating the adapter ROM segment location to host applications. In test modes, these registers are used for chip diagnostics.

2.2 Memory Mapping

The RAGE 128 uses a fully memory mapped programming model as shown in the diagram below.

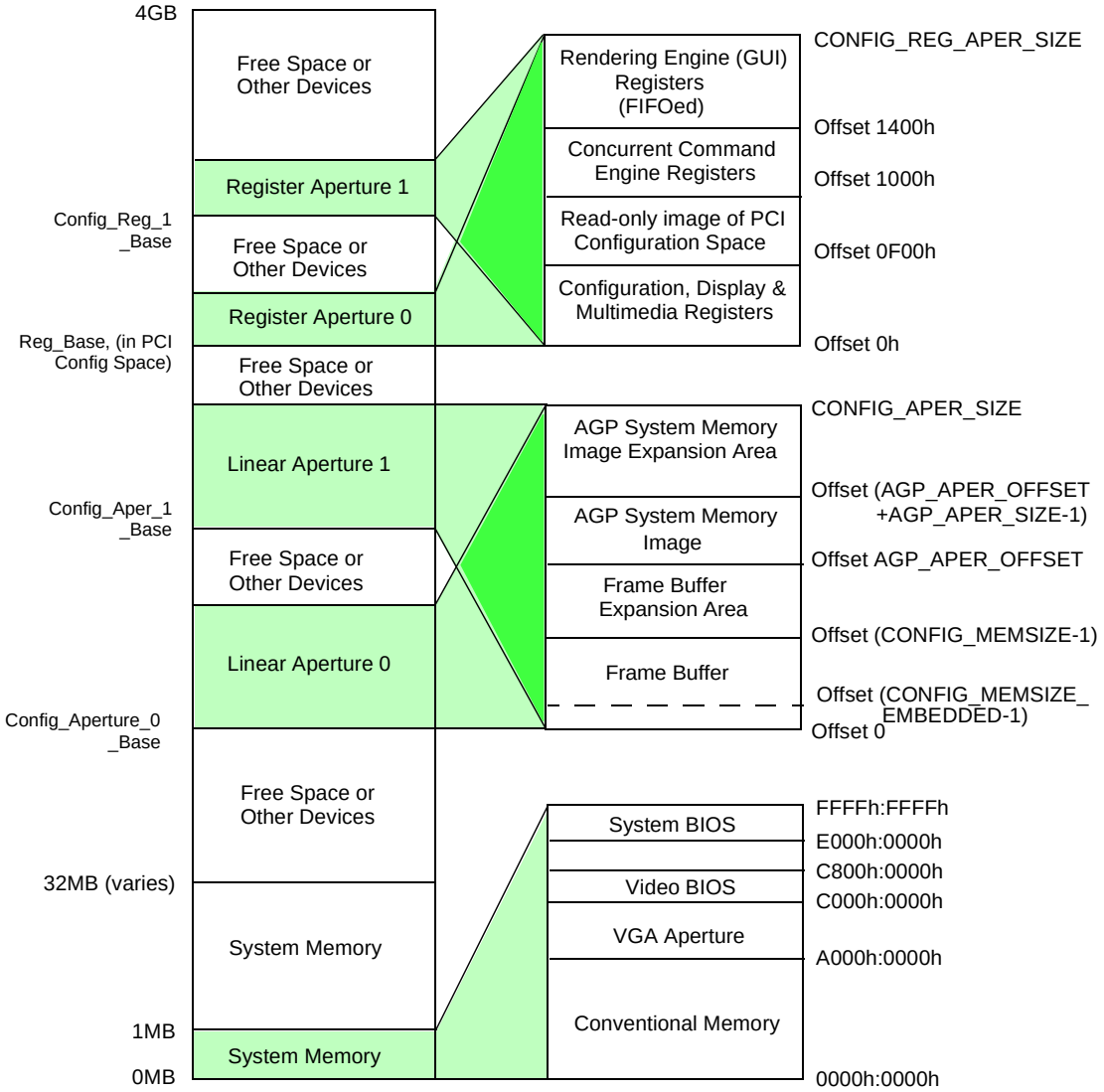


Figure 2-1. Register and Frame Buffer Mapping Within the Host Address Space

2.2.1 Description of Mapped Memory Apertures

Table 2-1 Memory Mapping Summary

Register Group	Space/Offset
PCI POS registers	PCI Configuration Space
VGA compatible registers	Standard VGA addresses in the I/0 space
Non-GUI registers, memory mapped as well as directly accessible in IOR space	0000h - 00FFh
Non-GUI registers, not accessible in IOR space	0100h - 0EFFh
Read-only copy of PCI configuration space	0F00h - 0FFFh
Concurrent Command Engine registers	1000h - 13FFh
GUI registers	1400h - 1FFFh

Video BIOS

The video BIOS for RAGE 128 is re-locatable using the PCI configuration space. The system BIOS will normally shadow the entire BIOS image to the area starting at segment C000h during system initialization. To directly access the actual BIOS memory, as opposed to the shadow copy in system memory, contact ATI for details.

PCI Configuration Space

The PCI POS registers exist only in the PCI configuration space.

VGA Memory Aperture

When enabled for VGA, RAGE 128 claims the standard VGA resources. The VGA memory aperture position and size are determined by the GRPH_ADRSEL bits. For most VGA graphics modes the aperture is 128KB starting at segment A000h.

Register Apertures

RAGE 128 uses two identical copies of the relocatable memory-mapped register aperture. For the PowerMac environment, this allows one aperture to be marked as cacheable. For the Wintel architecture, the second aperture may be used (but this serves no valid purpose).

These apertures contain all the direct-accessed registers on the chip (except VGA and PCI configuration registers). They also have index/data pairs for all indirectly accessed registers and memories.

To determine the base address of Register Aperture 0:

1. Use the REG_BASE register in the PCI configuration space.

-Or-

2. Read from the I/O Register Aperture using MM_INDEX <= F18h and read MM_DATA.

To determine the base address of Register Aperture 1, use the CONFIG_REG_1_BASE register, which can be read in Register Aperture 0 once its base has been found as indicated above. Reading CONFIG_REG_1_BASE is the only method of determining the location of Register Aperture 1 that is forward compatible with future generations of hardware.

To determine the size of each register aperture, use the CONFIG_REG_APER_SIZE register. The size may vary in future generations of the accelerator.

Linear Memory Apertures

There are also two identical copies of the relocatable Linear Memory Aperture in RAGE 128. For the PowerMac environment, this allows each to be independently marked as big-endian or little-endian. For the Wintel architecture, the second aperture may also be used (but this serves no valid purpose).

These apertures allow access to the frame buffer memory, and in AGP systems, access to the AGP memory as seen by the RAGE 128.

To determine the base address of Linear Aperture 0, use the CONFIG_APER_0_BASE register. To determine the base address of Linear Aperture 1, use the CONFIG_APER_1_BASE register. Both these registers can be read in any register aperture.

To determine the size of each linear aperture, use the CONFIG_APER_SIZE register. The size may vary in future generations of the accelerator.

Frame Buffer

The frame buffer image occupies the area in each aperture from offset 0 to CONFIG_MEMSIZE-1.

When CONFIG_MEMSIZE_EMBEDDED is greater than 0, the hardware has on-chip memory for the first piece of the frame buffer. This embedded memory is included in the CONFIG_MEMSIZE total. The RAGE 128 does not have any embedded memory.

The RAGE 128 supports up to 32MB of frame buffer memory. This limit may be expanded for future hardware generations, therefore the software should use the procedures outlined here for determining the aperture base addresses, AGP offsets and aperture size.

AGP System Memory Image

Each Linear Aperture contains an image of the AGP system memory as seen by the accelerator. This image starts at offset AGP_APER_OFFSET in the aperture.

The AGP image is intended for debug work. It allows a method to flush out pending AGP cycles still in the host chipset before the software directly accesses system memory. Software would normally directly access AGP system memory using the system processor. Using this AGP image will generate an AGP slave and an AGP bus master cycle for each access (or group of accesses), and therefore it is not recommended.

To determine the size of the AGP memory, use the AGP_APER_SIZE register. This register is an enumerated type that must be converted into a number (refer to the register definition).

The RAGE 128 supports up to 32MB of AGP memory. This limit may be expanded for future hardware generations, therefore software should use the procedures outlined here for determining the aperture base addresses, AGP offsets and aperture size.

2.2.2 Accessing Bytes, Words, and Dwords

The table below shows the register groups and how they are accessed (bytes, words, or Dwords).

Table 2-2 Accessing Registers

Register Group	Byte Addressing	Word Addressing	Dword Addressing
PCI POS registers	✔	✔	✔
VGA registers	✔	Note 1	Note 1
Display & Configuration	✔	✔ (Note 2)	✔ (Note 2)
GUI registers	✘	✘	✔
Multimedia registers	✘ (Note 3)	✘ (Note 3)	✔
PLL registers	✔	✘	✘

Notes:

1. If two or four VGA registers are continuous in the I/O space, 16 or 32 cycles may be used. The cycle will be broken up internally into 2 or 4 sequential cycles starting with the lowest address first.

2. The DAC_REGS register is actually four 8-bit registers. Word or Dword cycles will be split internally into 2 or 4 sequential cycles starting with the lowest address.
3. The multimedia registers that appear in I/O space are Dword-only registers. This means 32-bit IN or OUT operations must be used.
4. When performing a byte or word access to a 32-bit register, simply add 1, 2 or 3 to the absolute address.
5. It is not recommended to perform word or Dword cycles that span a Dword boundary. This will not work correctly in all cases.

2.2.3 Non-Intel Based Memory Mapping

When incorporating the RAGE 128 into a non-Intel platform (such as the Apple Power Macintosh), make sure the platform conforms to the PCI specification. For information on how to configure the RAGE 128 in non-Intel environments, refer to Chapter 2 of the *mach64 Programmer's Guide*.

This page intentionally left blank.

Chapter 3
Configuration

3.1 General I/O Control Register

GPIO_MONID [RW] 32-bits Access: 8/16/32			MMR: 68, MMR_1: 68, IOR: 68, IND: 68
Field Name	**Bits**	**Default**	**Description**
MONID_A	3:0	0	Software controlled out-going 'A' pin of each MONID pad register bit to pin mapping: GPIO_MONID(0) -> MONID0_AGPIO_MONID(1) -> MONID1_AGPIO_MONID(2) -> MONID2_AGPIO_MONID(3) -> MONID3_A
(reserved)	7:4		
MONID_Y (R)	11:8	0	The 'Y' pins of each pad allow software to read the logic state of each pad GPIO_MONID(5) => MONID0_YGPIO_MONID(6) => MONID1_YGPIO_MONID(7) => MONID2_YGPIO_MONID(8) => MONID3_Y
(reserved)	15:12		
MONID_EN	19:16	0	Pad direction control 0 = Input mode 1 = Output mode GPIO_MONID(8) => MONID0_ENGPIO_MONID(9) => MONID1_ENGPIO_MONID(10) => MONID2_ENGPIO_MONID(11) => MONID3_EN
(reserved)	23:20		
MONID_MASK	27:24	0	0 = GPIO turned off 1 = GPIO enable GPIO_MONID(12) => MONID0_MASKGPIO_MONID(13) => MONID1_MASKGPIO_MONID(14) => MONID2_MASKGPIO_MONID(15) => MONID3_MASK
(reserved)	31:28		

GPIO_MONIDB [RW] 32-bits Access: 8/16/32			MMR: 6C, MMR_1: 6C, IOR: 6C, IND: 6C
Field Name	**Bits**	**Default**	**Description**
MONIDB_A	1:0	0	Software controlled out-going 'A' pin of each AGPIO pad register bit to pin mapping: GPIO_MONIDB(0) -> AGPIO2_A GPIO_MONIDB(1) -> AGPIO3_A
(reserved)	7:2		
MONIDB_Y (R)	9:8	0	The 'Y' pins of each pad allow software to read the logic state of each pad GPIO_MONIDB(8) => AGPIO2_Y GPIO_MONIDB(9) => AGPIO3_Y
(reserved)	15:10		
MONIDB_EN	17:16	0	Pad direction control 0 = Input mode 1 = Output mode GPIO_MONIDB(16) => AGPIO2_EN GPIO_MONIDB(17) => AGPIO3_EN
(reserved)	23:18		
MONIDB_MASK	25:24	0	0 = GPIO turned off 1 = GPIO enable GPIO_MONIDB(24) => AGPIO2_MASK GPIO_MONIDB(25) => AGPIO3_MASK
(reserved)	31:26		

Description:

Monitor Identification register. Secondary Interface. This register permits software control of the secondary Monitor Display Data Channel (DDC) interface AGPIO[3:2] pins to perform Monitor DDC for monitor identification.

This register is used only with the graphics controllers where LCD is enabled. On the boards where LCD is disabled, the primary interface register, GPIO_MONID, should be used for Monitor DDC.

For the secondary Monitor ID interface to function, the CRTC_CUR_MODE[2] register field in the CRTC_GEN_CNTL register must be enabled. This bit enables the functionality of the AGPIO3 and AGPIO2 pins as Monitor DDC.

VIDEOMUX_CNTL [RW] 32-bits Access: 8/16/32			MMR: 190, MMR_1: 190, IND: 190
Field Name	**Bits**	**Default**	**Description**
VIDEO_EN_LOW	0	0	1 = Pdata port used for video capture, 0 = default
VIDEO_EN_HIGH	1	0	1 = part of memory bus (MD[70:79]) is used as extension to video capture, 0 = default
LCD_I2C	2	0	<No Description>
(reserved)	3		
AMCGPIO_GA_DRV	4	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(7:0)
AMCGPIO_GB_DRV	5	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(11:8)
AMCGPIO_GC_DRV	6	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(20:12)
AMCGPIO_GD_DRV	7	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(21)
AMCGPIO_GE_DRV	8	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(23:22)
AMCGPIO_GF_DRV	9	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(25:24)
AMCGPIO_GG_DRV	10	1	<No Description> 0 = supply 4mA 1 = supply 8mA to AMCGPIO(27:26)
(reserved)	15:11		
ROM_CLK_DIVIDE	20:16	5	ROMCLK_DIVIDE is used to adjust ROM timing to big range of xclk frequency.
STR_ROMCLK	21	0	<No Description>
(reserved)	31:22		

3.2 Host Path Interface Registers

HOST_PATH_CNTL [RW] 32-bits Access: 8/16/32			MMR: 130, MMR_1: 130, IND: 130
Field Name	**Bits**	**Default**	**Description**
MREG_ADDR_DELAY	1:0	3	Sets the time that the address and related signals will be valid before the read or write strobe is asserted for on-chip registers. This is set by the BIOS and should not be changed. Too low a value will corrupt register writes and reads. This affects writes to all non-GUI (not FIFOed) registers, and reads to all registers. 0 = 1 MCLK address prop. time 1 = 2 MCLK address prop. time 2 = 3 MCLK address prop. time 3 = 4 MCLK address prop. time
MREG_RD_DELAY	3:2	3	Sets the time the register files have to respond to read requests before the data is expected to be asserted on the shared read bus. This value will be set by the BIOS and should not be changed. Setting it too low could corrupt the register reads. This affects all on-chip registers. 0 = 1 MCLK reg. file read time 1 = 2 MCLK reg. file read time 2 = 3 MCLK reg. file read time 3 = 4 MCLK reg. file read time
MREG_RD_RETURN	6:4	7	Sets the time for register read data to propagate up the read bus to the host interface. Will be set by the BIOS and should not be changed. Register reads could be corrupted if set too low. This affects all on-chip registers. 0 = 1 MCLK read bus prop. time 1 = 2 MCLK read bus prop. time 2 = 3 MCLK read bus prop. time 3 = 4 MCLK read bus prop. time 4 = 5 MCLK read bus prop. time 5 = 6 MCLK read bus prop. time 6 = 7 MCLK read bus prop. time 7 = 8 MCLK read bus prop. time
(reserved)	7		

(Continued)

HOST_PATH_CNTL [RW] 32-bits Access: 8/16/32			MMR: 130, MMR_1: 130, IND: 130
Field Name	**Bits**	**Default**	**Description**
MREG_ARB_CNTL	9:8	0	Determines how access is granted to the internal merged non-GUI register bus when both the host (PCI/AGP) bus and the ProMo4 parser are both trying to use it at the same time. This does not affect access to FIFOed GUI registers. 0 = Round robin host and GUI 1 = Host wins over GUI 2 = GUI wins over host
(reserved)	11:10		
MREG_RCP_EXT	14:12	2	0 = 0 clocks 1 = 8 clocks 2 = 16 clocks 3 = 24 clocks 4 = 32 clocks 5 = 40 clocks 6 = 48 clocks 7 = 64 clocks
(reserved)	23:15		
HP_LIN_RD_CACHE_DIS	24	0	Selects whether to try to service linear aperture slave reads using data from previous read. 0 = Linear aperture slave reads taken from host path cache, if possible. 1 = Linear aperture slave reads always sent to pixel cache.
HP_VGA_RD_CACHE_DIS	25	0	Selects whether to try to service VGA aperture reads using the data from the previous read. 0 = VGA aperture slave reads taken from host path cache, if possible. 1 = VGA aperture slave reads always sent to pixel cache.
(reserved)	30:26		
HP_TEST_RST_CNTL	31	0	For HW test and debugging only. No use to software.

Description:

Controls for the Merged Register Bus internal to the controller for non-FIFO'ed register writes and all register reads.

3.3 Test and Debug Registers

TEST_DEBUG_CNTL [RW] 32-bits Access: 8/16/32			MMR: 120, MMR_1: 120, IND: 120
Field Name	**Bits**	**Default**	**Description**
TEST_DEBUG_OUT_EN	0	0	Enables test and debug output bus on AMC connector pins. Make sure MPP_TVO_EN@MPP_GP_CONFIG and MPP_GP_EN@MPP_GP_CONFIG are turned off when using test_debug bus 0 = Disable 1 = Enable
TEST_DEBUG_IN_LOW_EN	1	0	Enables lower test and debug input bus on video port pins. 0 = Disable 1 = Enable
TEST_DEBUG_IN_HIGH_EN	2	0	Enables upper test and debug input bus on video port pins. 0 = Disable 1 = Enable
TEST_IDDQ_EN	3	0	Sets the device into quiescent current mode. Disables built-in pull-up and pull-down resistors. 0 = Disable 1 = Enable
TEST_VID_WINDOW	4	0	<No Description> 0 = Disable 1 = Enable
TEST_PM4	5	0	<No Description> 0 = Disable 1 = Enable
TEST_DELAY_RING	6	0	<No Description> 0 = Disable 1 = Enable
TEST_REG_BLOCK	7	0	Enables merged register bus controller to assert data on the test and debug output bus. 0 = Disable 1 = Enable
TEST_PLL	8	0	<No Description> 0 = Disable 1 = Enable

(Continued)

TEST_DEBUG_CNTL [RW] 32-bits Access: 8/16/32			MMR: 120, MMR_1: 120, IND: 120
Field Name	**Bits**	**Default**	**Description**
TEST_DISPENG	9	0	<No Description> 0 = Disable 1 = Enable
TEST_RAMDAC	10	0	<No Description> 0 = Disable 1 = Enable
TEST_MEMCTLR	11	0	Select Memory Controller debug signals. 0 = Disable 1 = Enable
TEST_HBIU	12	0	<No Description> 0 = Disable 1 = Enable
TEST_AGP	13	0	<No Description> 0 = Disable 1 = Enable
TEST_2D_GUI	14	0	<No Description> 0 = Disable 1 = Enable
TEST_3D_GUI	15	0	<No Description> 0 = Disable 1 = Enable
(reserved)	16		
TEST_HOSTPATH	17	0	Enables host path controller to assert data on the test and debug output bus. 0 = Disable 1 = Enable
TEST_CMDFIFO_LOCK	18	0	<No Description> 0 = Disable 1 = Enable
TEST_BLOCK_GUI_ INITIATORS	19	0	<No Description> 0 = Disable 1 = Enable
TEST_BLOCK_PM4_ INITIATORS	20	0	<No Description> 0 = Disable 1 = Enable
(reserved)	31:21		

Description:

Test and debug mode control.

TEST_DEBUG_MUX [RW] 32-bits Access: 8/16/32			MMR: 124, MMR_1: 124, IND: 124
Field Name	**Bits**	**Default**	**Description**
TEST_DEBUG_MUX	3:0	0	Debug mode selection bits. Function dependent on which test mode is activated in TEST_DEBUG_CNTL.
TEST_DEBUG_BANK	5:4	0	Additional debug mode selection bits. Same purpose as TEST_DEBUG_MUX.
(reserved)	7:6		
TEST_DEBUG_CLK	12:8	0	Selects the clock signal to mux out as follows: 0 = No clock, output 0. 1 = Oscillator macro internal output (Xtalin) 2 = PPllClk/2 3 = PPll reference divider output 4 = PPll feedback divider output 5 = PPllClk output (slipable) 6 = PPllClk output (fixed) 9 = PCLK (Dispensing word clock) A = ECP (Overlay/scaler clock) B = XPllClk output C = XPllClk/2 D = XPll & MPll reference divider output E = XPll feedback divider output F = XCLK (memory controller main internal clock) 10 = YCLK (memory controller 2x clock for DDR and fast SDR) 11 = DLL test clock 0 12 = DLL test clock 1 13 = MPllClk output 14 = MPllClk/2 15 = MPll feedback divider output 16 = MCLK (non-dynamic version of main engine clock) 17 = GCP (dynamic 2D engine clock) 18 = RCP (dynamic register read/write clock) 19 = PIPE3D_CP (dynamic 3D engine clock) 1A = X1CLK (AGP interface X1 clock) 1B = X2CLK (AGP interface X2 clock) 1C = BCLK (main host interface clock) 1D = F1CP (video capture port 1 clock) 1E = F0CP (video capture port 0 clock)

(Continued)

TEST_DEBUG_MUX [RW] 32-bits Access: 8/16/32			MMR: 124, MMR_1: 124, IND: 124
Field Name	**Bits**	**Default**	**Description**
(reserved)	14:13		
TEST_DEBUG_CLK_INV	15	0	Enables inversion of test clock output. 0 = Non-inverted 1 = Inverted
(reserved)	31:16		

Description:

Mux and clock controls for test and debug modes.

TEST_DEBUG_OUT [R] 32-bits Access: 8/16/32			MMR: 12C, MMR_1: 12C, IND: 12C
Field Name	**Bits**	**Default**	**Description**
TEST_DEBUG_OUTR	10:0	0	Allows read-back of the current state of the TEST_DEBUG_OUT bus. Since register reads are multi-cycle events, the value read can only be considered valid if the value on the bus is static.
(reserved)	31:11		

Description:

Test and Debug Output.

HW_DEBUG [RW] 32-bits Access: 8/16/32			MMR: 128, MMR_1: 128, IND: 128
Field Name	**Bits**	**Default**	**Description**
HW_0_DEBUG	0	0	<No Description> 0 = Normal 1 = Force bm f2s transfers to PCI
HW_1_DEBUG	1	0	<No Description> 0 = Normal 1 = Force bm s2f transfers to PCI
HW_2_DEBUG	2	0	<No Description> 0 = Normal 1 = Enable locked cycle and bus master fix for Triton chip sets
HW_3_DEBUG	3	0	<No Description> 0 = Normal 1 = Enable FILL STALL FLUSH FIX in the Pixel Cache
HW_4_DEBUG	4	0	<No Description>
HW_5_DEBUG	5	0	<No Description>
HW_6_DEBUG	6	0	<No Description> 0 = Normal 1 = Force a soft reset for pm
HW_7_DEBUG	7	0	<No Description> 0 = Normal 1 = Force a soft reset for pc
HW_8_DEBUG	8	0	<No Description> 0 = Normal 1 = Force a soft reset for eng_pix
HW_9_DEBUG	9	0	<No Description> 0 = AGP Revision ID=2.0 1 = AGP Revision ID=1.0
HW_A_DEBUG	10	0	<No Description>
HW_B_DEBUG	11	0	<No Description> 0 = Normal 1 = A LOW Enable the FIX for SLOW BLOCKWRITE in the Pixel Cache
HW_C_DEBUG	12	0	<No Description> 0 = AGP SBA Bus Enabled 1 = AGP SBA Bus Disabled

(Continued)

HW_DEBUG [RW] 32-bits Access: 8/16/32			MMR: 128, MMR_1: 128, IND: 128
Field Name	**Bits**	**Default**	**Description**
HW_D_DEBUG	13	0	<No Description> 0 = Normal 1 = Force AGP accesses to tiled surfaces to be interpreted vertically, but fetched horizontally.
HW_E_DEBUG	14	0	<No Description> 0 = Use CRTC BLANKb to generated LCD DE 1 = Use CRTC DISPLAY_ENABLE to generate LCD DE
HW_F_DEBUG	15	0	<No Description>
(reserved)	31:16		

Description:

For use in chip debugging and minor revisions.

CRC_CMDFIFO_ADDR [R] 32-bits Access: 32			MMR: 740, MMR_1: 740, IND: 740
Field Name	**Bits**	**Default**	**Description**
CRC_CMDFIFO_ADDR	10:0	0	<No Description>
(reserved)	31:11		

CRC_CMDFIFO_DOUT [R] 32-bits Access: 32			MMR: 744, MMR_1: 744, IND: 744
Field Name	**Bits**	**Default**	**Description**
CRC_CMDFIFO_DOUT	31:0	0	<No Description>

3.4 Clock Control and PLL Registers

CLOCK_CNTL_INDEX [RW] 32-bits Access: 8/16/32			MMR: 08, MMR_1: 08, IOR: 08, IND: 08
Field Name	**Bits**	**Default**	**Description**
PLL_ADDR	4:0	0	<No Description>
(reserved)	6:5		
PLL_WR_EN	7	0	<No Description> 0 = Disable 1 = Enable
PPLL_DIV_SEL	9:8	0	<No Description> 0 = PPLL_DIV0 1 = PPLL_DIV1 2 = PPLL_DIV2 3 = PPLL_DIV3
(reserved)	31:10		

CLOCK_CNTL_DATA [RW] 32-bits Access: 8/16/32			MMR: 0C, MMR_1: 0C, IOR: 0C, IND: 0C
Field Name	**Bits**	**Default**	**Description**
PLL_DATA	31:0	0	<No Description>

CLK_PIN_CNTL [RW] 32-bits Access: 8/16/32			PLL: 01
Field Name	**Bits**	**Default**	**Description**
OSC_EN	0	1	<No Description> 0 = Disable 1 = Enable
DCLK_TRI_STATE	1	1	<No Description> 0 = OutputEn 1 = TriState

(Continued)

CLK_PIN_CNTL [RW] 32-bits Access: 8/16/32			PLL: 01
Field Name	**Bits**	**Default**	**Description**
XTL_LOW_GAIN	2	1	<No Description> 0=High GAIN 1=Low GAIN
(reserved)	3		
HCLK0_OUT_EN	4	1	<No Description> 0 = CLK0 pin tristated 1 = CLK0 pin output enabled
HCLK0b_OUT_EN	5	1	<No Description> 0 = CLK0b pin tristated 1 = CLK0b pin output enabled
HCLK1_OUT_EN	6	1	<No Description> 0 = CLK1 pin tristated 1 = CLK1 pin output enabled
HCLK1b_OUT_EN	7	1	<No Description> 0 = CLK1b pin tristated 1 = CLK1b pin output enabled
(reserved)	31:8		

Description:

Control of clock pins.

PPLL_CNTL			PLL: 02
[RW] 32-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
PPLL_RESET	0	1	<No Description> 0 = Not Reset 1 = Reset
PPLL_SLEEP	1	1	<No Description> 1 = Powerdown
(reserved)	7:2		
PPLL_PC_GAIN	10:8	4	<No Description>
PPLL_VC_GAIN	12:11	1	<No Description>
PPLL_DCYC	14:13	2	<No Description>
PPLL_RANGE	15	1	<No Description>
PPLL_ATOMIC_UPDATE_EN	16	0	<No Description> 0 = Atomic Update Disabled 1 = Atomic Update Enabled
PPLL_VGA_ATOMIC_ UPDATE_EN	17	0	<No Description> 0 = VGA Atomic Update Disabled 1 = VGA Atomic Update Enabled
PPLL_ATOMIC_UPDATE_ SYNC	18	0	<No Description> 0 = Update ASAP 1 = Update in VSYNC
(reserved)	31:19		

PPLL_REF_DIV			PLL: 03
[RW] 32-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
PPLL_REF_DIV	9:0	0	Reference divider for PPLL. This is 'M' in the PLL frequency equation: PPllClk = N*PPLL_REF/MReference divider must be >= 2, otherwise divider stops operating. Upper limit determined by PPLL_REF/M must be >= 200kHz.In general, M is set as large as possible to satisfy the last restriction.

(Continued)

PPLL_REF_DIV [RW] 32-bits Access: 8/16/32			PLL: 03
Field Name	**Bits**	**Default**	**Description**
(reserved)	14:10		
PPLL_ATOMIC_ UPDATE_R **(R)**	15	0	Indicates progress of last request for update to PPLL_REF_DIV and/or PPLL_FBx_DIV. 0 = Update done 1 = Update Pending
PPLL_ATOMIC_ UPDATE_W **(W)**	15	0	Used to update new settings into PPLL reference and feedback dividers for GEN-locking. Should be used when setting new PPLL_REF_DIV or PPLL_FBx_DIV, or when changing PPLL_DIV_SEL or VGA_CKSEL to use different PPLL_FBx_DIV. Not required if not enabled in PPLL_ATOMIC_UPDATE_EN or PPLL_VGA_ATOMIC_UPDATE_EN. 0 = No Update 1 = Update
PPLL_REF_DIV_SRC	17:16	0	<No Description> 0 = PPLL_REF = XTALIN 1 = PPLL_REF = PLLMCLK/2 2 = PPLL_REF = PLLXCLK/2
(reserved)	31:18		

Description:

Pixel clock PLL reference divider controls.

Note: The table below represents 4 similar registers: PLL_DIV_0 through to PLL_DIV_3

PLL_DIV_[3:0] [RW] 32-bits Access: 8/16/32			PLL: 04 to 07
Field Name	**Bits**	**Default**	**Description**
PPLL_FB[3:0]_DIV	10:0	0	Feedback divider for PPLL when clock select is [3:0]. This is 'N' in the PLL frequency equation: PPllClk = (N*PPLL_REF)/MFeedback divider must be >= 4, otherwise divider stops operating. Legal range for PPllClk frequency is 125 MHz to 250 MHz. M and N must be chosen to satisfy this restriction, and the upper bound limit on M stated for PPLL_REF_DIV.
(reserved)	14:11		
PPLL_ATOMIC_ UPDATE_R **(R)**	15	0	Indicates progress of last request for update to PPLL_REF_DIV and/or PPLL_FBx_DIV. 0 = Update done 1 = Update Pending
PPLL_ATOMIC_ UPDATE_W **(W)**	15	0	Used to update new settings into PPLL reference and feedback dividers for GEN-locking. Should be used when setting new PPLL_REF_DIV or PPLL_FBx_DIV, or when changing PPLL_DIV_SEL or VGA_CKSEL to use different PPLL_FBx_DIV. Not required if not enabled in PPLL_ATOMIC_UPDATE_EN or PPLL_VGA_ATOMIC_UPDATE_EN. 0 = No Update 1 = Update
PPLL_POST[3:0]_DIV	18:16	0	Selects PPLL post-divider for when PPLL clock select is [3:0]. If doing TV output, then must be set to /2 or /3 as indicated. 0 = VCLK = VCLK_SRC 1 = VCLK = VCLK_SRC/2, required for TV out 565 2 = VCLK = VCLK_SRC/4 3 = VCLK = VCLK_SRC/8 4 = VCLK = VCLK_SRC/3, required for TV out 888 5 = reserved 6 = VCLK = VCLK_SRC/6 7 = VCLK = VCLK_SRC/12
(reserved)	31:19		

Description:

PPLL feedback and post divider settings for when PPLL clock select is [3:0]. PPLL

clock select is VGA_CKSEL@GENMO in VGA modes or
PPLL_DIV_SEL@CLOCK_CNTL_INDEX in non-VGA modes.
CRTC_EXT_DISP_EN@CRTC_GEN_CNTL= 0 indicates VGA mode.

VCLK_ECP_CNTL [RW] 32-bits Access: 8/16/32			PLL: 08
Field Name	**Bits**	**Default**	**Description**
VCLK_SRC_SEL	1:0	0	Selects source of VCLK. If set to BYTE_CLK, then see BYTE_CLK_POST_DIV below to select the VCLK source. Both the clock source you are switching to and from must be running, or the switch will not occur. 0 = VCLK_SRC=PCICLK (input pin) 1 = VCLK_SRC=PCLK (input pin) 2 = VCLK_SRC=BYTE_CLK (see below) 3 = VCLK_SRC=PPllClk
(reserved)	3:2		
VCLK_INVERT	4	0	Used to invert VCLK to get opposite duty cycle. Only takes effect when VCLK_SRC_SEL is using PPllClk, and PPLL_POSTx_DIV is divide-by-1. Don't care in other cases. 0 = Not Invert 1 = Invert
(reserved)	7:5		
ECP_DIV	9:8	0	Determines horizontal replication factor for back-end overlay/scaler output. ECP can not exceed 125 MHz.If VCLK <= 125 MHz, then set ECP = VCLK.If 125 MHz < VCLK <= 250 MHz, then set ECP = VCLK/2.etc. Overlay/scaler will produce one scaled output pixel for each period of ECP. 0 = ECP=VCLK 1 = ECP=VCLK/2 2 = ECP=VCLK/4
(reserved)	15:10		

(Continued)

VCLK_ECP_CNTL [RW] 32-bits Access: 8/16/32			PLL: 08
Field Name	**Bits**	**Default**	**Description**
BYTE_CLK_POST_DIV	17:16	0	Selects the source of BYTE_CLK when TV output port is enabled. Select BYTCLK input pin when ImpacTV/RAGE THEATER are to be the dot clock source. Select the appropriate post-divider when RAGE graphics chip is to supply the dot clock source. In this case, this post-divider determines the speed of BYTE_CLK to the ImpacTV/RAGE THEATER, and the PPLL_POSTx_DIV determines if 16 or 24 data bits per pixel are sent over.Don't care if VCLK_SRC_SEL <> 10. 0 = BYTE_CLK=BYTCLK (input pin) 1 = BYTE_CLK=PPllClk /2 2 = BYTE_CLK=PPllClk /3 3 = BYTE_CLK=PPllClk /4
ECP_FORCE_ON	18	0	Controls the dynamic clock control for the back-end overlay/scaler. Set to low for power reduction. 0 = SCALER ACTIVITY 1 = CONTINUOUS
(reserved)	19		
BYTE_CLK_OUT_EN	20	0	Controls the function of the BYTCLK pin.When BYTE_CLK_POST_DIV = 00 (BYTCLK input), then this bit should be low (tri state). For other settings of BYTE_CLK_POST_DIV, this bit should be high to drive the clock to the TV encoder chip. 0 = Tri State BYTCLK output 1 = Enable BYTCLK output = BYTE_CLK
(reserved)	23:21		

(Continued)

VCLK_ECP_CNTL [RW] 32-bits Access: 8/16/32			PLL: 08
Field Name	**Bits**	**Default**	**Description**
BYTE_CLK_SKEW	26:24	0	Selects phase of internally generated BYTE_CLK to VCLK.Don't care if BYTE_CLK_POST_DIV = 00 (external input). Used to do alignment of TV out data with clock when RAGE graphics chip is generating the dot clock for TV output. Selects phase in 1/2 PPllClk increments. Valid range depends on BYTE_CLK_POST_DIV setting, and may not exceed (2*byte clock post divider)-1.e.g. for byte clock post divider of 3 (BYTE_CLK_POST_DIV=10), then (2*3)-1=5, so BYTE_CLK_SKEW has range 0 to 5.
(reserved)	31:27		

Description:

General controls for the display clocks.VCLK is the pixel, or dot, clock.ECP is the overlay/scaler clock.

HTOTAL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 09
Field Name	**Bits**	**Default**	**Description**
HTOT_PIX_SLIP	3:0	0	Pixel accurate control of HTOTAL. Selects the extra number of pixels to add to each display line. Valid range is 0 to 7. For VGA modes with SEQ_PCLKBY2 = 1 each increment adds two pixels to the line total. For 9-dot VGA text modes, it is not possible to add 8/9ths of a character extra to the HTOTAL value.
(reserved)	7:4		
HTOT_VCLK_SLIP	11:8	0	Not yet implemented. Reserved for future use.
(reserved)	15:12		

(Continued)

HTOTAL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 09
Field Name	**Bits**	**Default**	**Description**
HTOT_PPLL_SLIP	18:16	0	Finest adjustment control. This selects the number of VCO phase slips to do in the PLL at every HSYNC. Each VCO phase slip is equal to 0.2 of a PLLVCLK period.
(reserved)	23:19		
HTOT_CNTL_EDGE	24	0	Select which HTOTAL edge the correction is done on: 0 = rising edge of HSYNC 1 = falling edge of HSYNC
(reserved)	27:25		
HTOT_CNTL_VGA_EN	28	0	Select whether the HTOT_PPLL_SLIP & HTOT_VCLK_SLIP are enable for VGA display modes. 0 = not enabled for VGA modes 1 = enabled for VGA modes
(reserved)	31:29		

Description:

This register is used to fine-tune the horizontal total. This lengthens the time of each display line by sub-character and/or sub-pixel amounts. The purpose is fine adjustment of the overall frame refresh rate for applications that require it (e.g. TV output, GEN-lock to video input).

X_MPLL_REF_FB_DIV [RW] 32-bits Access: 8/16/32			PLL: 0A
Field Name	**Bits**	**Default**	**Description**
X_MPLL_REF_DIV	7:0	0	Reference divider for both MPLL and XPLL. This is 'M' in the PLL frequency equation: PllClk = 2*N*Xtalin/MReference divider must be >= 2, otherwise divider stops operating. Upper limit determined by Xtalin/M must be >= 400kHz.In general, M is set as large as possible to satisfy the last restriction.
XPLL_FB_DIV	15:8	0	Feedback divider for XPLL. This is 'N' in the PLL frequency equation: XPllClk = 2*N*Xtalin/MFeedback divider must be >= 2, otherwise divider stops operating. Legal range for XPllClk frequency is 125 MHz to 250 MHz. M and N must be chosen to satisfy this restriction, and the upper bound limit on M stated above.
MPLL_FB_DIV	23:16	0	Feedback divider for MPLL. This is 'N' in the PLL frequency equation: MPllClk = 2*N*Xtalin/MFeedback divider must be >= 2, otherwise divider stops operating. Legal range for MPllClk frequency is 125 MHz to 250 MHz. M and N must be chosen to satisfy this restriction, and the upper bound limit on M stated above.
(reserved)	31:24		

Description:

PLL reference and feedback settings for MPLL and XPLL.

XPLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0B
Field Name	**Bits**	**Default**	**Description**
XPLL_RESET	0	1	<No Description> 0 = Not Reset 1 = Reset
XPLL_SLEEP	1	1	<No Description> 1 = Powerdown
(reserved)	7:2		
XPLL_PC_GAIN	10:8	4	XPLL charge pump gain setting
XPLL_VC_GAIN	12:11	1	XPLL VCGEN gain setting
XPLL_DCYC	14:13	2	XPLL duty cycle control
XPLL_RANGE	15	1	<No Description>
(reserved)	31:16		

Description:

PLL macro controls for XPLL.

XDLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0C
Field Name	**Bits**	**Default**	**Description**
XDLL0_SLEEP	0	1	Sleep mode for DLL0. 0 = Enabled 1 = PowerDown
XDLL0_RESET	1	1	Reset for DLL0. 0 = Enabled 1 = Reset
XDLL0_RANGE	3:2	2	Frequency range for DLL0. 2 = 80 MHz to 110 MHz 3 = 110 MHz to 150 MHz

(Continued)

XDLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0C
Field Name	**Bits**	**Default**	**Description**
XDLL0_REF_SEL	5:4	0	Reference select for DLL0. 0 = XCLK 1 = HCLK0 pad 2 = not YCLK
XDLL0_FB_SEL	7:6	0	Feedback select for DLL0. 0 = HCLK0 pad 1 = XCLK 2 = Internal feedback
XDLL0_REF_SKEW	10:8	0	Skew of reference signal selected by XDLL0_REF_SEL.
(reserved)	11		
XDLL0_FB_SKEW	14:12	0	Skew of feedback signal selected by XDLL0_FB_SEL.
(reserved)	15		
XDLL1_SLEEP	16	1	Sleep mode for DLL1. 0 = Enabled 1 = PowerDown
XDLL1_RESET	17	1	Reset for DLl1. 0 = Enabled 1 = Reset
XDLL1_RANGE	19:18	2	Frequency range of DLL1. 2 = 80 MHz to 110 MHz 3 = 110 MHz to 150 MHz
XDLL1_REF_SEL	21:20	0	Reference select for DLL1. 0 = XCLK 1 = HCLK0 pad 2 = YCLKb
XDLL1_FB_SEL	23:22	0	Feedback select for DLL1. 0 = HCLK0 pad 1 = XCLK 2 = Internal feedback
XDLL1_REF_SKEW	26:24	0	Skew of reference signal selected by XDLL1_REF_SEL.
(reserved)	27		

(Continued)

XDLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0C
Field Name	**Bits**	**Default**	**Description**
XDLL1_FB_SKEW	30:28	0	Skew of feedback signal selected by XDLL1_FB_SEL.
(reserved)	31		

Description:

DLL Control Register.

XCLK_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0D
Field Name	**Bits**	**Default**	**Description**
XCLK_SRC_SEL	2:0	0	Selection for XCLK. 0 = XCLK = not CPUCLK 1 = XCLK = XPllClk 2 = XCLK = XPllClk/2 3 = XCLK = XPllClk/4 4 = XCLK = XPllClk/8 5 = XCLK = HCLK0 (direct) 6 = XCLK = HCLK1 (direct) 7 = XCLK = XDLL0CLK
(reserved)	3		
YCLK_SRC_SEL	6:4	0	Selection for YCLK. 0 = YCLK = not CPUCLK 1 = YCLK = XPllClk 2 = YCLK = XPllClk/2 3 = YCLK = XPllClk/4 4 = YCLK = XPllClk/8 5 = YCLK = HCLK0 (direct) 6 = YCLK = HCLK1 (direct) 7 = YCLK = XDLL0CLK
(reserved)	7		

(Continued)

XCLK_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0D
Field Name	**Bits**	**Default**	**Description**
HCLK0_SEL	10:8	0	Selection for HCLK0 pin. 0 = HCLK0 = XCLK 1 = HCLK0 = not XCLK 2 = HCLK0 = not YCLK 3 = HCLK0 = YCLK/2 4 = HCLK0 = XDLL0CLK
HCLK0_REC	11	0	Receiver mode for HCLK0 pin. 0 = hysteresis receiver 1 = differential receiver
HCLK1_SEL	14:12	0	Selection for HCLK1 pin. 0 = HCLK1 = XCLK 1 = HCLK1 = not XCLK 2 = HCLK1 = not YCLK 3 = HCLK1 = YCLK/2 4 = HCLK1 = XDLL1CLK
HCLK1_REC	15	0	Receiver mode for HCLK1 pin. 0 = hysteresis receiver 1 = differential receiver
(reserved)	31:16		

Description:

Clock control register for XCLK clock family.

MPLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0E
Field Name	**Bits**	**Default**	**Description**
MPLL_SLEEP	0	1	<No Description> 1 = Powerdown
MPLL_RESET	1	1	<No Description> 0 = Not Reset 1 = Reset
(reserved)	7:2		

(Continued)

MPLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0E
Field Name	**Bits**	**Default**	**Description**
MPLL_PC_GAIN	10:8	4	<No Description>
MPLL_VC_GAIN	12:11	1	<No Description>
MPLL_DCYC	14:13	2	<No Description>
MPLL_RANGE	15	1	<No Description>
(reserved)	31:16		

MCLK_CNTL [RW] 32-bits Access: 8/16/32			PLL: 0F
Field Name	**Bits**	**Default**	**Description**
MCLK_SRC_SEL	2:0	0	MCLK (Main Clock) source selection. Must always switch from a running clock to a running clock, or hang can occur. 0 = CPUCLK 1 = MPllClk/1 2 = MPllClk/2 3 = MPllClk/4 4 = MPllClk/8 5 = XCLK 6 = XTALIN
(reserved)	15:3		
FORCE_GCP	16	0	Controls the dynamic clocking for the 2D engine. Set to low for power reduction. 0 = Dynamic 1 = ForceOn
FORCE_PIPE3D_CP	17	0	Controls the dynamic clocking for the 3D engine. Set to low for power reduction. 0 = Dynamic 1 = ForceOn
FORCE_RCP	18	0	Controls the dynamic clocking for the internal registers. Set to low for power reduction. 0 = Dynamic 1 = ForceOn
(reserved)	31:19		

Description:

General controls for the 'Engine' clock. Also known as the 'Main' clock.

AGP_PLL_CNTL [RW] 32-bits Access: 8/16/32			PLL: 10
Field Name	**Bits**	**Default**	**Description**
APLL_SLEEP	0	0	<No Description>
APLL_RESET	1	0	<No Description>
(reserved)	7:2		
APLL_XSEL	9:8	0	<No Description>
(reserved)	15:10		
APLL_X1_CLK_SKEW	18:16	0	<No Description>
(reserved)	19		
APLL_X2_CLK_SKEW	22:20	0	<No Description>
(reserved)	23		
APLL_TST_EN	24	0	<No Description>
APLL_PUMP_GAIN	25	0	<No Description>
(reserved)	31:26		

FCP_CNTL [RW] 32-bits Access: 8/16/32			PLL: 12
Field Name	**Bits**	**Default**	**Description**
FCP0_SRC_SEL	2:0	4	<No Description> 0 = CPUCLK 1 = DCLK 2 = DCLKb 3 = HREF 4 = GND 5 = HREFb
(reserved)	7:3		

(Continued)

FCP_CNTL [RW] 32-bits Access: 8/16/32			PLL: 12
Field Name	**Bits**	**Default**	**Description**
FCP1_SRC_SEL	10:8	4	<No Description> 0 = CPUCLK 1 = DCLK 2 = DCLKb 3 = HREF 4 = GND 5 = HREFb
(reserved)	31:11		

PLL_TEST_CNTL [RW] 32-bits Access: 8/16/32			PLL: 13
Field Name	**Bits**	**Default**	**Description**
(reserved)	7:0		
TST_DIVIDERS	8	0	<No Description>
PLL_MASK_READ_B	9	0	<No Description>
(reserved)	15:10		
ANALOG_MON	19:16	0	<No Description>
(reserved)	23:20		
TEST_COUNT	31:24	0	<No Description>

3.5 Pixel Cache Registers

PC_GUI_MODE [RW] 32-bits Access: 32			MMR: 1744, MMR_1: 1744, IND: 1744
Field Name	**Bits**	**Default**	**Description**
PC_GUI_PRIORITY	0	0	0 = Host requests are granted when possible regardless of the value of iGUIBUSY (Reset Value) 1 = Host requests are not accepted while the iGUIBUSY signal is asserted. When signal is negated Host requests are granted.
PC_RISE_DF_EN	1	0	0 = The Pixel Cache does not respond to rising edges on the iDOFLUSH input. 1 = The Pixel Cache initiates a flush operation whenever there is a rising edge on iDOFLUSH. A flush initiated this way is reported in the PC_RISE_FLUSH_BSY bit (Reset Value). This functionality is inhibited when PC_BYPASS_EN is set.
PC_FALL_DF_EN	2	0	0 = The Pixel Cache does not respond to falling edges on the iDOFLUSH input (Reset Value). 1 = The Pixel Cache initiates a flush operation whenever there is a falling edge on iDOFLUSH. A flush initiated this way is reported in the PC_FALL_FLUSH_BSY bit. This functionality is inhibited when PC_BYPASS_EN is set.
PC_BYPASS_EN	3	0	0 = The Pixel Cache is not only bypass mode (Reset Value). 1 = The Pixel Cache bypasses all of its internal storage and translates each client operation request directly to requests on the memory ports.
PC_CACHE_SIZE	4	0	0 = Normal Cache Size (Reset Value). 1 = Half-size mode; half of each bank of physical cache used. This functionality is inhibited when either PC_7P2_MODE or PC_BYPASS_EN are set.

(Continued)

PC_GUI_MODE [RW] 32-bits Access: 32			MMR: 1744, MMR_1: 1744, IND: 1744
Field Name	**Bits**	**Default**	**Description**
PC_IGNORE_UNIFY	5	0	0 = Allow the UNIFY hint to cause to the PC to work as a single 4-way set associative cache for allocations and de-allocations (Reset Value) 1 = Ignore the UNIFY hint. In this case, the Pixel Cache behaves as two parallel 2-way set associative caches for allocations and de-allocations, and one 4-way set associative cache for operations.
PC_IGNORE_WRHINT	6	0	0 = Allow the host port to lock out other operations on the Source/Z channel while it is waiting to write the cache (Reset Value). 1 = Disallow host port write locking.
PC_IGNORE_RDHINT	7	0	0 = Allow the host port to lock out other operations on the Source/Z channel while it is waiting to write the cache (Reset Value). 1 = Disallow host port write locking.
PC_RISE_DP_EN	8	0	0 = The Pixel Cache does not respond to rising edges on the iDOPURGE input. 1 = The Pixel Cache initiates (re-initiates) a purge operation when there is a rising edge on iDOPURGE (Reset Value). This functionality is inhibited when PC_BYPASS_EN is set.
(reserved)	31:9		

Description:

This register is used for static setup information that does not change after initialization. Writes to this register are not synchronized, so this should never be written to while the Pixel Cache is operation. Reads from this register reflect the current value of the settings and do not have any side-effects.

PC_NGUI_MODE [RW] 32-bits Access: 32			MMR: 180, MMR_1: 180, IND: 180
Field Name	**Bits**	**Default**	**Description**
PC_GUI_PRIORITY	0	0	0 = Host requests are granted when possible regardless of the value of iGUIBUSY (Reset Value) 1 = Host requests are not accepted while the iGUIBUSY signal is asserted. When signal is negated Host requests are granted.
PC_RISE_DF_EN	1	0	0 = The Pixel Cache does not respond to rising edges on the iDOFLUSH input. 1 = The Pixel Cache initiates a flush operation whenever there is a rising edge on iDOFLUSH. A flush initiated this way is reported in the PC_RISE_FLUSH_BSY bit (Reset Value). This functionality is inhibited when PC_BYPASS_EN is set.
PC_FALL_DF_EN	2	0	0 = The Pixel Cache does not respond to falling edges on the iDOFLUSH input (Reset Value). 1 = The Pixel Cache initiates a flush operation whenever there is a falling edge on iDOFLUSH. A flush initiated this way is reported in the PC_FALL_FLUSH_BSY bit. This functionality is inhibited when PC_BYPASS_EN is set.
PC_BYPASS_EN	3	0	0 = The Pixel Cache is not only bypass mode (Reset Value). 1 = The Pixel Cache bypasses all of its internal storage and translates each client operation request directly to requests on the memory ports.
PC_CACHE_SIZE	4	0	0 = Normal Cache Size (Reset Value). 1 = Half-size mode; half of each bank of physical cache used. This functionality is inhibited when either PC_7P2_MODE or PC_BYPASS_EN are set.
PC_IGNORE_UNIFY	5	0	0 = Allow the UNIFY hint to cause to the PC to work as a single 4-way set associative cache for allocations and de-allocations (Reset Value) 1 = Ignore the UNIFY hint. In this case, the Pixel Cache behaves as two parallel 2-way set associative caches for allocations and de-allocations, and one 4-way set associative cache for operations.

(Continued)

PC_NGUI_MODE [RW] 32-bits Access: 32			MMR: 180, MMR_1: 180, IND: 180
Field Name	**Bits**	**Default**	**Description**
PC_IGNORE_WRHINT	6	0	0 = Allow the host port to lock out other operations on the Source/Z channel while it is waiting to write the cache (Reset Value). 1 = Disallow host port write locking.
PC_IGNORE_RDHINT	7	0	0 = Allow the host port to lock out other operations on the Source/Z channel while it is waiting to write the cache (Reset Value). 1 = Disallow host port write locking.
PC_RISE_DP_EN	8	0	0 = The Pixel Cache does not respond to rising edges on the iDOPURGE input. 1 = The Pixel Cache initiates (re-initiates) a purge operation when there is a rising edge on iDOPURGE (Reset Value). This functionality is inhibited when PC_BYPASS_EN is set.
(reserved)	31:9		

Description:

This register is used for static setup information that does not change after initialization. Writes to this register are not synchronized, so this should never be written to while the Pixel Cache is operation. Reads from this register reflect the current value of the settings and do not have any side-effects.

PC_GUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 1748, MMR_1: 1748, IND: 1748
Field Name	**Bits**	**Default**	**Description**
PC_FLUSH_GUI	1:0	0	Writing 11 to this bitfield forces all the dirty data in the Pixel Cache to be flushed to memory. Writing 01 or 10 - forces dirty data in just the destination on Source/Z channels to be flushed. Operations (both GUI and non-GIU) are still accepted by the Pixel Cache while the flush is occurring. Once a bit in the field is set, it will remain set until the flush is complete, then the Pixel Cache will automatically clear to. This bitfield is reflected in the oGUIBUSY output signal and the PC_GUI_BUSY register bit. Writing 00 to this bitfield has no effect. Writing a 1 to any bit in this bitfield when it is already set resets the flush counter. If any bit in this field is set in the same write operation as any of the bits in PC_FLUSH_NONGUI, only one flush will occur, but it will be reflected in both sets of register bits and the oGUIBUSY output signal. If this bitfield is set after any bit in PC_FLUSH_NONGUI is already set, the flush counter is reset and the flush behaves as if both bits were set in the same write. If any bit in this bitfield is set after any bit in PC_FLUSH_DOFLUSH is already set, the flush counter is reset and the flush operation continues; the set bit(s) in PC_FLUSH_DOFLUSH then remains asserted until the flush cycle completes. Resets to 00.

(Continued)

PC_GUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 1748, MMR_1: 1748, IND: 1748
Field Name	**Bits**	**Default**	**Description**
PC_RI_GUI	3:2	0	(Read Invalidate) Writing 1 to this location forces all data in the Pixel Cache to be marked Read Invalid. Once this bit is set, it will remain set until the marking is complete, then the PC will automatically clear it. While this bit is set, no operations will be accepted from GUI clients (Source, Z, and Destination); however, any operations that had already been accepted will continue to be processed. This bit is reflected in the oGUIBUSY output signal and the PC_GUI_BUSY register bit. Writing 0 to this bit has no effect. Writing 1 to this bit when it is already set resets the read-invalidate counter. If this bit is set in the same write operation as PC_RI_NONGUI, only one read invalidation will occur, but it will be reflected in both sets of register bits and in the oGUIBUSY signal, and all client operations will be blocked. If this bit is set after PC_RI_NONGUI is already set the read-invalidate counter is reset and the read-invalidation behaves as if both bits were set in the same write. Resets to 00.
PC_FLUSH_NONGUI	5:4	0	This bitfield is analogous to the PC_FLUSH_GUI field, except that it causes the non-GUI busy to be set. If a bit in this bitfield is set and a bit in the PC_FLUSH_GUI bitfield is set, both busies will be asserted.
PC_RI_NONGUI	7:6	0	This bitfield is analogous to the PC_RI_GUI field, except that it blocks non-GUI operations and causes the non-GUI busy to be set. If a bit in this bitfield is set and a bit in the PC_RI_GUI bitfield is set, both types of operations will be blocked and busies will be asserted.

(Continued)

PC_GUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 1748, MMR_1: 1748, IND: 1748
Field Name	**Bits**	**Default**	**Description**
PC_PURGE_GUI	8	0	Writing 1 to this bit causes the Pixel Cache to initiate a purge operation. This not only causes the dirty data to be flushed and the read-valid data to be marked invalid, but also it forces the Pixel Cache to de-allocate the memory locations as well. While this is occurring the Pixel Cache will signal itself as being GUI Busy. If this bit is set while a flush or read invalidate is already in progress, that flush or read invalidate will be restarted as a purge.
PC_PURGE_NONGUI	9	0	Writing 1 to this bit causes the Pixel Cache to initiate a purge-operation. This not only causes the dirty data to be flashed and the read-valid data to be marked invalid, but also it forces the PC to de-allocate the memory locations as well. While this is occurring, the Pixel Cache will signal itself as being NONGUI Busy. If this bit is set while a flush or read invalidate is already in progress, that flush or read invalidate will be restarted as a purge.
(reserved)	23:10		
PC_DIRTY **(R)**	24	0	0 = The pixel Cache has no dirty data in it (Reset Value). 1 = The Pixel Cache has at least one piece of data that has been modified but not yet written out to the external memory.
PC_PURGE_DOPURGE **(R)**	25	0	0 = The Pixel Cache is not processing a purge initiated by a rising edge on iDOPURGE (Reset Value). 1 = The Pixel Cache is processing a purge that was initiated by a rising edge on iDOPURGE. If an iDOPURGE edge triggers a purge operation when the PC is already doing a flush, purge, or read invalidate operation, the flush/purge counter is reset and the flush/purge continues.

(Continued)

PC_GUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 1748, MMR_1: 1748, IND: 1748
Field Name	**Bits**	**Default**	**Description**
PC_FLUSH_DOFLUSH **(R)**	26	0	0 = The Pixel Cache is not processing a flush initiated by an edge on iDOFLUSH (Reset Value). 1 = The Pixel Cache is processing a flush that was initiated by an edge on iDOFLUSH. If an iDOFLUSH edge triggers a flush operation when the Pixel Cache is already doing a flush operation (either from a previous iDOFLUSH edge or from setting PC_FLUSH_GUI, or PC_FLUSH_NONGUI, or both), the flush counter is reset and the flush continues.
PC_BUSY_INIT **(R)**	27	0	0 = The Pixel Cache is not processing an initialize operation. 1 = The Pixel Cache is currently going through an init cycle. This occurs just after reset negates.
PC_BUSY_FLUSH **(R)**	28	0	0 = The Pixel Cache is not processing a flush operation. 1 = The Pixel Cache is currently a Flush, Read Invalidate, or Purge operation.
PC_BUSY_GUI **(R)**	29	0	0 = The Pixel Cache is not processing an GUI operations (Reset Value). 1 = The Pixel Cache is performing a GUI operation. This includes operations initiated by Source/Z or Destination client requests, setting PC_FLUSH_GUI, or setting PC_RI_GUI.
PC_BUSY_NGUI **(R)**	30	0	0 = The Pixel Cache is not processing an non-GUI operations (Reset Value). 1 = The Pixel Cache is performing a non-GUI operation. This includes operations initiated by Hostclient requests, setting PC_FLUSH_NONGUI, or setting PC_RI_NONGUI.
PC_BUSY **(R)**	31	0	0 = The Pixel Cache is completely idle. This does NOT imply there is no dirty data in the Pixel Cache (Reset Value). 1 = The Pixel Cache has at least one pending client operation. In effect, this is logical OR of PC_BUSY_INIT, PC_BUSY_GUI, PC_BUSY_NONGUI, and PC_FLUSH_DOFLUSH.

Description:

This is the real-time interface between the software and the Pixel Cache. Writes to this register are synchronized and cause the Pixel Cache to immediately change its behavior. Reads from this register reflect current status and have no side effects.

PC_NGUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 184, MMR_1: 184, IND: 184
Field Name	**Bits**	**Default**	**Description**
PC_FLUSH_GUI	1:0	0	Writing 11 to this bitfield forces all the dirty data in the Pixel Cache to be flushed to memory. Writing 01 or 10 - forces dirty data in just the destination on Source/Z channels to be flushed. Operations (both GUI and non-GIU) are still accepted by the Pixel Cache while the flush is occurring. Once a bit in the field is set, it will remain set until the flush is complete, then the Pixel Cache will automatically clear to. This bitfield is reflected in the oGUIBUSY output signal and the PC_GUI_BUSY register bit. Writing 00 to this bitfield has no effect. Writing a 1 to any bit in this bitfield when it is already set resets the flush counter. If any bit in this field is set in the same write operation as any of the bits in PC_FLUSH_NONGUI, only one flush will occur, but it will be reflected in both sets of register bits and the oGUIBUSY output signal. If this bitfield is set after any bit in PC_FLUSH_NONGUI is already set, the flush counter is reset and the flush behaves as if both bits were set in the same write. If any bit in this bitfield is set after any bit in PC_FLUSH_DOFLUSH is already set, the flush counter is reset and the flush operation continues; the set bit(s) in PC_FLUSH_DOFLUSH then remains asserted until the flush cycle completes. Resets to 00.

(Continued)

PC_NGUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 184, MMR_1: 184, IND: 184
Field Name	**Bits**	**Default**	**Description**
PC_RI_GUI	3:2	0	(Read Invalidate) Writing 1 to this location forces all data in the Pixel Cache to be marked Read Invalid. Once this bit is set, it will remain set until the marking is complete, then the PC will automatically clear it. While this bit is set, no operations will be accepted from GUI clients (Source, Z, and Destination); however, any operations that had already been accepted will continue to be processed. This bit is reflected in the oGUIBUSY output signal and the PC_GUI_BUSY register bit. Writing 0 to this bit has no effect. Writing 1 to this bit when it is already set resets the read-invalidate counter. If this bit is set in the same write operation as PC_RI_NONGUI, only one read invalidation will occur, but it will be reflected in both sets of register bits and in the oGUIBUSY signal, and all client operations will be blocked. If this bit is set after PC_RI_NONGUI is already set the read-invalidate counter is reset and the read-invalidation behaves as if both bits were set in the same write. Resets to 00.
PC_FLUSH_NONGUI	5:4	0	This bitfield is analogous to the PC_FLUSH_GUI field, except that it causes the non-GUI busy to be set. If a bit in this bitfield is set and a bit in the PC_FLUSH_GUI bitfield is set, both busies will be asserted.
PC_RI_NONGUI	7:6	0	This bitfield is analogous to the PC_RI_GUI field, except that it blocks non-GUI operations and causes the non-GUI busy to be set. If a bit in this bitfield is set and a bit in the PC_RI_GUI bitfield is set, both types of operations will be blocked and busies will be asserted.

(Continued)

PC_NGUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 184, MMR_1: 184, IND: 184
Field Name	**Bits**	**Default**	**Description**
PC_PURGE_GUI	8	0	Writing 1 to this bit causes the Pixel Cache to initiate a purge operation. This not only causes the dirty data to be flushed and the read-valid data to be marked invalid, but also it forces the Pixel Cache to de-allocate the memory locations as well. While this is occurring the Pixel Cache will signal itself as being GUI Busy. If this bit is set while a flush or read invalidate is already in progress, that flush or read invalidate will be restarted as a purge.
PC_PURGE_NONGUI	9	0	Writing 1 to this bit causes the Pixel Cache to initiate a purge-operation. This not only causes the dirty data to be flashed and the read-valid data to be marked invalid, but also it forces the PC to de-allocate the memory locations as well. While this is occurring, the Pixel Cache will signal itself as being NONGUI Busy. If this bit is set while a flush or read invalidate is already in progress, that flush or read invalidate will be restarted as a purge.
(reserved)	23:10		
PC_DIRTY	24	0	0 = The pixel Cache has no dirty data in it (Reset Value). 1 = The Pixel Cache has at least one piece of data that has been modified but not yet written out to the external memory.
PC_PURGE_DOPURGE	25	0	0 = The Pixel Cache is not processing a purge initiated by a rising edge on iDOPURGE (Reset Value). 1 = The Pixel Cache is processing a purge that was initiated by a rising edge on iDOPURGE. If an iDOPURGE edge triggers a purge operation when the PC is already doing a flush, purge, or read invalidate operation, the flush/purge counter is reset and the flush/purge continues.

(Continued)

PC_NGUI_CTLSTAT [RW] 32-bits Access: 32			MMR: 184, MMR_1: 184, IND: 184
Field Name	**Bits**	**Default**	**Description**
PC_FLUSH_DOFLUSH	26	0	0 = The Pixel Cache is not processing a flush initiated by an edge on iDOFLUSH (Reset Value). 1 = The Pixel Cache is processing a flush that was initiated by an edge on iDOFLUSH. If an iDOFLUSH edge triggers a flush operation when the Pixel Cache is already doing a flush operation (either from a previous iDOFLUSH edge or from setting PC_FLUSH_GUI, or PC_FLUSH_NONGUI, or both), the flush counter is reset and the flush continues.
PC_BUSY_INIT **(R)**	27	0	0 = The Pixel Cache is not processing an initialize operation. 1 = The Pixel Cache is currently going through an init cycle. This occurs just after reset negates.
PC_BUSY_FLUSH **(R)**	28	0	0 = The Pixel Cache is not processing a flush operation. 1 = The Pixel Cache is currently a Flush, Read Invalidate, or Purge operation.
PC_BUSY_GUI **(R)**	29	0	0 = The Pixel Cache is not processing an GUI operations (Reset Value). 1 = The Pixel Cache is performing a GUI operation. This includes operations initiated by Source/Z or Destination client requests, setting PC_FLUSH_GUI, or setting PC_RI_GUI.
PC_BUSY_NGUI **(R)**	30	0	0 = The Pixel Cache is not processing an non-GUI operations (Reset Value). 1 = The Pixel Cache is performing a non-GUI operation. This includes operations initiated by Hostclient requests, setting PC_FLUSH_NONGUI, or setting PC_RI_NONGUI.
PC_BUSY **(R)**	31	0	0 = The Pixel Cache is completely idle. This does NOT imply there is no dirty data in the Pixel Cache (Reset Value). 1 = The Pixel Cache has at least one pending client operation. In effect, this is logical OR of PC_BUSY_INIT, PC_BUSY_GUI, PC_BUSY_NONGUI, and PC_FLUSH_DOFLUSH.

Description:

This is the real-time interface between the software and the Pixel Cache. Writes to this register are synchronized and cause the Pixel Cache to immediately change its behavior. Reads from this register reflect current status and have no side effects.

PC_DEBUG_MODE [RW] 32-bits Access: 32			MMR: 1760, MMR_1: 1760, IND: 1760
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

3.6 Power Management Interface Registers

PMI_CAP_ID [W] 8-bits Access: 8/16/32			CFG: 5C, MMR: 00, MMR_1: 00, IND: 00
Field Name	**Bits**	**Default**	**Description**
PMI_CAP_ID	7:0	1	Indicates this capability is the PCI Power Management Interface (PMI) register. 1 = PCI Bus Power Management Interface (PMI) register section

Description:

Capability ID.

PMI_REGISTER [R] 32-bits Access: 8/16/32			CFG: 5C, MMR: F5C, MMR_1: F5C, IND: F5C
Field Name	**Bits**	**Default**	**Description**
PMI_CAP_ID	7:0	1	<No Description>
PMI_NXT_CAP_PTR	15:8	0	<No Description>
PMI_REV	18:16	1	<No Description>
PME_CLOCK	19	0	<No Description>
AUX_POWER_SRC	20	0	<No Description>
DSI_DEV_SPECIFIC_INIT	21	0	<No Description>
(reserved)	24:22		
D1_SUPPORT	25	1	<No Description>
D2_SUPPORT	26	1	<No Description>
PME_SUPPORT	31:27	0	<No Description>

PMI_NXT_CAP_PTR [R] 8-bits Access: 8/16/32			CFG: 5D, MMR: F5D, MMR_1: F5D, IND: F5D
Field Name	**Bits**	**Default**	**Description**
PMI_NXT_CAP_PTR	7:0	0	0 = Last function in capabilities list

Description:

Next capability pointer.

PMI_PMC_REG [R] 16-bits Access: 8/16/32			CFG: 5E, MMR: F5E, MMR_1: F5E, IND: F5E
Field Name	**Bits**	**Default**	**Description**
PMI_VERSION	2:0	1	1 = Compliant with PMI Specification version 1.0
PMI_PME_CLOCK	3	0	0 = No PCI clock needed to generate PME#. Function can not assert PME#.
(reserved)	4		
PMI_DEV_SPECIFIC_INIT	5	0	0 = Device specific initialization not needed for this device.
(reserved)	8:6		
PMI_D1_SUPPORT	9	1	1 = Power state D1 (standby) supported by this device.
PMI_D2_SUPPORT	10	0	0 = Power state D2 (suspend) not supported by this device.
PMI_PME_SUPPORT	15:11	0	00000 = Device can not assert PME# from any power state.

Description:

PCI PMI Power Management Capabilities (PMC).

PMI_PMCSR_REG [R/W] 16-bits Access: 8/16/32			CFG: 60, MMR: F60 [R], MMR_1: F60 [R], IND: F60 [R]
Field Name	**Bits**	**Default**	**Description**
PMI_POWER_STATE	1:0	0	Write: Sets device into specified power state. Writes of unsupported states are not accepted. Read: Indicates current power state of the device. 00 = D0 state (on). 01 = D1 state (standby). 10 = D2 state (suspend, not support on Rage 128). 11 = D3 state (off).
(reserved)	7:2		
PMI_PME_EN	8	0	0 = Device does not support PME# generation.
PMI_DATA_SELECT	12:9	0	Device does not support the PMI data register. Will read back zeros.
PMI_DATA_SCALE	14:13	0	Device does not support the PMI data register. Will read back zeros.
PMI_PME_STATUS	15	0	0 = Device does not support PME# generation.

Description:

PCI PMI Power Management Control/Status.

PMI_DATA [R] 8-bits Access: 8/16/32			CFG: 63, MMR: F63, MMR_1: F63, IND: F63
Field Name	**Bits**	**Default**	**Description**
PMI_DATA	7:0	0	PMI data register not supported in Rage 128. Will read back zeros.

Description:

PCI PMI Data Register.

BUS_CNTL1 [RW] 32-bits Access: 8/16/32			MMR: 34, MMR_1: 34, IOR: 34
Field Name	**Bits**	**Default**	**Description**
PMI_IO_DISABLE	0	0	<No Description> 0 = Normal 1 = Disable
PMI_MEM_DISABLE	1	0	<No Description> 0 = Normal 1 = Disable
PMI_BM_DISABLE	2	0	<No Description> 0 = Normal 1 = Disable
PMI_INT_DISABLE	3	0	<No Description> 0 = Normal 1 = Disable
(reserved)	31:4		

This page intentionally left blank.

4.1 PCI Configuration Registers

VENDOR_ID [R] 16-bits Access: 8/16/32			CFG: 00, MMR: F00, MMR_1: F00, IND: F00
Field Name	**Bits**	**Default**	**Description**
VENDOR_ID	15:0	1002	ATI vendor id number.

DEVICE_ID [R] 16-bits Access: 8/16/32			CFG: 02, MMR: F02, MMR_1: F02, IND: F02
Field Name	**Bits**	**Default**	**Description**
DEVICE_ID	15:0	524b	Device ID number. Two character ASCII code indicating device and configuration.For Rage 128 the following device ID's are defined: 'RE' (5245h) = 312+16 BGA, PCI'RF' (5246h) = 312+16 BGA, AGP 1x and 2x'RK' (524Bh) = 256+16 BGA, PCI'RL' (524Ch) = 256+16 BGA, AGP 1x and 2x.

Description:

Device ID code.

COMMAND [RW] 16-bits Access: 8/16/32			CFG: 04, MMR: F04 [R], MMR_1: F04 [R], IND: F04 [R]
Field Name	**Bits**	**Default**	**Description**
IO_ACCESS_EN	0	0	<No Description> 0=Disable 1=Enable

(Continued)

COMMAND [RW] 16-bits Access: 8/16/32			CFG: 04, MMR: F04 [R], MMR_1: F04 [R], IND: F04 [R]
Field Name	**Bits**	**Default**	**Description**
MEM_ACCESS_EN	1	0	Enable PCI Memory Cycles 0 = Disable 1 = Enable
BUS_MASTER_EN	2	0	Enable Busmaster 0 = Disable 1 = Enable
SPECIAL_CYCLE_EN	3	0	Monitor Special Cycles 0 = Disable (always)
MEM_WRITE_INVALIDATE_EN	4	0	Enable use of memory unite and invalidate for busmaster 0 = Disable (always)
PAL_SNOOP_EN	5	0	Enable Palette Snooping 0 = Disable 1 = Enable
PARITY_ERROR_EN	6	0	Monitor Parity Error 0 = Disable
AD_STEPPING	7	1	Use AD Stepping
SERR_EN	8	0	Enable SERR# 0 = Disable (always)
FAST_B2B_EN	9	0	Enable Fast back-to-back cycle 0 = Disable 1 = Enable
(reserved)	15:10		

STATUS [RW] 16-bits Access: 8/16/32			CFG: 06, MMR: F06 [R], MMR_1: F06 [R], IND: F06 [R]
Field Name	**Bits**	**Default**	**Description**
(reserved)	3:0		
CAP_LIST	4	1	Support Capabilities
PCI_66_EN	5	1	Support PCI 66MHz mode 0 = Disable 1 = Enable

(Continued)

STATUS [RW] 16-bits Access: 8/16/32			CFG: 06, MMR: F06 [R], MMR_1: F06 [R], IND: F06 [R]
Field Name	**Bits**	**Default**	**Description**
UDF_EN	6	0	Support UDF 0 = Disable (always)
FAST_BACK_CAPABLE	7	1	Support Fast back-to-back cycle
(reserved)	8		
DEVSEL_TIMING	10:9	1	Medium decoding
SIGNAL_TARGET_ABORT	11	0	Target Abort not supported. 0 = Disable (always)
RECEIVED_TARGET_ABORT	12	0	Recieved Target Abort 0 = Inactive 1 = Active
RECEIVED_MASTER_ ABORT	13	0	Recieved Master Abort 0 = Inactive 1 = Active
SIGNALED_SYSTEM_ ERROR	14	0	Signalled SERR# 0 = Disable (always)
PARITY_ERROR_DETECTED	15	0	Parity error detected 0 = Disable (always)

REVISION_ID [R] 8-bits Access: 8/16/32			CFG: 08, MMR: F08, MMR_1: F08, IND: F08
Field Name	**Bits**	**Default**	**Description**
MINOR_REV_ID	3:0	0	Incremented for minor revisions. Normally involving debugging, but not new significant features. 0000 = initial RAGE 128 revision.
MAJOR_REV_ID	7:4	0	Indicates major revisions within the RAGE 128 family. Normally incremented when major features added. DEVICE_ID's are normally changed when MAJOR_REV_ID changes. 0000 = Initial version of RAGE 128.

Description:

Indicates relative position of the device within the 'Rage 128' family.

REGPROG_ID [R] 8-bits Access: 8/16/32			CFG: 09, MMR: F09, MMR_1: F09, IND: F09
Field Name	**Bits**	**Default**	**Description**
REG_LEVEL_PROG_INF	7:0	0	<No Description>

SUB_CLASS [R] 8-bits Access: 8/16/32			CFG: 0A, MMR: F0A, MMR_1: F0A, IND: F0A
Field Name	**Bits**	**Default**	**Description**
(reserved)	6:0		
SUB_CLASS_INF	7	1	<No Description>

BASE_CODE [R] 8-bits Access: 8/16/32			CFG: 0B, MMR: F0B, MMR_1: F0B, IND: F0B
Field Name	**Bits**	**Default**	**Description**
BASE_CLASS_CODE	7:0	3	<No Description>

CACHE_LINE [RW] 8-bits Access: 8/16/32			CFG: 0C, MMR: F0C [R], MMR_1: F0C [R], IND: F0C [R]
Field Name	**Bits**	**Default**	**Description**
CACHE_LINE_SIZE	7:0	0	<No Description>

LATENCY [RW] 8-bits Access: 8/16/32			CFG: 0D, MMR: F0D [R], MMR_1: F0D [R], IND: F0D [R]
Field Name	**Bits**	**Default**	**Description**
LATENCY_TIMER	7:0	0	<No Description>

HEADER [R] 8-bits Access: 8/16/32			CFG: 0E, MMR: F0E, MMR_1: F0E, IND: F0E
Field Name	**Bits**	**Default**	**Description**
HEADER_TYPE	6:0	0	<No Description>
DEVICE_TYPE	7	0	<No Description> 0=Single-Function Device 1=Multi-Function Device

BIST [R] 8-bits Access: 8/16/32			CFG: 0F, MMR: F0F, MMR_1: F0F, IND: F0F
Field Name	**Bits**	**Default**	**Description**
BIST_COMP	3:0	0	<No Description>
(reserved)	5:4		
BIST_STRT	6	0	<No Description>
BIST_CAP	7	0	<No Description>

MEM_BASE [RW] 32-bits Access: 8/16/32			CFG: 10, MMR: F10 [R], MMR_1: F10 [R], IND: F10 [R]
Field Name	**Bits**	**Default**	**Description**
(reserved)	2:0		
PREFETCH_EN	3	1	<No Description>
(reserved)	26:4		
MEM_BASE	31:27	0	<No Description>

IO_BASE [RW] 32-bits Access: 8/16/32			CFG: 14, MMR: F14 [R], MMR_1: F14 [R], IND: F14 [R]
Field Name	**Bits**	**Default**	**Description**
BLOCK_IO_BIT	7:0	1	<No Description> NOTE: Bits 7:1 of this field are hardwired to ZERO
IO_BASE	31:8	0	<No Description>

REG_BASE [RW] 32-bits Access: 8/16/32			CFG: 18, MMR: F18 [R], MMR_1: F18 [R], IND: F18 [R]
Field Name	**Bits**	**Default**	**Description**
(reserved)	13:0		
REG_BASE	31:14	0	<No Description>

ADAPTER_ID [R] 32-bits Access: 8/16/32			CFG: 2C, MMR: F2C, MMR_1: F2C, IND: F2C
Field Name	**Bits**	**Default**	**Description**
SUBSYSTEM_VENDOR_ID	15:0	0	<No Description>
SUBSYSTEM_ID	31:16	0	<No Description>

BIOS_ROM [RW] 32-bits Access: 8/16/32			CFG: 30, MMR: F30 [R], MMR_1: F30 [R], IND: F30 [R]
Field Name	**Bits**	**Default**	**Description**
BIOS_ROM_EN	0	0	<No Description> 0=Disable 1=Enable
(reserved)	16:1		
BIOS_BASE_ADDR	31:17	0	<No Description>

CAPABILITIES_PTR [R] 32-bits Access: 8/16/32			CFG: 34, MMR: F34, MMR_1: F34, IND: F34
Field Name	**Bits**	**Default**	**Description**
CAP_PTR	7:0	0	<No Description>
(reserved)	31:8		

INTERRUPT_LINE [RW] 8-bits Access: 8/16/32			CFG: 3C, MMR: 3C [R], MMR_1: 3C [R], IND: 3C [R]
Field Name	**Bits**	**Default**	**Description**
INTERRUPT_LINE	7:0	ff	<No Description>

INTERRUPT_PIN [RW] 8-bits Access: 8/16/32			CFG: 3D, MMR: F3D MMR_1: F3D, IND: F3D
Field Name	**Bits**	**Default**	**Description**
INTERRUPT_PIN [**R**]	0	0	Indicates to system if device wants an interrupt resource. 0 = No interrupt wanted (strapped to disable interrupt). 1 = INTA# requested (strapped to enable interrupt).
(reserved)	7:1		

Description:

Interrupt resource request.

MIN_GRANT [R] 8-bits Access: 8/16/32			CFG: 3E, MMR: F3E MMR_1: F3E, IND: F3E
Field Name	**Bits**	**Default**	**Description**
MIN_GNT	7:0	8	<No Description>

MAX_LATENCY [R] 8-bits Access: 8/16/32			CFG: 3F, MMR: F3F, MMR_1: F3F, IND: F3F
Field Name	**Bits**	**Default**	**Description**
MAX_LAT	7:0	0	<No Description>

ADAPTER_ID_W [W] 32-bits Access: 8/16/32			CFG: 4C
Field Name	**Bits**	**Default**	**Description**
SUBSYSTEM_VENDOR_ID	15:0	0	<No Description>
SUBSYSTEM_ID	31:16	0	<No Description>

CAPABILITIES_ID [R] 32-bits Access: 8/16/32			CFG: 50, MMR: F50, MMR_1: F50, IND: F50
Field Name	**Bits**	**Default**	**Description**
CAP_ID	7:0	2	<No Description>
NEXT_PTR	15:8	5c	<No Description>
AGP_MINOR	19:16	0	<No Description>
AGP_MAJOR	23:20	1	<No Description>
(reserved)	31:24		

PWR_MNGMT_CNTL_STATUS [RW] 32-bits Access: 8/16/32			CFG: 60, MMR: F60 [R], MMR_1: F60 [R], IND: F60 [R]
Field Name	**Bits**	**Default**	**Description**
POWER_STATE	1:0	0	<No Description>
(reserved)	31:2		

CONFIG_CNTL [RW] 32-bits Access: 8/16/32			MMR: E0, MMR_1: E0, IOR: E0
Field Name	**Bits**	**Default**	**Description**
APER_0_ENDIAN	1:0	0	<No Description> 0 = Little endian: (no swapping) 1 = Big endian: 16 bpp swapping 2 = Big endian: 32 bpp swapping
APER_1_ENDIAN	3:2	0	<No Description> 0 = Little endian: (no swapping) 1 = Big endian: 16 bpp swapping 2 = Big endian: 32 bpp swapping
APER_REG_ENDIAN	4	0	<No Description> 0 = Little endian: (no swapping) 1 = Big endian: 32 bpp swapping
(reserved)	7:5		
CFG_VGA_RAM_EN	8	0	<No Description> 0 = Disable 1 = Enable
CFG_VGA_IO_DIS	9	0	<No Description> 0 = VGA I/O decode enabled if VGA_DISABLE@CONFIG_XSTRAP=0 1 = VGA I/O decode disabled
(reserved)	15:10		
CFG_ATI_REV_ID (R)	19:16	0	<No Description>
(reserved)	31:20		

CONFIG_XSTRAP [RW] 32-bits Access: 8/16/32			MMR: E4 MMR_1: E4, IOR: E4, IND: E4
Field Name	**Bits**	**Default**	**Description**
VGA_DISABLE (R)	0	0	<No Description>
BUS_CLK_SEL (R)	1	0	<No Description>
IDSEL (R)	2	0	<No Description>
ENINTB (R)	3	0	<No Description>

(Continued)

CONFIG_XSTRAP [RW] 32-bits Access: 8/16/32			MMR: E4 MMR_1: E4, IOR: E4, IND: E4
Field Name	**Bits**	**Default**	**Description**
BUSTYPE (R)	5:4	0	<No Description>
AGPSKEW	7:6	0	<No Description>
X1CLK_SKEW	9:8	0	<No Description>
FLASH_ROM (R)	10	0	<No Description>
LCDPE (R)	11	0	<No Description>
(reserved)	31:12		

CONFIG_BONDS [RW] 32-bits Access: 8/16/32			MMR: E8, MMR_1: E8, IOR: E8, IND: E8
Field Name	**Bits**	**Default**	**Description**
RSTRAP (R)	1:0	0	<No Description>
PKGTYPE (R)	2	0	<No Description>
CRIPPLEb (R)	3	0	<No Description>
STRSTb (R)	4	0	<No Description>
(reserved)	5		
AVCOGN (R)	6	0	<No Description>
(reserved)	7		
LCDPE_OVERRIDE	8	0	<No Description>
(reserved)	31:9		

CONFIG_MEMSIZE [RW] 32-bits Access: 8/16/32			MMR: F8, MMR_1: F8, IOR: F8, IND: F8
Field Name	**Bits**	**Default**	**Description**
CONFIG_MEMSIZE	25:0	0	Size of the frame buffer in bytes. Includes embedded memory if present. NOTE: Bits 20:0 of this field are hardwired to ZERO.
(reserved)	31:26		

Description:

Frame Buffer Size.

CONFIG_APER_0_BASE [R] 32-bits Access: 8/16/32			MMR: 100, MMR_1: 100, IND: 100
Field Name	**Bits**	**Default**	**Description**
(reserved)	26:0		
APER_0_BASE	31:27	0	Base address of image 0 of the linear aperture.

Description:

Linear Aperture 0 Base.

CONFIG_APER_1_BASE [RW] 32-bits Access: 8/16/32			MMR: 104, MMR_1: 104, IND: 104
Field Name	**Bits**	**Default**	**Description**
(reserved)	25:0		

CONFIG_APER_1_BASE [RW] 32-bits Access: 8/16/32			MMR: 104, MMR_1: 104, IND: 104
Field Name	**Bits**	**Default**	**Description**
APER_1_BASE	31:27	0	Base address of image 1 of the linear aperture. Both the first and second linear apertures function the same. The second aperture is mainly for use on PowerMac systems, where each aperture can have its bi-endian swapping set independently (see CONFIG_CNTL). NOTE: Bit 0 of this field is hardwired to ONE.

Description:

Linear Aperture 1 Base.

PCI_GART_PAGE [RW] 32-bits Access: 8/16/32			MMR: 17C, MMR_1: 17C, IND: 17C
Field Name	**Bits**	**Default**	**Description**
PCI_GART_DIS	0	1	<No Description> 0 = Enable 1 = Disable Note: PCI_GART_DIS =1 is required if AGP is used.
(reserved)	11:1		
PCI_GART_PAGE	31:12	0	This is a 32bit physical memory address to a 32KB table of page entries. The table has the following form: DWORD PhysPageNo[8192] When in PCI mode and Promo4 BusMastering is enabled, any reference to AGP offsets now uses this paging mechanism to reference physical memory. Given any 32bit physical 'AGP-offset' [24:12] is the pageIndex PhysPageNo[pageIndex] is the host memory physical page number. Actual referenced address is: PhysAddr[31:12]<-PhysPageNo PhysAddr[11:0]<-Offset[11:0] PhysAddr is then used as the address of a PCI busmastering read.

Description:

With PCI_GART_PAGE, you can get up to a 32MB continuous address space into PCI system memory via a scatter-gather mechanism.

CONFIG_APER_SIZE [R] 32-bits Access: 8/16/32			MMR: 108, MMR_1: 108, IND: 108
Field Name	**Bits**	**Default**	**Description**
APER_SIZE	26:0	4000000	Size of linear apertures (both 0 and 1). This includes both the frame buffer image and the AGP system memory image area. NOTE: Bits 25:0 of this field are hardwired to ZERO
(reserved)	31:27		

Description:

Linear Aperture Size.

CONFIG_REG_1_BASE [R] 32-bits Access: 8/16/32			MMR: 10C, MMR_1: 10C, IND: 10C
Field Name	**Bits**	**Default**	**Description**
(reserved)	12:0		
REG_1_BASE	31:13	0	Base address of register aperture 1. The base address of register aperture 0 is found in PCI configuration space. The first and second register apertures are identical. The second is intended for use in PowerMac systems, but functions in all systems. NOTE: Bit 0 of this field is hardwired to ONE.

Description:

Register Aperture 1 Base.

CONFIG_REG_APER_SIZE [R] 32-bits Access: 8/16/32			MMR: 110, MMR_1: 110, IND: 110
Field Name	**Bits**	**Default**	**Description**
REG_APER_SIZE	13:0	2000	Size in bytes of each of the register apertures (both 0 and 1). NOTE: Bits 12:0 of this field are hardwired to ZERO
(reserved)	31:14		

Description:

Register Aperture Size.

CONFIG_MEMSIZE_EMBEDDED [R] 32-bits Access: 8/16/32			MMR: 114, MMR_1: 114, IND: 114
Field Name	**Bits**	**Default**	**Description**
CONFIG_MEMSIZE_EMB	25:0	0	Reserved for future use.This will indicate the size in bytes of the on-chip portion of the frame buffer.
(reserved)	31:26		

Description:

Embedded Memory Size.

MM_INDEX [W] 32-bits Access: 8/16/32			MMR: 00, MMR_1: 00, IOR: 00
Field Name	**Bits**	**Default**	**Description**
MM_ADDR	26:0	0	<No Description> NOTE: Bits 1:0 of this field are hardwired to ZERO
(reserved)	30:27		
MM_APER	31	0	<No Description> 0 = Register Aperture 1 = Linear Aperture 0

MM_DATA [W] 32-bits Access: 8/16/32			MMR: 04, MMR_1: 04, IOR: 04
Field Name	**Bits**	**Default**	**Description**
MM_DATA	31:0	0	<No Description>

SURFACE_DELAY [RW] 32-bits Access: 32			MMR: B00, MMR_1: B00, IND: B00
Field Name	**Bits**	**Default**	**Description**
SURF_POW2_DELAY	3:0	3	<No Description>
SURF_NONPOW2_DELAY	7:4	5	<No Description>
SURF_TRANSLATION_DIS	8	1	<No Description> 0 = Enable 1 = Disable
(reserved)	31:9		

SURFACE0_LOWER_BOUND [RW] 32-bits Access: 32			MMR: B04, MMR_1: B04, IND: B04
Field Name	**Bits**	**Default**	**Description**
SURF0_LOWER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE1_LOWER_BOUND [RW] 32-bits Access: 32			MMR: B14, MMR_1: B14, IND: B14
Field Name	**Bits**	**Default**	**Description**
SURF1_LOWER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE2_LOWER_BOUND [RW] 32-bits Access: 32			MMR: B24, MMR_1: B24, IND: B24
Field Name	**Bits**	**Default**	**Description**
SURF2_LOWER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.

SURFACE2_LOWER_BOUND [RW] 32-bits Access: 32			MMR: B24, MMR_1: B24, IND: B24
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:26		

SURFACE3_LOWER_BOUND [RW] 32-bits Access: 32			MMR: B34, MMR_1: B34, IND: B34
Field Name	**Bits**	**Default**	**Description**
SURF3_LOWER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE0_UPPER_BOUND [RW] 32-bits Access: 32			MMR: B08, MMR_1: B08, IND: B08
Field Name	**Bits**	**Default**	**Description**
SURF0_UPPER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE1_UPPER_BOUND [RW] 32-bits Access: 32			MMR: B18, MMR_1: B18, IND: B18
Field Name	**Bits**	**Default**	**Description**
SURF1_UPPER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE2_UPPER_BOUND [RW] 32-bits Access: 32			MMR: B28, MMR_1: B28, IND: B28
Field Name	**Bits**	**Default**	**Description**
SURF2_UPPER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE3_UPPER_BOUND [RW] 32-bits Access: 32			MMR: B38, MMR_1: B38, IND: B38
Field Name	**Bits**	**Default**	**Description**
SURF3_UPPER	25:0	0	<No Description> NOTE: Bits 5:0 of this field are hardwired to ZERO.
(reserved)	31:26		

SURFACE0_INFO [RW] 32-bits Access: 32			MMR: B0C, MMR_1: B0C, IND: B0C
Field Name	**Bits**	**Default**	**Description**
SURF0_PITCHSEL	4:0	0	<No Description> 0 = Linear/No translation 1 = 64 bytes 2 = 128 bytes 3 = 256 bytes 4 = 512 bytes 5 = 1024 bytes 6 = 2048 bytes 7 = 4096 bytes 8 = 640 bytes 9 = 1280 bytes 10 = 2560 bytes 11 = 5120 bytes 12 = 1600 bytes 13 = 3200 bytes 14 = 6400 bytes 15 = 832 bytes 16 = 1664 bytes 17 = 3328 bytes 18 = 1920 bytes 19 = 3840 bytes

SURFACE0_INFO [RW] 32-bits Access: 32			MMR: B0C, MMR_1: B0C, IND: B0C
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:5		

SURFACE1_INFO [RW] 32-bits Access: 32			MMR: B1C, MMR_1: B1C, IND: B1C
Field Name	**Bits**	**Default**	**Description**
SURF1_PITCHSEL	4:0	0	<No Description> 0 = Linear/No translation 1 = 64 bytes 2 = 128 bytes 3 = 256 bytes 4 = 512 bytes 5 = 1024 bytes 6 = 2048 bytes 7 = 4096 bytes 8 = 640 bytes 9 = 1280 bytes 10 = 2560 bytes 11 = 5120 bytes 12 = 1600 bytes 13 = 3200 bytes 14 = 6400 bytes 15 = 832 bytes 16 = 1664 bytes 17 = 3328 bytes 18 = 1920 bytes 19 = 3840 bytes
(reserved)	31:5		

SURFACE2_INFO [RW] 32-bits Access: 32			MMR: B2C, MMR_1: B2C, IND: B2C
Field Name	**Bits**	**Default**	**Description**
SURF2_PITCHSEL	4:0	0	<No Description> 0 = Linear/No translation 1 = 64 bytes 2 = 128 bytes 3 = 256 bytes 4 = 512 bytes 5 = 1024 bytes 6 = 2048 bytes 7 = 4096 bytes 8 = 640 bytes 9 = 1280 bytes 10 = 2560 bytes 11 = 5120 bytes 12 = 1600 bytes 13 = 3200 bytes 14 = 6400 bytes 15 = 832 bytes 16 = 1664 bytes 17 = 3328 bytes 18 = 1920 bytes 19 = 3840 bytes
(reserved)	31:5		

SURFACE3_INFO [RW] 32-bits Access: 32			MMR: B3C, MMR_1: B3C, IND: B3C
Field Name	**Bits**	**Default**	**Description**
SURF3_PITCHSEL	4:0	0	<No Description> 0 = Linear/No translation 1 = 64 bytes 2 = 128 bytes 3 = 256 bytes 4 = 512 bytes 5 = 1024 bytes 6 = 2048 bytes 7 = 4096 bytes 8 = 640 bytes 9 = 1280 bytes 10 = 2560 bytes 11 = 5120 bytes 12 = 1600 bytes 13 = 3200 bytes 14 = 6400 bytes 15 = 832 bytes 16 = 1664 bytes 17 = 3328 bytes 18 = 1920 bytes 19 = 3840 bytes
(reserved)	31:5		

4.2 AGP Registers

AGP_STATUS [R] 32-bits Access: 8/16/32			CFG: 54, MMR: F54, MMR_1: F54, IND: F54
Field Name	**Bits**	**Default**	**Description**
RATE1X	0	1	<No Description>
RATE2X	1	1	<No Description>
(reserved)	8:2		
SBA	9	1	<No Description>
(reserved)	23:10		
RQ	31:24	1f	<No Description>

AGP_COMMAND [RW] 32-bits Access: 8/16/32			CFG: 58, MMR: F58 [R], MMR_1: F58 [R], IND: F58 [R]
Field Name	**Bits**	**Default**	**Description**
DATA_RATE	1:0	0	<No Description>
(reserved)	7:2		
AGP_EN	8	0	<No Description> 0 = Disable 1 = Enable
SBA_EN	9	1	<No Description> 0 = Disable 1 = Enable
(reserved)	23:10		
RQ_DEPTH	31:24	0	<No Description>

AGP_BASE [RW] 32-bits Access: 8/16/32			MMR: 170, MMR_1: 170, IND: 170
Field Name	**Bits**	**Default**	**Description**
AGP_BASE_ADDR	31:0	0	AGP Base Address: NOTE: Bits 21:0 of this field are hardwired to ZERO

AGP_CNTL [RW] 32-bits Access: 8/16/32			MMR: 174, MMR_1: 174, IND: 174
Field Name	**Bits**	**Default**	**Description**
AGP_APER_SIZE	5:0	0	AGP aperture size 0 = 000000 = 256MB 32 = 100000 = 128MB 48 = 110000 = 64MB 56 = 111000 = 32MB 60 = 111100 = 16MB 62 = 111110 = 8MB 63 = 111111 = 4MB = 63
(reserved)	7:6		
MAX_IDLE_CLK	15:8	0	This is the number of clocks (MAX_IDLE_CLK x 32) that the AGP block will wait before stopping the generation of the 2X sideband strobe after it no longer has a request to service.
HOLD_RD_FIFO	16	0	<No Description> 0 = Normal Operation 1 = Hold Fifo
HOLD_RQ_FIFO	17	0	<No Description> 0 = Normal Operation 1 = Hold Fifo
HOLD_WR_FIFO	18	0	<No Description> 0 = Normal Operation 1 = Hold Fifo
AGP_OCTWD_ALGN	19	0	<No Description> 0 = Normal QW Alignment 1 = Use OCTWD Alignment

(Continued)

AGP_CNTL [RW] 32-bits Access: 8/16/32			MMR: 174, MMR_1: 174, IND: 174
Field Name	**Bits**	**Default**	**Description**
AGP_TG_EXTSENSE	20	1	<No Description> 0 = Short Fifo Sensing 1 = Extended Fifo Sensing
AGP_RQ_EXTSENSE	21	1	<No Description> 0 = Short Fifo Sensing 1 = Extended Fifo Sensing
AGP_RD_EXTSENSE	22	1	<No Description> 0 = Short Fifo Sensing 1 = Extended Fifo Sensing
AGP_WR_EXTSENSE	23	1	<No Description> 0 = Short Fifo Sensing 1 = Extended Fifo Sensing
RQ_ARB_MAX_CNT	27:24	0	<No Description>
RQ_ARB_IDLE_CNT	29:28	0	<No Description>
(reserved)	31:30		

AGP_APER_OFFSET [R] 32-bits Access: 8/16/32			MMR: 178, MMR_1: 178, IND: 178
Field Name	**Bits**	**Default**	**Description**
AGP_APER_OFFSET	25:0	2000000	<No Description> NOTE: Bits 24:0 of this field are hardwired to ZERO
(reserved)	31:26		

4.3 Bus Control Register

BUS_CNTL [RW] 32-bits Access: 8/16/32			MMR: 30 MMR_1: 30 IOR: 30
Field Name	**Bits**	**Default**	**Description**
BUS_DBL_RESYNC	0	1	<No Description> 0 = Normal 1 = Add extra re-synchronizing clock
BUS_MSTR_RESET (W)	1	0	<No Description> 0 = Normal 1 = Reset
BUS_FLUSH_BUF (W)	2	0	<No Description> 0 = Normal 1 = Flush
BUS_STOP_REQ_DIS	3	0	<No Description> 0 = Normal 1 = Disable
BUS_QUE_ACTIVE_DIS	4	0	<No Description> 0 = Normal 1 = Disable
BUS_ROTATION_DIS	5	0	<No Description> 0 = Enable 1 = Disable
BUS_MASTER_DIS	6	1	<No Description> 0 = Enable 1 = Disable
BIOS_ROM_WRT_EN	7	0	<No Description> 0 = Disable 1 = Enable
BUS_OS_READ_REQ	11:8	f	<No Description>
BIOS_DIS_ROM	12	0	<No Description> 0 = Enable 1 = Disable
BUS_PCI_READ_RETRY_EN	13	0	<No Description> 0 = Normal 1 = Enable
BUS_AGP_AD_STEPPING_EN	14	1	<No Description> 0 = No stepping in AGP 1 = AD Stepping in AGP and PCI

(Continued)

BUS_CNTL			MMR: 30 MMR_1: 30
[RW] 32-bits Access: 8/16/32			IOR: 30
Field Name	**Bits**	**Default**	**Description**
BUS_PCI_WRT_RETRY_EN	15	0	<No Description> 0 = Normal 1 = Enable
BUS_RETRY_WS	19:16	f	<No Description>
BUS_MSTR_RD_MULT	20	0	<No Description> 0 = Read line 1 = Read multiple
BUS_MSTR_RD_LINE	21	0	<No Description> 0 = Read multiple 1 = Read line
BUS_SUSPEND	22	0	<No Description> 0 = Resume BM transfer 1 = Suspend BM transfer
LAT_16X	23	0	<No Description> 0 = 1X 1 = 16X
BUS_RD_DISCARD_EN	24	0	<No Description> 0 = Disable 1 = Enable
BUS_RD_ABORT_EN	25	0	<No Description> 0 = Disable 1 = Enable
BUS_MSTR_WS	26	0	<No Description> 0 = 8 wait states 1 = 32 wait states
BUS_PARKING_DIS	27	1	<No Description> 0 = Enable 1 = Disable
BUS_MSTR_DISCONNECT_EN	28	0	<No Description> 0 = Disable 1 = Enable
BUS_WRT_BURST	29	0	<No Description> 0 = Disable 1 = Enable

(Continued)

BUS_CNTL			**MMR: 30 MMR_1: 30**
[RW] 32-bits Access: 8/16/32			**IOR: 30**
Field Name	**Bits**	**Default**	**Description**
BUS_READ_BURST	30	0	<No Description> 0 = Disable 1 = Enable
BUS_RDY_READ_DLY	31	1	<No Description> 0 = no RDY delay 1 = RDY delayed 1 memory clk

This page intentionally left blank.

Chapter 5
VGA Registers

5.1 General VGA Status and Configuration Registers

GENMO_WT [W] 8-bits Access: 8/16/32			VGA_IO: 3C2
Field Name	**Bits**	**Default**	**Description**
GENMO_MONO_ ADDRESS_B	0	0	Emulation Addressing Mode (write) 0=Monochrome 1=Color/Graphic
VGA_RAM_EN	1	0	Enables/Disables CPU access to video RAM (write) 0=Disable 1=Enable
VGA_CKSEL	3:2	0	Selects pixel clock frequency to use. 0=25.1744MHz (640 Pels) 1=28.3212MHz (720 Pels) 2=Reserved 3=Reserved
(reserved)	4		
ODD_EVEN_MD_PGSEL	5	0	This bit is used in odd/even display modes (A/N modes: 0, 1, 2, 3, and 7). This bit is ignored when either bit GRA06[1] or SEQ4[3] are enabled. Used to determine if the VGA aperture maps into the lower (even) or upper (odd) page of memory. 0=Selects odd (high) memory locations 1=Selects even (low) memory locations
VGA_VSYNC_POL	6	0	Determines polarity of horizontal sync (HSYNC) for VGA modes. 0 = HSYNC pulse active high 1 = HSYNC pulse active low The convention of VGA is to use active low VSYNC for 400 (and 200) and 480 line modes. Active high is normally used for 350 line modes.

(Continued)

GENMO_WT [W] 8-bits Access: 8/16/32			VGA_IO: 3C2
Field Name	**Bits**	**Default**	**Description**
VGA_HSYNC_POL	7	0	Determines polarity of vertical sync (VSYNC) for VGA modes. 0 = VSYNC pulse active high 1 = VSYNC pulse active low The convention of VGA is to use active high VSYNC for 400 (and 200) line modes. Active low is normally used for 350 and 480 line modes.

Description:

Miscellaneous output register (write only).

GENMO_RD [R] 8-bits Access: 8/16/32			VGA_IO: 3CC
Field Name	**Bits**	**Default**	**Description**
GENMO_MONO_ ADDRESS_B	0	0	Emulation addressing mode (read). 0=Monochrome 1=Color/Graphic
VGA_RAM_EN	1	0	Enables/Disables CPU access to video RAM (read). 0=Disable 1=Enable
VGA_CKSEL	3:2	0	Selects pixel clock frequency to use. 0=25.1744MHz (640 Pels) 1=28.3212MHz (720 Pels) 2=Reserved 3=Reserved
(reserved)	4		

(Continued)

GENMO_RD [R] 8-bits Access: 8/16/32			VGA_IO: 3CC
Field Name	**Bits**	**Default**	**Description**
ODD_EVEN_MD_PGSEL	5	0	This bit is used in odd/even display modes (A/N modes: 0, 1, 2, 3, and 7). This bit is ignored when either bit GRA06[1] or SEQ4[3] are enabled.Used to determine if the VGA aperture maps into the lower (even) or upper (odd) page of memory. 0=Selects odd (high) memory locations 1=Selects even (low) memory locations
VGA_VSYNC_POL	6	0	Determines polarity of horizontal sync (HSYNC) for VGA modes. 0 = HSYNC pulse active high 1 = HSYNC pulse active low The convention of VGA is to use active low VSYNC for 400 (and 200) and 480 line modes. Active high is normally used for 350 line modes.
VGA_HSYNC_POL	7	0	Determines polarity of vertical sync (VSYNC) for VGA modes. 0 = VSYNC pulse active high 1 = VSYNC pulse active low The convention of VGA is to use active high VSYNC for 400 (and 200) line modes. Active low is normally used for 350 and 480 line modes.

Description:

Miscellaneous output register (read only).

GENFC_RD [R] 8-bits Access: 8/16/32			VGA_IO: 3CA
Field Name	**Bits**	**Default**	**Description**
(reserved)	2:0		
VSYNC_SEL	3	0	Vertical sync select (read). 0=Normal vertical sync 1=Sync is 'vertical sync' ORed with 'vertical display enable'
(reserved)	7:4		

Description:

Feature control register (read only).

GENFC_WT [W] 8-bits Access: 8/16/32			VGA_IO: 3BA, VGA_IO: 3DA
Field Name	**Bits**	**Default**	**Description**
(reserved)	2:0		
VSYNC_SEL	3	0	Vertical sync select (write). 0=Normal vertical sync 1=Sync is 'vertical sync' ORed with 'vertical display enable'
(reserved)	7:4		

Description:

Feature control register (write only).

GENS0 [R] 8-bits Access: 8/16/32			VGA_IO: 3C2
Field Name	**Bits**	**Default**	**Description**
(reserved)	3:0		
SENSE_SWITCH	4	0	DAC comparator read back. Used for monitor detection. Mirror of DAC_CMP_OUTPUT@DAC_CNTL.
(reserved)	6:5		
CRT_INTR	7	0	CRT Interrupt: 0=Vertical retrace interrupt is cleared 1=Vertical retrace interrupt is pending

Description:

Input status 0 register.

GENS1 [R] 8-bits Access: 8/16/32			VGA_IO: 3BA, VGA_IO: 3DA
Field Name	**Bits**	**Default**	**Description**
NO_DIPLAY	0	0	Display enable: 0=Enable 1=Disable
(reserved)	2:1		
VGA_VSTATUS	3	0	Vertical Retrace Status 0=VRetraceInactive 1=VRetraceActive
PIXEL_READ_BACK	5:4	0	Diagnostic bits 0, 1 respectively. This two bits are connected to two of the eight color outputs (P7:P0) of the attribute controller. Connections are controlled by ATTR12(5,4) as follows: 0=P2,P0 1=P5,P4 2=P3,P1 3=P7,P6
(reserved)	7:6		

Description:

Input status 1 register.

GENENB [R] 8-bits Access: 8/16/32			VGA_IO: 3C3
Field Name	**Bits**	**Default**	**Description**
BLK_IO_BASE	7:0	0	Readback of block I/O aperture base offset. Mirror of the PCI configuration space register. Used here so software can find the apertures if they are relocated by the OS.

Description:

Block I/O Base.

5.2 VGA DAC Registers

DAC_DATA			VGA_IO: 3C9
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
DAC_DATA	7:0	0	VGA Palette (DAC) Data. Use DAC_R_INDEX and DAC_W_INDEX to set read or write mode, and entry to access. Access order is Red, Green, Blue, and then auto-increment occurs to next entry. DAC_8BIT_EN controls whether 6 or 8 bit access.

Description:

VGA Palette (DAC) Data.

DAC_MASK			VGA_IO: 3C6
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
DAC_MASK	7:0	ff	Masks off usage of individual palette index bits before pixel index is looked-up in the palette. 0 = do not use this bit of the index 1 = use this bit of the index Only has an effect in VGA emulation modes (CRTC_EXT_DISP_EN=0), not for VESA modes or extended display modes.

Description:

Palette index mask for VGA emulation modes.

DAC_R_INDEX [RW] 8-bits Access: 8/16/32			VGA_IO: 3C7
Field Name	**Bits**	**Default**	**Description**
DAC_R_INDEX	7:0	0	Write: Sets the index for a palette (DAC) read operation. Index auto-increments after every third read of DAC_DATA. Read: Indicates if palette in read or write mode. 0 = Palette in write mode (DAC_W_INDEX last written). 3 = Palette in read mode (DAC_R_INDEX last written). Also see DAC_W_INDEX.

Description:

Palette (DAC) Read Index

DAC_W_INDEX [RW] 8-bits Access: 8/16/32			VGA_IO: 3C8
Field Name	**Bits**	**Default**	**Description**
DAC_W_INDEX	7:0	0	Sets the index for a palette (DAC) write operation. Index auto-increments after every third write of DAC_DATA. Also see DAC_R_INDEX.

Description:

Palette (DAC) Write Index.

5.3 VGA Sequencer Registers

SEQ8_IDX			VGA_IO: 3C4
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
SEQ_IDX	2:0	0	This index points to one of the sequencer registers (SEQ_ at I/O port address 3C5, for the next SEQ read/write operation.
(reserved)	7:3		

SEQ8_DATA			VGA_IO: 3C5
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
SEQ_DATA	7:0	0	<No Description>

SEQ00			SEQ: 00
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
SEQ_RST0B	0	1	Synchronous reset bit 0: 0=Follows SEQ_RST1B 1=Sequencer runs unless SEQ_RST1B=0
SEQ_RST1B	1	1	Synchronous reset bit 1: 0=Disable character clock, display requests, and H/V syncs 1=Sequencer runs unless SEQ_RST0B=0
(reserved)	7:2		

Description:

Reset register.

SEQ01 [RW] 8-bits Access: 8/16/32			SEQ: 01
Field Name	**Bits**	**Default**	**Description**
SEQ_DOT8	0	1	8/9 Dot Clocks (Modes 1, 2, 3, and 7 use 9-dot characters. To change bit 0, GENVS(0) must be logical 0). 0=9 dot char clock. Modes 0, 1, 2, 3 & 7 1=8 dot char clock.
(reserved)	1		
SEQ_SHIFT2	2	0	Shift load bits. 0=Load video serializer every clock, if SEQ_SHIFT4=0 1=Load video serializer every other clock, if SEQ_SHIFT4=0
SEQ_PCLKBY2	3	0	Dot Clock (typically, 320 and 360 horizontal modes use divide-by-2 to provide 40 column displays. To change this bit SEQ00[0:0] must be first set to zero.)). 0=Dot clock is normal 1=Dot clock is divided by 2
SEQ_SHIFT4	4	0	Shift load bits. 0=SEQ_SHIFT2 determines serializer loading 1=Load video serializer every fourth clock. Ignore SEQ_SHIFT2.
SEQ_MAXBW	5	1	Screen off: 0=Normal. Screen on 1=Sreen off and blanked. CPU has uninterrupted access to frame buffer
(reserved)	7:6		

Description:

Clock mode register.

SEQ02 [RW] 8-bits Access: 8/16/32			SEQ: 02
Field Name	**Bits**	**Default**	**Description**
SEQ_MAP0_EN	0	0	Enable map 0: 0=Disable write to memory map 0 1=Enable write to memory map 0
SEQ_MAP1_EN	1	0	Enable map 1: 0=Disable write to memory map 1 1=Enable write to memory map 1
SEQ_MAP2_EN	2	0	Enable map 2: 0=Disable write to memory map 2 1=Enable write to memory map 2
SEQ_MAP3_EN	3	0	Enable map 3: 0=Disable write to memory map 3 1=Enable write to memory map 3
(reserved)	7:4		

Description:

Map mask register.

SEQ03 [RW] 8-bits Access: 8/16/32			SEQ: 03
Field Name	**Bits**	**Default**	**Description**
SEQ_FONT_B1	0	0	Character Map Select B Bit 1
SEQ_FONT_B2	1	0	Character Map Select B Bit 2
SEQ_FONT_A1	2	0	Character Map Select A Bit 1
SEQ_FONT_A2	3	0	Character Map Select A Bit 2
SEQ_FONT_B0	4	0	Character Map Select B Bit 0
SEQ_FONT_A0	5	0	Character Map Select A Bit 0
(reserved)	7:6		

Description:

Character map select register.

SEQ04 [RW] 8-bits Access: 8/16/32			SEQ: 04
Field Name	**Bits**	**Default**	**Description**
(reserved)	0		
SEQ_256K	1	0	Extended memory - 1 indicates 256 KB of video memory is present. It also enables the character map selection in SEQ03. 0 = 64KB memory present. Has no effect since 256KB always available 1 = 256KB memory present.
SEQ_ODDEVEN	2	0	Odd/Even: 0 = Even CPU address (A0=0) accesses maps 0 and 2. Odd address accesses maps 1 and 3. 1 = Enables sequential access to maps for odd/even modes. SEQ02 (Map Mask) selects which maps are used.
SEQ_CHAIN	3	0	Chain (when logical 1, it takes priority over off/even mode bits SEQ04[2] and GRA05[4]. Unlike odd/even mode, SEQ04[2] is the only bit used to enable chain mode (double odd/even). Chain does not affect CRTC access to video memory. Odd/even bit SEQ04[2] should be the opposite of GRA05[4]. 0 = Enables sequential access to maps. SEQ02 (Map Mask) selects which maps are used. 1 = For 256 color modes. Map select by CPU address bits A1:A0.
(reserved)	7:4		

Description:

Memory mode register.

5.4 VGA CRT Registers

CRTC8_IDX [RW] 8-bits Access: 8/16/32			VGA_IO: 3B4, VGA_IO: 3D4
Field Name	**Bits**	**Default**	**Description**
VCRTC_IDX	5:0	0	This index points to one of the internal registers of the CRT controller (CRTC) at address 3?5, for the next CRTC read/write operation.
(reserved)	7:6		

Description:

CRTC index register.

CRTC8_DATA [RW] 8-bits Access: 8/16/32			VGA_IO: 3B5, VGA_IO: 3D5
Field Name	**Bits**	**Default**	**Description**
VCRTC_DATA	7:0	0	<No Description>

CRT00 [RW] 8-bits Access: 8/16/32			CRT: 00
Field Name	**Bits**	**Default**	**Description**
H_TOTAL	7:0	0	These bits define the active horizontal display in a scan line, including the retrace period. The value is five less than the total number of displayed characters in a scan line.

Description:

Horizontal total register.

CRT01 [RW] 8-bits Access: 8/16/32			CRT: 01
Field Name	**Bits**	**Default**	**Description**
H_DISP_END	7:0	0	These bits define the active horizontal display in a scan line. The value is one less than the total number of displayed characters in a scan line.

Description:

Horizontal display enable end register.

CRT02 [RW] 8-bits Access: 8/16/32			CRT: 02
Field Name	**Bits**	**Default**	**Description**
H_BLANK_START	7:0	0	These bits define the horizontal character count that represents the character count in the active display area plus the right border. In other words, the count is from the start of active display to the start of triggering of the H blanking pulse.

Description:

Start horizontal blanking register.

CRT03 [RW] 8-bits Access: 8/16/32			CRT: 03
Field Name	**Bits**	**Default**	**Description**
H_BLANK_END	4:0	0	H blanking bits 4-0 respectively. These are the five low-order bits (of six bits in total) of horizontal character count for triggering the end of the horizontal blanking pulse.
H_DE_SKEW	6:5	0	Display-enable skew: 0=0Skew 1=1Skew 2=2Skew 3=3Skew
CR10CR11_R_DIS_B	7	0	Compatibility Read: 0=WrtOnlyToCRT10-11 1=WrtRdToCRT10-11

Description:

End horizontal blanking register.

CRT04 [RW] 8-bits Access: 8/16/32			CRT: 04
Field Name	**Bits**	**Default**	**Description**
H_SYNC_START	7:0	0	These bits define the horizontal character count at which the horizontal retrace pulse becomes active.

Description:

Start horizontal retrace register.

CRT05 [RW] 8-bits Access: 8/16/32			CRT: 05
Field Name	**Bits**	**Default**	**Description**
H_SYNC_END	4:0	0	H Retrace Bits (these are the 5-bit result from the sum of CRT0 plus the width of the horizontal retrace pulse, in character clock units).
H_SYNC_SKEW	6:5	0	H Retrace Delay bits (these two bits skew the horizontal retrace pulse).
H_BLANK_END_B5	7	0	H blocking end bit 5 (this is the bit of the 6-bit character count for the H blanking end pulse). The other five low-order bits are CRT03[4:0].

Description:

End horizontal retrace register.

CRT06 [RW] 8-bits Access: 8/16/32			CRT: 06
Field Name	**Bits**	**Default**	**Description**
V_TOTAL	7:0	0	These are the eight low-order bits of the 10-bit vertical total register. The 2 high-order bits are CRT07[5:0] in the CRTC overflow register. The value of this register represents the total number of H raster scans plus vertical retrace (active display, blanking), minus two scan lines.

Description:

Vertical total register.

CRT07 [RW] 8-bits Access: 8/16/32			CRT: 07
Field Name	**Bits**	**Default**	**Description**
V_TOTAL_B8	0	0	V Total Bit 8 (CRT06). Bit 8 of 10 bit vertical count for V Total. For functional description see CRT06 register.
V_DISP_END_B8	1	0	End V Display Bit 8 (CRT12). Bit 8 of 10-bit vertical count for V Display enable. For functional description see CRT12 register.
V_SYNC_START_B8	2	0	Start V Retrace Bit 8 (CRT10). Bit 8 of 10-bit vertical count for V Retrace start. For functional description see CRT10 register.
V_BLANK_START_B8	3	0	Start V Blanking Bit 8 (CRT15). Bit 8 of the 10-bit vertical count for V Blanking start. For functional description see CRT15 register.
LINE_CMP_B8	4	0	Line compare bit 8 (CRT18). Bit 8 of the 10-bit vertical count for line compare. For functional description see CRT18 register.
V_TOTAL_B9	5	0	V Total Bit 9 (CRT06). Bit 9 of 10-bit vertical count for V Total. For functional description see CRT06 register.
V_DISP_END_B9	6	0	End V Display Bit 9 (CRT12). Bit 9 of 10-bit vertical count for V Display enable end (for functional description see CRT12 register).
V_SYNC_START_B9	7	0	Start V Retrace Bit (CRT10). Bit 9 of 10-bit vertical count for V Retrace start. For functional description see CRT10 register.

Description:

CRTC overflow register.

CRT08 [RW] 8-bits Access: 8/16/32			CRT: 08
Field Name	**Bits**	**Default**	**Description**
ROW_SCAN_START	4:0	0	Preset row scan bit 4:0. This register is used for software-controlled vertical scrolling in text or graphics modes. The value specifies the first line to be scanned after a V retrace (in the next frame). Each H Retrace pulse increments the counter by 1, up to the maximum scan line value programmed by CRT09, then the counter is cleared.
BYTE_PAN	6:5	0	Byte panning control bits 1 and 0 (respectively). Bits 6 and 5 extend the capability of byte panning (shifting) by up to three characters (for description H_PEL Panning register ATTR13).
(reserved)	7		

Description:

Preset row scan register.

CRT09 [RW] 8-bits Access: 8/16/32			CRT: 09
Field Name	**Bits**	**Default**	**Description**
MAX_ROW_SCAN	4:0	0	Maximum scan line bits. These bits define a value that is the actual number of scan line per character minus 1.
V_BLANK_START_B9	5	0	Start V Blanking bit 9 (CRT15). Bit 9 of 10-bit vertical count for line compare. For functional description see CRT18 register.
LINE_CMP_B9	6	0	Line Compare Bit 9 (CRT18). Bit 9 of 10-bit vertical count for line compare. For functional description see CRT18 register.
DOUBLE_CHAR_HEIGHT	7	0	200/400 line scan. NOTE H/V display and blanking timings etc. (in CRT00-CRT06 registers) are not affected. 0=200LineScan 1=400LineScan

Description:

Maximum scan line register.

CRT0A [RW] 8-bits Access: 8/16/32			CRT: 0A
Field Name	**Bits**	**Default**	**Description**
CURSOR_START	4:0	0	Cursor start bits 4:0 (respectively). These bits define a value that is the starting scan line (on a character row) for the line cursor. The 5-bit value is equal to the actual number minus one. This value is used together with the Cursor End Bits CRT0B[4:0] to determine the height of the cursor. The cursor height in VGA does not wrap around (as in EGA) and is actually absent when the 'end' value is less than the 'start' value. In EGA when the 'end' value is less, the cursor is a full block cursor the same height as the character cell.
CURSOR_DISABLE	5	0	Cursor On/Off. 0=On 1=Off
(reserved)	7:6		

Description:

Cursor start register.

CRT0B [RW] 8-bits Access: 8/16/32			CRT: 0B
Field Name	**Bits**	**Default**	**Description**
CURSOR_END	4:0	0	Cursor End Bits 4-0, respectively.- These bits define the ending scan row (on a character line) for the line cursor. In EGA, this 5-bit value is equal to the actual number of lines plus one. The cursor height in VGA does not wrap around (as in EGA) and is actually absent when the 'end' value is less than the 'start' value. In EGA when the 'end' value is less, the cursor is a full block cursor the same height as the character cell.
CURSOR_SKEW	6:5	0	Cursor Skew Bits 1 and 0, respectively.- These bits define the number of characters the cursor is to be shifted to the right (skewed) from the character pointed at by the cursor location (registers CRT0E and CRT0F), in VGA mode. Skew values when in EGA mode are enclosed in brackets.
(reserved)	7		

Description:

Cursor end register.

CRT0C [RW] 8-bits Access: 8/16/32			CRT: 0C
Field Name	**Bits**	**Default**	**Description**
DISP_START	7:0	0	SA bits 15:8-These are the eight high-order bits of the 16-bit display buffer start location. The low order bits are contained in CRT0D.-In split screen mode, CRT0C = CRT0D point to the starting location of screen A (top half.) The starting address for screen B is always zero.

Description:

Start address (high byte) register.

CRT0D [RW] 8-bits Access: 8/16/32			CRT: 0D
Field Name	**Bits**	**Default**	**Description**
DISP_START	7:0	0	SA bits 7:0- These are the eight low-order bits of the 16-bit display buffer start location. The high-order bits are contained in CRT0C. - In split screen mode, CRT0C + CRT0D points to the starting location _of screen A (top half.) The starting address for screen B is always zero.

Description:

Start address (low byte) register.

CRT0E [RW] 8-bits Access: 8/16/32			CRT: 0E
Field Name	**Bits**	**Default**	**Description**
CURSOR_LOC_HI	7:0	0	CA bits 15:8- These are the eight high-order bits of the 16 bit cursor start address. The low-order CA bits are contained in CRT0F. This address is relative to the start of physical display memory address pointed to by CRT0C + CRT0D. In other words, if CRT0C + CRT0D is changed, the cursor still pints to the same character as before.

Description:

Cursor location (high byte) register.

CRT0F [RW] 8-bits Access: 8/16/32			CRT: 0F
Field Name	**Bits**	**Default**	**Description**
CURSOR_LOC_LO	7:0	0	CA bits These are the eight low-order bits of the 16 bit cursor start address. The high-order CA bits are contained in CRT0E. This address is relative to the start of physical display memory address pointed to by CRT0C + CRT0D. In other words, if CRT0C + T0D is changed, the cursor still points to the same character as before.

Description:

Cursor location (low byte) register.

CRT10 [RW] 8-bits Access: 8/16/32			CRT: 10
Field Name	**Bits**	**Default**	**Description**
V_SYNC_START	7:0	0	These are the eight low-order bits of the 10-bit vertical retrace start count. The two high-order bits are CRT07[2:7], located in the CRTC overflow register. These bits define the horizontal scan count that triggers the V retrace pulse.

Description:

Start vertical retrace register.

CRT11 [RW] 8-bits Access: 8/16/32			CRT: 11
Field Name	**Bits**	**Default**	**Description**
V_SYNC_END	3:0	0	V Retrace End Bits 3-0- Bits CRT11[0:3] define the horizontal scan count that triggers the end of the V Retrace pulse.
V_INTR_CLR	4	0	V Retrace Interrupt Set: 0=VRetraceIntCleared 1=Not Cleared
V_INTR_EN	5	0	V Retrace Interrupt Disabled: 0=VRetraceIntEna 1=Disable
(reserved)	6		
C0T7_WR_ONLY	7	0	Write Protect (CRT00-CRT06). All register bits except CRTO7[4] are write protected. 0=EnaWrtToCRT00-07 1=C0T7B4WrtOnly

Description:

End vertical retrace register.

CRT12 [RW] 8-bits Access: 8/16/32			CRT: 12
Field Name	**Bits**	**Default**	**Description**
V_DISP_END	7:0	0	These are the eight low-order bits of the 10-bit register containing the horizontal scan count indicating where the active display on the screen should end. The high-order bits are CRT07 [1:6] in the CRT overflow register.

Description:

Vertical display enable end register.

CRT13 [RW] 8-bits Access: 8/16/32			CRT: 13
Field Name	**Bits**	**Default**	**Description**
DISP_PITCH	7:0	0	These bits define an offset value, equal to the logical line width of the screen (from the first character of the current line to the first character of the next line). Memory organization is dependent on the video mode.Bit CRT17[6] selects byte or word mode. Bit CRT14[6], which overrides the byte/word mode setting, selects Double-Word mode when it is logical one. The first character of the next line is specified by the start address (CRT0C + CRT0D) plus the offset. The offset for byte mode is 2x CRT13; for word mode, 4x; for double word mode 8x.

Description:

Define offset register.

CRT14 [RW] 8-bits Access: 8/16/32			CRT: 14
Field Name	**Bits**	**Default**	**Description**
UNDRLN_LOC	4:0	0	H Row Scan Bits. These bits define the horizontal scan row, from the top of the character line, that should be used for underlining. The 5-bit value is equal to the actual number minus one.
ADDR_CNT_BY4	5	0	Count-by-4:_ 0 = Char. Clock 1 = CountBy4
DOUBLE_WORD	6	0	Double-Word Mode: 0 = Disable 1 = DoubleWordMdEna
(reserved)	7		

Description:

Underline location register.

CRT15 [RW] 8-bits Access: 8/16/32			CRT: 15
Field Name	**Bits**	**Default**	**Description**
V_BLANK_START	7:0	0	These are the eight low-order bits of the 10-bit vertical blanking start register. Bit 9 is CRT09[5]; bit 8 is CRT07[3]. The 10 bits specify the starting location of the vertical blanking pulse, in units of horizontal scan lines. The value is equal to the actual number of displayed lines minus one.

Description:

Start vertical blanking register.

CRT16 [RW] 8-bits Access: 8/16/32			CRT: 16
Field Name	**Bits**	**Default**	**Description**
V_BLANK_END	7:0	0	These bits define the point at which to trigger the end of the vertical blanking pulse. The location is specified in units of horizontal scan lines. The value to be stored in this register is the seven low-order bits of the sum of 'pulse width count' plus the content of Start Vertical Blanking register (CRT15) minus one.

Description:

End vertical blinking register.

CRT17 [RW] 8-bits Access: 8/16/32			CRT: 17
Field Name	**Bits**	**Default**	**Description**
RA0_AS_A13B	0	0	Compatibility Mode:
RA1_AS_A14B	1	0	Select Row Scan Counter:
VCOUNT_BY2	2	0	Vertical_by_2 NOTE: When bit 2 is logical one, other vertical register values should be adjusted as well (CRT06, CRT10, CRT12, CRT15, and CRT18).
ADDR_CNT_BY2	3	0	Count_by_2: ENGINEERING NOTE: Bit can be written and read, but has no effect.
(reserved)	4		
WRAP_A15TOA0	5	0	Address Wrap: ENGINEERING NOTE: Bit can be written and read, but has no effect.
BYTE_MODE	6	0	Byte/Word Mode: 0=WordMode 1=ByteMode
CRTC_SYNC_EN	7	0	H/V Retrace Enable: 0=Disable HVSync 1=EnaHVSync

Description:

CRT mode register.

CRT18 [RW] 8-bits Access: 8/16/32			CRT: 18
Field Name	**Bits**	**Default**	**Description**
LINE_CMP	7:0	0	These bits are the eight low-order of the 10-bit line compare register. Bit 8 is CRT07[4], bit 9 is CRT09[6]. The value of this register is used to disable scrolling on a portion of the display screen, as when split screen is active. When the vertical counter reaches this value, the memory address and row scan counters are cleared. The screen area above the line specified by the register is commonly called screen A. The screen below is screen B. Screen B cannot be scrolled, but it can panned only together with screen A, controlled by the PEL panning compatibility bit _ATTR10[5]. (For a description of this control bit see ATTR10[5].)

Description:

Line compare register.

CRT1E [RW] 8-bits Access: 8/16/32			CRT: 1E
Field Name	**Bits**	**Default**	**Description**
(reserved)	0		
GRPH_DEC_RD1	1	0	This register is used to read back the graphics controller index decode.
(reserved)	7:2		

Description:

Graphics controller index decode register.

CRT1F [RW] 8-bits Access: 8/16/32			CRT: 1F
Field Name	**Bits**	**Default**	**Description**
GRPH_DEC_RD0	7:0	0	This register is used to read back the graphics controller index decode.

Description:

Graphics controller index decode register.

CRT22 [RW] 8-bits Access: 8/16/32			CRT: 22
Field Name	**Bits**	**Default**	**Description**
GRPH_LATCH_DATA	7:0	0	This register is used to read the data in the Graphics Controller CPU data latches. The Graphics Controller Read Map Select register bits 0 and 1 determines which byte is read back.

Description:

RAM data latch readback register.

CRT00_S [RW] 8-bits Access: 8/16/32			CRT: 40
Field Name	**Bits**	**Default**	**Description**
H_TOTAL_S	7:0	0	<No Description>

CRT01_S			CRT: 41
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
H_DISP_END_M	7:0	0	<No Description>

CRT02_S			CRT: 42
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
H_BLANK_START_S	7:0	0	<No Description>

CRT03_S			CRT: 43
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
H_BLANK_END_S	4:0	0	<No Description>
H_DE_SKEW_S	6:5	0	<No Description> 0=0Skew 1=1Skew 2=2Skew 3=3Skew
CR10CR11_R_DIS_B_M	7	0	<No Description> 0=WrtOnlyToCRT10-11 1=WrtRdToCRT10-11

CRT04_S			CRT: 44
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
H_SYNC_START_S	7:0	0	<No Description>

CRT05_S			CRT: 45
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
H_SYNC_END_S	4:0	0	<No Description>
H_SYNC_SKEW_S	6:5	0	<No Description>
H_BLANK_END_B5_S	7	0	<No Description>

CRT06_S			CRT: 46
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_TOTAL_S	7:0	0	<No Description>

CRT07_S			CRT: 47
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_TOTAL_B8_S	0	0	<No Description>
V_DISP_END_B8_M	1	0	<No Description>
V_SYNC_START_B8_S	2	0	<No Description>
V_BLANK_START_B8_S	3	0	<No Description>
LINE_CMP_B8_M	4	0	<No Description>
V_TOTAL_B9_S	5	0	<No Description>
V_DISP_END_B9_M	6	0	<No Description>
V_SYNC_START_B9_S	7	0	<No Description>

CRT08_S			CRT: 48
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
ROW_SCAN_START_M	4:0	0	<No Description>
BYTE_PAN_M	6:5	0	<No Description>
(reserved)	7		

CRT09_S			CRT: 49
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
MAX_ROW_SCAN_M	4:0	0	<No Description>
V_BLANK_START_B9_S	5	0	<No Description>
LINE_CMP_B9_M	6	0	<No Description>
DOUBLE_CHAR_HEIGHT_M	7	0	<No Description> 0=200LineScan 1=400LineScan

CRT0A_S			CRT: 4A
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
CURSOR_START_M	4:0	0	<No Description>
CURSOR_DISABLE_M	5	0	<No Description> 0=on 1=off
(reserved)	7:6		

CRT0B_S			CRT: 4B
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
CURSOR_END_M	4:0	0	<No Description>
CURSOR_SKEW_M	6:5	0	<No Description>
(reserved)	7		

CRT0C_S			CRT: 4C
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
DISP_START_M	7:0	0	<No Description>

CRT0D_S			CRT: 4D
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
DISP_START_M	7:0	0	<No Description>

CRT0E_S			CRT: 4E
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
CURSOR_LOC_HI_M	7:0	0	<No Description>

CRT0F_S			CRT: 4F
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
CURSOR_LOC_LO_M	7:0	0	<No Description>

CRT10_S			CRT: 50
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_SYNC_START_S	7:0	0	<No Description>

CRT11_S			CRT: 51
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_SYNC_END_S	3:0	0	<No Description>
V_INTR_CLR_M	4	0	<No Description> 0=VRetraceIntCleared 1=Not Cleared
V_INTR_EN_M	5	0	<No Description> 0=VRetraceIntEna 1=Disable
(reserved)	6		
C0T7_WR_ONLY_M	7	0	<No Description> 0=EnaWrtToCRT00-07 1=C0T7B4WrtOnly

CRT12_S			CRT: 52
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_DISP_END_M	7:0	0	<No Description>

CRT13_S			CRT: 53
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
DISP_PITCH_M	7:0	0	<No Description>

CRT14_S			CRT: 54
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
UNDRLN_LOC_M	4:0	0	<No Description>
ADDR_CNT_BY4_M	5	0	<No Description> 0=Char. Clock 1=CountBy4
DOUBLE_WORD_M	6	0	<No Description> 0=Disable 1=DoubleWordMdEna
(reserved)	7		

CRT15_S			CRT: 55
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_BLANK_START_S	7:0	0	<No Description>

CRT16_S			CRT: 56
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
V_BLANK_END_S	7:0	0	<No Description>

CRT17_S			CRT: 57
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
RA0_AS_A13B_M	0	0	<No Description>
RA1_AS_A14B_M	1	0	<No Description>
VCOUNT_BY2_S	2	0	<No Description>

(Continued)

CRT17_S			CRT: 57
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
ADDR_CNT_BY2_M	3	0	<No Description>
(reserved)	4		
WRAP_A15TOA0_M	5	0	<No Description>
BYTE_MODE_M	6	0	<No Description> 0 = WordMode 1 = ByteMode
CRTC_SYNC_EN_M	7	0	<No Description> 0 = Disable HVSync 1 = EnaHVSync

CRT18_S			CRT: 58
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
LINE_CMP_M	7:0	0	<No Description>

CRT1E_S			CRT: 5E
[R] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
(reserved)	0		
GRPH_DEC_RD1_M	1	0	<No Description>
(reserved)	7:2		

CRT1F_S			CRT: 5F
[R] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
GRPH_DEC_RD0_M	7:0	0	<No Description>

CRT22_S [R] 8-bits Access: 8/16/32			CRT: 62
Field Name	**Bits**	**Default**	**Description**
GRPH_LATCH_DATA_M	7:0	0	<No Description>

CRTC_DEBUG [RW] 32-bits Access: 8/16/32			MMR: 21C, MMR_1: 21C, IND: 21C
Field Name	**Bits**	**Default**	**Description**
CRTC_GUI_TRIG_BYPASS_EN	0	0	<No Description> 0=Don't bypass gui triggers generated by disp eng 1=bypass gui triggers generated by disp eng
GUI_TRIG_VLINE_BYPASS	1	0	<No Description>
GUI_TRIG_OFFSET_BYPASS	2	0	<No Description>
GUI_TRIG_PITCH_ADD_BYPASS	3	0	<No Description>
(reserved)	31:4		

5.5 VGA Graphics Registers

GRPH8_IDX			VGA_IO: 3CE
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
GRPH_IDX	3:0	0	This index is used to address one of the internal registers of the graphics controller (GRAC) at I/O port 3CRF.
(reserved)	7:4		

GRPH8_DATA			VGA_IO: 3CF
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
GRPH_DATA	7:0	0	<No Description>

GRA00			GRPH: 00
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
GRPH_SET_RESET0	0	0	Set/Reset Map 0:
GRPH_SET_RESET1	1	0	Set/Reset Map 1:
GRPH_SET_RESET2	2	0	Set/Reset Map 2:
GRPH_SET_RESET3	3	0	Set/Reset Map 3:
(reserved)	7:4		

Description:

Set/reset register.

GRA01 [RW] 8-bits Access: 8/16/32			GRPH: 01
Field Name	**Bits**	**Default**	**Description**
GRPH_SET_RESET_ENA0	0	0	Enable Set/Reset Map 0:
GRPH_SET_RESET_ENA1	1	0	Enable Set/Reset Map 1:
GRPH_SET_RESET_ENA2	2	0	Enable Set/Reset Map 2:
GRPH_SET_RESET_ENA3	3	0	Enable Set/Reset Map 3:
(reserved)	7:4		

Description:

Enable set/reset register.

GRA02 [RW] 8-bits Access: 8/16/32			GRPH: 02
Field Name	**Bits**	**Default**	**Description**
GRPH_CCOMP	3:0	0	Color Compare Map bits 3:0. In Read mode (GRA05[3] being logical 1), the 4 bits from this register are compared with the 4-bit PEL value (made up of one bit from each map), from bit positions 0 through 7. As long as the color don't care bits (GRA07[0:3]) for the respective maps are logical 1's, the compare takes place only on those bits of the PEL value, and the CPU reads a one for a match in that bit position. If Color Don't Care bit for one map is a logical zero, the latched data from the map is excluded from the compare, and only the remaining three bits are compared to generate bus data.
(reserved)	7:4		

Description:

Color compare register.

GRA03 [RW] 8-bits Access: 8/16/32			GRPH: 03
Field Name	**Bits**	**Default**	**Description**
GRPH_ROTATE	2:0	0	Rotate Count Bits 2-0. Specifies the number of bit positions that the CPU data is to be rotated to the right, before doing the function selected by bits 3 and 4 above and subsequency bit mask select and write operations. Rotation is carried out only in write modes 0 and 3. In these two modes, the CPU data is rotated first, the operated only the function bits GRA03[4:3], the updated by the bit mask register GRA05.
GRPH_FN_SEL	4:3	0	Function Select Bits 1 and 2. These functions are performed on the CPU data before the selected bits are updated by the bit mask register, and then written to the display buffers. 0=Replace 1=AND 2=OR 3=XOR
(reserved)	7:5		

Description:

Data rotate register.

GRA04 [RW] 8-bits Access: 8/16/32			GRPH: 04
Field Name	**Bits**	**Default**	**Description**
GRPH_RMAP	1:0	0	Read Mode 0 Only: GRA controller returns the contents of one of the four latched buffer bytes to CPU each time a CPU read loads these latches. The 2 bits (0 and 1) define a value that represents the bit map where CPU is to read data - useful in transferring bit map data between the maps and system RAM.
(reserved)	7:2		

Description:

Read map select register.

GRA05 [RW] 8-bits Access: 8/16/32			GRPH: 05
Field Name	**Bits**	**Default**	**Description**
GRPH_WRITE_MODE	1:0	0	Write Mode: 0=Write mode 0 1=Write mode 1 2=Write mode 2 3=Write mode 3
(reserved)	2		
GRPH_READ1	3	0	Read Mode: 0=Read mode 0, byte oriented 1=Read mode 1, pixel oriented
CGA_ODDEVEN	4	0	Odd/Even Addressing Enable. Used to enable CGA emulation, this bit enables off/even addressing mode when it is logical one. Normally, this bit and memory mode bit SEQ04[2] are set to agree with each other in enabling odd/even mode emulation. 0=Disable Odd/Even Addressing 1=Enable Odd/Even Addressing
GRPH_OES	5	0	Shift Register Mode: This bit controls how data form memory is loaded into the shift registers M0D0:M0D7, M1D0:M1D7; M2D0:M2D7, and M3D0:M3D7 are representations of this data. 0=Linear shift mode 1=Tiled shift mode
GRPH_PACK	6	0	256 Color Mode. This bit also controls how data from memory is loaded into the shift registers. 0=Use shift register mode as per GRAP_OES 1=256 color mode, read as packed pixels, ignore GRPH_OES
(reserved)	7		

Description:Description

Graphics mode register.

GRA06			GRPH: 06
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
GRPH_GRAPHICS	0	0	Graphics/Alphanumeric Mode: 0=Alpha Numeric Mode 1=Graphics Mode
GRPH_ODDEVEN	1	0	Chains Odd Maps to Even: 0=Normal 1=Chain Odd maps to Even
GRPH_ADRSEL	3:2	0	Memory Map Read Bits 1 and 0, respectively. 0=A0000-128K 1=A0000-64K 2=B0000-32K 3=B8000-32K
(reserved)	7:4		

Description:

Graphics miscellaneous register.

GRA07			GRPH: 07
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
GRPH_XCARE0	0	0	Ignore Map 0 0=Ignore map 0 1=Use map 0 for read mode 1
GRPH_XCARE1	1	0	Ignore Map 1. 0=Ignore map 1 1=Use map 1 for read mode 1
GRPH_XCARE2	2	0	Ignore Map 2. 0=Ignore map 2 1=Use map 2 for read mode 1

(Continued)

GRA07 [RW] 8-bits Access: 8/16/32			GRPH: 07
Field Name	**Bits**	**Default**	**Description**
GRPH_XCARE3	3	0	Ignore Map 3. 0=Ignore map 3 1=Use map 3 for read mode 1
(reserved)	7:4		

Description:

Color don't care register.

GRA08 [RW] 8-bits Access: 8/16/32			GRPH: 08
Field Name	**Bits**	**Default**	**Description**
GRPH_BMSK	7:0	0	Bit Mask:

Description:

Bit mask register.

5.6 VGA Attribute Registers

ATTRX			VGA_IO: 3C0
[RW] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
ATTR_IDX	4:0	0	ATTR Index bits 4-0. This index points to one of the internal registers of the attribute controller (TTR) at addresses 3C1/ 3C0, for the next ATTR read/write operation. Since both the index and data registers are at the same I/O, a pointer to the registers is necessary. This pointer can be initialized to point to the index register by a read of GENS1.
ATTR_PAL_RW_ENB	5	0	Palette Address Source. After loading the color palette, this bit should be set to logical 1. 0 = Processor to load 1 = Memory data to access
(reserved)	7:6		

Description:

ATTR index register.

ATTRDW			VGA_IO: 3C0
[W] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
ATTR_DATA	7:0	0	<No Description>

ATTRDR			VGA_IO: 3C1
[R] 8-bits Access: 8/16/32			
Field Name	**Bits**	**Default**	**Description**
ATTR_DATA	7:0	0	<No Description>

Note: The table below covers 16 identical registers — from ATTR00 to ATTR0F

ATTR[0F:00] [RW] 8-bits Access: 8/16/32			ATTR: 00
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL0 to ATTR_PALF	5:0	0	Color Bits 5-0. Bits 0-5 map the text attribute or graphics color input value to a display color on the screen. Color is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.
(reserved)	7:6		

Description:

Palette register 0.

ATTR10 [RW] 8-bits Access: 8/16/32			ATTR: 10
Field Name	**Bits**	**Default**	**Description**
ATTR_GRPH_MODE	0	0	Graphics/Alphanumeric Mode. 0 = Alphanumeric Mode 1 = Graphic Mode
ATTR_MONO_EN	1	0	Monochrome/Color Attributes Select: 0 = Color Disp 1 = MonoChrome Disp
ATTR_LGRPH_EN	2	0	Line Graphics Enable: 0 = Disable 1 = Line Graphics Enable
ATTR_BLINK_EN	3	0	Blink Enable/Background Intensity 0 = Disable 1 = Blink Enable
(reserved)	4		
ATTR_PANTOPONLY	5	0	PEL Panning Compatibility: 0 = Panning both 1 = Panning only the top half screen

(Continued)

ATTR10 [RW] 8-bits Access: 8/16/32			ATTR: 10
Field Name	**Bits**	**Default**	**Description**
ATTR_PCLKBY2	6	0	PEL Clock Select: 0 = Shift register clocked every dot clock 1 = Packed Pixel Mode
ATTR_CSEL_EN	7	0	Alternate Color Source: 0 = Select ATTR00-0F bit 5:4 as P4 and P5 1 = Select ATTR14 bit 1:0 as P4 and P5

Description:

Mode control register.

ATTR11 [RW] 8-bits Access: 8/16/32			ATTR: 11
Field Name	**Bits**	**Default**	**Description**
ATTR_OVSC	7:0	0	Overscan Color

Description:

Overscan color register.

ATTR12 [RW] 8-bits Access: 8/16/32			ATTR: 12
Field Name	**Bits**	**Default**	**Description**
ATTR_MAP_EN	3:0	0	Enable Color Map bits. 0 = Disables data from respective map from being used for video output. 1 = Enables data from respective map for use in video output.

(Continued)

ATTR12 [RW] 8-bits Access: 8/16/32			ATTR: 12
Field Name	**Bits**	**Default**	**Description**
ATTR_VSMUX	5:4	0	Video Status Mux bits. These are control bits for the multiplexer on color bits P0-P7. The bit selection is also indicated at GENS1[5:4]: 00 = P2, P0 01 = P5, P4 10 = P3, P1 11 = P7, P6
(reserved)	7:6		

Description:

Color map enable register.

ATTR13 [RW] 8-bits Access: 8/16/32			ATTR: 13
Field Name	**Bits**	**Default**	**Description**
ATTR_PPAN	3:0	0	Shift Count Bits. The shift count value (0-8) indicates how many pixel positions to shift left. Shift in respective modes Count 0+, 1+, 2+, 13 All other Value 3+, 7, 7+ 0 1 0 01 2 - 12 3 1 23 4 - 34 5 2 45 6 - 56 7 3 67 8 - 78 0 - -
(reserved)	7:4		

Description:

Horizontal PEL panning register.

ATTR14 [RW] 8-bits Access: 8/16/32			ATTR: 14
Field Name	**Bits**	**Default**	**Description**
ATTR_CSEL1	1:0	0	Color bits P5 and P6, respectively. These are the color output bits (instead of bits 5 and 4 of the internal palette registers ATTTR00-0F) when alternate color source, bit ATTR10[7] is logical 1.
ATTR_CSEL2	3:2	0	Color bits P7 and P6, respectively. These two bits are the two high-order bits of the 8-bit color, used for rapid color set switching (addressing different parts of the DAC color lookup table). The lower order bits are in registers ATTR00-0F.
(reserved)	7:4		

Description:

Color select register.

This page intentionally left blank.

Chapter 6
CRTC and DAC

6.1 CRTC Registers

The CRTC generates the horizontal sync, vertical sync, and blank signals used to position the pixel data on the display monitor. All horizontal parameters are in terms of characters (pixels * 8). All vertical parameters are in terms of lines. Accurate display centering is possible by adjusting CRTC_HORZ_SYNC_DLY. A vertical blank and vertical line interrupt allows video synchronization without motion tearing artifacts. Monitor power management is controlled through CRTC_HSYNC_DIS and CRTC_VSYNC_DIS.

CRTC_GEN_CNTL			**MMR: 50, MMR_1: 50,**
[RW] 32-bits Access: 8/16/32			**IOR: 50**
Field Name	**Bits**	**Default**	**Description**
CRTC_DBL_SCAN_EN	0	0	Double scan enable. Double scan only affects the calculation of display addresses by adding the CRTC_PITCH every second line (also applies to the hardware cursor pitch). Enabling double scan does not change the CRTC vertical programming or VSYNC timing. The overscan top & bottom are not affected and remain the number of lines programmed (i.e. not double). The hardware cursor programming remains in terms of physical lines (not logical lines). The cursor vertical position must begin on an even line number when in double scan. The cursor itself is limited to 64 physical lines in height, which means only 32 logical lines. This is because the cursor pitch is only added at the end of odd scan lines, but the CRTC vertical logic stops the cursor after 64 physical lines. 0 = Disable 1 = Enable
CRTC_INTERLACE_EN	1	0	Interlace enable. 0 = Non-Interlace 1 = Interlace
(reserved)	3:2		

(Continued)

CRTC_GEN_CNTL [RW] 32-bits Access: 8/16/32			MMR: 50, MMR_1: 50, IOR: 50
Field Name	**Bits**	**Default**	**Description**
CRTC_C_SYNC_EN	4	0	Enables composite sync on horizontal sync output. 0 = Disable 1 = Enable
(reserved)	7:5		
CRTC_PIX_WIDTH	10:8	0	Display pixel width: 1 = 4bpp 2 = 8bpp 3 = 15bpp 4 = 16bpp 5 = 24bpp 6 = 32bpp
(reserved)	15:11		
CRTC_CUR_EN	16	0	Hardware Cursor enable 0 = Disable 1 = Enable
CRTC_CUR_MODE	19:17	0	Hardware Cursor Mode: 0 = 2bpp monochrome 64x64. 2 colour, transparent, inverse. Others = reserved for future use. 0 = Normal AGPIO2/3 & VGA_VSTATUS until vcount=vtotal, DISP_ADDR loads when vcount=vtotal 1 = Normal AGPIO2/3 & VGA_VSTATUS until vblank end, DISP_ADDR loads when vcount=vtotal 2 = Normal AGPIO2/3 & VGA_VSTATUS until vcount=vtotal,DISP_ADDR loads in vsync start 3 = Normal AGPIO2/3 & VGA_VSTATUS until vblank end,DISP_ADDR loads in vsync start 4 = PANELID on AGPIO2/3 & VGA_VSTATUS until vcount=vtotal, DISP_ADDR loads when vcount=vtotal 5 = PANELID on AGPIO2/3 & VGA_VSTATUS until vblank end, DISP_ADDR loads when vcount=vtotal 6 = PANELID on AGPIO2/3 & VGA_VSTATUS until vcount=vtotal,DISP_ADDR loads in vsync start 7 = PANELID on AGPIO2/3 & VGA_VSTATUS until vblank end,DISP_ADDR loads in vsync start
(reserved)	23:20		

(Continued)

CRTC_GEN_CNTL [RW] 32-bits Access: 8/16/32			MMR: 50, MMR_1: 50, IOR: 50
Field Name	**Bits**	**Default**	**Description**
CRTC_EXT_DISP_EN	24	0	Extended display mode enable: (default = 0) 0 = VGA 1 = Extended
CRTC_EN	25	0	Enables CRT controller: (default = 0) 0 = Reset 1 = Enable
CRTC_DISP_REQ_EN_B	26	1	Display Request Enable: 0 = Enable 1 = Disable (default)
(reserved)	31:27		

CRTC_EXT_CNTL [RW] 32-bits Access: 8/16/32			MMR: 54, MMR_1: 54, I0R: 54, IND: 54
Field Name	**Bits**	**Default**	**Description**
CRTC_VGA_XOVERSCAN	0	0	VGA Overscan: 0 = Disable extended overscan in VGA 1 = Enable extended overscan in VGA
VGA_BLINK_RATE	2:1	0	Controls number of frames per blink for VGA modes. 0 = Default VGA blink rate (16 frames) 1 = 1/2 default VGA blink rate (32 frames) 2 = 1/3 default VGA blink rate (48 frames) 3 = 1/4 default VGA blink rate (64 frames)
VGA_ATI_LINEAR	3	0	Enable linear addressing through VGA aperture 0 = Disable 1 = Enable
VGA_128KAP_PAGING	4	0	Enable extended aperture paging in 128K VGA aperture mode: 0 = Normal 1 = Enable
VGA_TEXT_132	5	0	Extended text mode select (linear address 132 column text mode) 0 = inActive 1 = Active

(Continued)

CRTC_EXT_CNTL [RW] 32-bits Access: 8/16/32			MMR: 54, MMR_1: 54, I0R: 54, IND: 54
Field Name	**Bits**	**Default**	**Description**
VGA_XCRT_CNT_EN	6	0	Extended CRTC display address counter enable. Active High 0 = Disable 1 = Enable Extended CRTC Counter
(reserved)	7		
CRTC_HSYNC_DIS	8	0	Disables horizontal sync output. 0 = Enable 1 = Disable
CRTC_VSYNC_DIS	9	0	Disables vertical sync output. 0 = Enable 1 = Disable
CRTC_DISPLAY_DIS	10	0	Disables the display, forcing the blanking signal to be active. 0 = Enable 1 = Blanked
CRTC_SYNC_TRISTATE	11	0	Sync Tristate Enable: 0 = Normal 1 = Tristate HSYNC and VSYNC outputs
(reserved)	16:12		
VGA_CUR_B_TEST	17	0	Test cursor blinking. Active High. 0 = Disable VGA cursor test 1 = Test VGA cursor blinking
VGA_PACK_DIS	18	0	Controls host write pipe for packed VGA modes (e.g. mode 13): 0 = Fast VGA write in packed modes 1 = Normal VGA write in packed modes
VGA_MEM_PS_EN	19	0	VGA Page Select Enable: 0 = Do not use MEM_VGA_WP_SEL and MEM_VGA_RP_SEL registers 1 = Use MEM_VGA_WP_SEL and MEM_VGA_RP_SEL registers
VGA_READ_PREFETCH_DIS	20	0	VGA read pre-fetching control: 0 = Prefetch VGA read data for next byte after each read. 1 = Disable VGA read prefetching.

(Continued)

CRTC_EXT_CNTL [RW] 32-bits Access: 8/16/32			MMR: 54, MMR_1: 54, I0R: 54, IND: 54
Field Name	**Bits**	**Default**	**Description**
DFIFO_EXTSENSE	21	1	Extended Sensing control for display FIFO macro 0 = Enable use of the AGPIO0 (VSYNC)and AGPIO1 (HSYNC) pins as a separate pair of V/HSYNC signals for the Flat Panel LCD interface going to an external TMDS transmitter. The AGPIO[0:1] pins are only enabled as V/HSYNC when this bit is set to '0' and RAGE128 and the board are LCD-enabled. In this case, AGP is forced to 'ON' (AGP_STOPB = 0) and CLKRUN is forsed to 'ON' (CLKRUN = 1). 1 = Normal functionality of the AGPIO[0:1] pins.
FP_OUT_EN	22	0	Flat Panel output control: 0 = Tri-state flat panel outputs 1 = Enable flat panel outputs, if strapped for panel mode
FP_ACTIVE	23	0	Flat Panel strap override. Setting low will return pins to other operation if strapped for panel. No affect if not strapped for panel operation: 0 = Flat panel outputs not set to panel function 1 = Flat panel outputs set to panel data from RAMDAC, if strapped for panel mode
VCRTC_IDX_MASTER	30:24	0	VGA CRTC master index. Only bits 5:0 of the VGA CRTC index can be written (or read) in VGA I/O space at 3B4 or 3D4. Bit 6 controls whether the master or shadow set of VGA CRTC registers is seen in VGA I/O space. The shadow set is for use when supporting panel operation in VGA modes. The BIOS will leave either the master or shadow set active as needed after a mode switch call.
(reserved)	31		

Description:

Extended general CRTC controls.

CRTC_STATUS [RW] 32-bits Access: 8/16/32			MMR: 5C, MMR_1: 5C, IOR: 5C, IND: 5C
Field Name	**Bits**	**Default**	**Description**
CRTC_VBLANK_CUR **(R)**	0	0	<No Description> 0 = Not in vertical blank 1 = In vertical blank
CRTC_VBLANK_SAVE_ CLEAR **(W)**	1	0	<No Description> 0 = No effect 1 = Clear CRTC_VBLANK_SAVE
CRTC_VBLANK_SAVE **(R)**	1	0	<No Description> 0 = No vertical blank since last clear 1 = Vertical Blank since last cleared
CRTC_VLINE_SYNC **(R)**	2	0	<No Description> 0 = Even scan line 1 = Odd scan line
CRTC_FRAME **(R)**	3	0	<No Description> 0 = Even frame 1 = Odd frame
(reserved)	31:4		

Description:

Status bits to determine current state of the display.

CRTC_H_TOTAL_DISP [RW] 32-bits Access: 8/16/32			MMR: 200, MMR_1: 200, IND: 200
Field Name	**Bits**	**Default**	**Description**
CRTC_H_TOTAL	8:0	0	Horizontal Total (pixels * 8). Sum of display width, overscan right, front porch and overscan left.
(reserved)	15:9		
CRTC_H_DISP	23:16	0	Horizontal display end (pixels * 8). Determines number of visible pixels, not including overscan
(reserved)	31:24		

Description:

Horizontal total control.

CRTC_H_SYNC_STRT_WID [RW] 32-bits Access: 8/16/32			MMR: 204, MMR_1: 204, IND: 204
Field Name	**Bits**	**Default**	**Description**
CRTC_H_SYNC_STRT_PIX	2:0	0	Horizontal sync start delay in pixels. Allows pixel accurate horizontal positioning by delaying sync position within character position set below.
CRTC_H_SYNC_STRT_CHAR	11:3	0	Horizontal sync start (pixels * 8). Sum of display width, overscan right and front porch.
(reserved)	15:12		
CRTC_H_SYNC_WID	21:16	0	Horizontal sync width (pixels * 8)
(reserved)	22		
CRTC_H_SYNC_POL	23	0	Horizontal sync polarity (1 = active low) 0=Positive 1=Negative
(reserved)	31:24		

Description:

Horizontal sync control.

CRTC_V_TOTAL_DISP [RW] 32-bits Access: 8/16/32			MMR: 208, MMR_1: 208, IND: 208
Field Name	**Bits**	**Default**	**Description**
CRTC_V_TOTAL	10:0	0	Vertical total. Sum of display height, overscan bottom, front porch, sync width, back porch and overscan top.
(reserved)	15:11		
CRTC_V_DISP	26:16	0	Vertical display end. Determines number of visible lines, not including overscan.
(reserved)	31:27		

Description:

Vertical total control.

CRTC_V_SYNC_STRT_WID [RW] 32-bits Access: 8/16/32			MMR: 20C, MMR_1: 20C, IND: 20C
Field Name	**Bits**	**Default**	**Description**
CRTC_V_SYNC_STRT	10:0	0	Vertical sync start. Sum of display height, overscan bottom and front porch.
(reserved)	15:11		
CRTC_V_SYNC_WID	20:16	0	Vertical sync width
(reserved)	22:21		
CRTC_V_SYNC_POL	23	0	Vertical sync polarity: 0=Positive 1=Negative
(reserved)	31:24		

Description:

Vertical sync control.

CRTC_VLINE_CRNT_VLINE [RW] 32-bits Access: 8/16/32			MMR: 210, MMR_1: 210, IND: 210
Field Name	**Bits**	**Default**	**Description**
CRTC_VLINE	10:0	0	Vertical line at which vertical line interrupt is triggered.
(reserved)	15:11		
CRTC_CRNT_VLINE [**R**]	26:16	0	Current vertical line.
(reserved)	31:27		

Description:

Display current vertical line.

CRTC_GUI_TRIG_VLINE [RW] 32-bits Access: 8/16/32			MMR: 218, MMR_1: 218, IND: 218
Field Name	**Bits**	**Default**	**Description**
CRTC_GUI_TRIG_VLINE_START	10:0	0	The START (upper in display, lower in memory) for the GUI_TRIG_VLINE compare. First line of display is line 0.
(reserved)	15:11		
CRTC_GUI_TRIG_VLINE_END	26:16	0	The END (lower in display, higher in memory) for the GUI_TRIG_VLINE compare. First line of display is line 0.
(reserved)	30:27		
CRTC_GUI_TRIG_VLINE [**R**]	31	0	This signal is active high when the raster is between the START and END, i.e. START <= raster <= END. 0 = Current line not between VLINE start and end 1 = Current line is between VLINE start and end, inclusive

Description:

Trigger to GUI engine activated in certain vertical region of the display, when the raster is between START and END. Normally used to delay rendering operations until the raster has passed a specific point.

CRTC_OFFSET [RW] 32-bits Access: 8/16/32			MMR: 224, MMR_1: 224, IND: 224
Field Name	**Bits**	**Default**	**Description**
CRTC_OFFSET	24:0	0	Primary graphics display base address in frame buffer in terms of 64 bit words. Updated for buffer flips and for virtual desktop movement. Must always point to the start of a character of display data (i.e. can not move horizontally by sub-character amount). Must pan by 16 pixels in 4bpp modes.In tiling mode, this field should be written with the start address of the display after tiling but before checkboarding. It means, in tiling mode the hardware won't convert this address to a linear equivalent (so this is not the 'virtual' tiled address but rather the 'real' physical address). But the programmer does not need to checkerboard this address, the hardware will do it.e.g.: surface offset is zero and the display will start at line 3: CRTC_OFFSET = (surface base) + ((start line) * 40) = C0 because each tile is 64 bytes wide. If the vertical offset exceeds the height of a tile (16 lines), then again the real address (before checkerboarding) of the start of the first line must be calculated, i.e. for tiled: CRTC_OFFSET = (surface base) + (((start line) DIV 16) * CRTC_PITCH) + (((start line) MOD 16)* 40). NOTE: Bits 2:0 of this field are hardwired to ZERO
(reserved)	29:25		
CRTC_GUI_TRIG_OFFSET **(R)**	30	0	Indicates if visible buffer is last written, or still the previous one. This register is read only. Goes high when an offset has been written but the corresponding buffer does not appear on screen yet. It goes low again when display starts for that address. 0 = Last CRTC_OFFSET written is being displayed 1 = Last CRTC_OFFSET written not yet displayed

(Continued)

CRTC_OFFSET [RW] 32-bits Access: 8/16/32			MMR: 224, MMR_1: 224, IND: 224
Field Name	**Bits**	**Default**	**Description**
CRTC_OFFSET_LOCK	31	0	Prevents hardware from internally updating the following fields until cleared: CRTC_OFFSET, CRTC_TILE_LINE.It permits atomic update of CRTC_OFFSET and CRTC_TILE_LINE. Normal operation is with the lock in zero. 0 = Unlock these registers 1 = Lock these registers

Description:

Graphics base address offset.

CRTC_OFFSET_CNTL [RW] 32-bits [RW] 32-bits Access: 8/16/32			MMR: 228, MMR_1: 228, IND: 228
Field Name	**Bits**	**Default**	**Description**
CRTC_TILE_LINE	4:0	0	When CRTC_TILE_EN = 1, this indicates how many lines down in the first tile the CRTC_OFFSET starts. The display address generator needs to know this to determine the proper pitch to add at the end of each display line. This is normally 0, unless the display is in a virtual desktop mode. For example, if the surface offset is zero and the display starts in line 3, CRTC_TILE_LINE= 3CRTC_OFFSET = C0Note that tiles are 16 lines high, but this register must contain ((start line) MOD 32) in order to do the checkerboarding correctly. Do not worry about what checkerboarding is, you shouldn't need to know.
(reserved)	7:5		

(Continued)

CRTC_OFFSET_CNTL [RW] 32-bits [RW] 32-bits Access: 8/16/32			MMR: 228, MMR_1: 228, IND: 228
Field Name	**Bits**	**Default**	**Description**
CRTC_TILE_ALIGN	10:8	0	Alignment of graphics display surface in tiled mode. Indicates memory alignment of the display surface (i.e. first tile), which is not the same as CRTC_OFFSET if using virtual desktop: 0 = 64 byte aligned surface 1 = 2k byte aligned surface 2 = 4k byte aligned surface 3 = 8k byte aligned surface 4 = 16k byte aligned surface
(reserved)	14:11		
CRTC_TILE_EN	15	0	Graphics display tiling enable: 0 = Display Surface uses linear addressing 1 = Display surface uses tiled addressing
CRTC_OFFSET_FLIP_CNTL	16	0	Selects position within the frame at which new CRTC_OFFSET will be used.Should be normally zero. If set to one, a new offset will be taken at the end of the line instead of the end of the frame. 0 = Use new CRTC_OFFSET on vertical blank 1 = Use new CRTC_OFFSET on any horizontal blank. Note, this can cause the display to tear.
(reserved)	29:17		
CRTC_GUI_TRIG_ OFFSET [**R**]	30	0	Indicates if visible buffer is last written, or still the previous one: 0 = Last CRTC_OFFSET written is being displayed 1 = Last CRTC_OFFSET written not yet displayed
CRTC_OFFSET_LOCK	31	0	Prevents hardware from internally updating the following fields until cleared. CRTC_OFFSET, CRTC_TILE_LINE: 0 = Unlock these registers 1 = Lock them

Description:

Graphics display address generator control.

CRTC_PITCH [RW] 32-bits Access: 8/16/32			MMR: 22C, MMR_1: 22C, IND: 22C
Field Name	**Bits**	**Default**	**Description**
CRTC_PITCH	9:0	0	Display line pitch in (pixels * 8). Note that for 24bpp the display uses pixels * 8 for the pitch, but the rendering engine uses bytes * 8 for the pitch. For tiled display this is the same pitch as used for the surface in the rendering engine (except for 24bpp, as above). This must be a multiple of 64 bytes (the basic tile width).
(reserved)	31:10		

Description:

Graphics display address pitch.

CRTC_CRNT_FRAME [RW] 32-bits Access: 8/16/32			MMR: 214, MMR_1: 214, IND: 214
Field Name	**Bits**	**Default**	**Description**
CRTC_CRNT_FRAME (R)	20:0	0	Readback of current value of display frame counter. Used by display time sensitive applications such as video playback.
(reserved)	31:21		

Description:

Current Frame.

DDA_CONFIG [RW] 32-bits Access: 8/16/32			MMR: 2E0, MMR_1: 2E0, IND: 2E0
Field Name	**Bits**	**Default**	**Description**
DDA_XCLKS_PER_XFER	13:0	0	Amount of time in XCLKs that one transfer to the Display FIFO occupies
(reserved)	15:14		
DDA_PRECISION	19:16	0	Integer.Fraction precision point for DDA_XCLKS_PER_XFER DDA_ON DDA_OFF
DDA_LOOP_LATENCY	24:20	0	Display FIFO control parameter to reflect the number of XCLKs of latency required in the hardware.
(reserved)	31:25		

Description:

Contains DDA parameters that set the way data is fetched from the memory to be displayed.

DDA_ON_OFF [RW] 32-bits Access: 8/16/32			MMR: 2E4, MMR_1: 2E4, IND: 2E4
Field Name	**Bits**	**Default**	**Description**
DDA_OFF	15:0	0	The Display memory request off threshold time in terms of XCLKs
DDA_ON	31:16	0	The display memory request on threshold time in terms of XCLKs

Description:

Indicates at what levels of the FIFO to start and end fetching data.

VGA_DDA_CONFIG [RW] 32-bits Access: 8/16/32			MMR: 2E8, MMR_1: 2E8, IND: 2E8
Field Name	**Bits**	**Default**	**Description**
VGA_DDA_XCLKS_PER_XFER	13:0	0	Amount of time in XCLKs that one transfer to the display FIFO occupies in VGA modes
(reserved)	19:14		
VGA_DDA_PREC_PCLKBY2	23:20	0	Integer.fraction precision point for: VGA_DDA_PREC_PCLK+1
VGA_DDA_PREC_PCLK	27:24	0	Integer.fraction precision point for: DDA_XCLKS_PER_XFER DDA_ON DDA_OFF
(reserved)	31:28		

VGA_DDA_0N_OFF [RW] 32-bits Access: 8/16/32			MMR: 2EC, MMR_1: 2EC, IND: 2EC
Field Name	**Bits**	**Default**	**Description**
VGA_DDA_OFF	15:0	0	The display memory request off threshold time in terms of XCLKs for VGA modes
VGA_DDA_ON	31:16	0	The display memory request on threshold time in terms of XCLKs for VGA modes

6.2 Overscan Registers

Display overscan is enabled if any of the overscan width values is non-zero. The left and right overscan widths are described in terms of pixels * 8 and the top and bottom overscan widths are described in terms of vertical lines. The overscan color is defined by an 8 bit index and a 24 bit color. In all display modes the 24 bit color will be used by the internal RAMDAC and displayed on the monitor attached to the RAGE128. Note this is always a true color which is not mapped through the palette. The 8 bit index color is used in 4 bpp and 8 bpp modes for data going out on the 8 bit feature connector. The receiving board is expected to index all 4 and 8 cpp data through it's own palette.

OVR_CLR [RW] 32-bits Access: 8/16/32			MMR: 230, MMR_1: 230, IND: 230
Field Name	**Bits**	**Default**	**Description**
OVR_CLR_B	7:0	0	Blue overscan color, to internal DAC.
OVR_CLR_G	15:8	0	Green overscan color, to internal DAC.
OVR_CLR_R	23:16	0	Red overscan color, to internal DAC.
(reserved)	31:24		

Description:

Overscan color. Always 24 bit, independent of pixel depth.

OVR_WID_LEFT_RIGHT [RW] 32-bits Access: 8/16/32			MMR: 234, MMR_1: 234, IND: 234
Field Name	**Bits**	**Default**	**Description**
OVR_WID_RIGHT	5:0	0	Right overscan width (in pixels * 8)
(reserved)	15:6		
OVR_WID_LEFT	21:16	0	Left overscan width (in pixels * 8)
(reserved)	31:22		

Description:

Overscan border left/right width control.

OVR_WID_TOP_BOTTOM [RW] 32-bits Access: 8/16/32			MMR: 238, MMR_1: 238, IND: 238
Field Name	**Bits**	**Default**	**Description**
OVR_WID_BOTTOM	8:0	0	Bottom overscan width (in scan lines)
(reserved)	15:9		
OVR_WID_TOP	24:16	0	Top overscan width (in scan lines)
(reserved)	31:25		

Description:

Overscan border top/bottom width control.

6.3 Hardware Cursor Registers

CUR_OFFSET [RW] 32-bits Access: 8/16/32			MMR: 260, MMR_1: 260, IND: 260
Field Name	**Bits**	**Default**	**Description**
CUR_OFFSET	24:0	0	Hardware cursor address offset. Must be in the frame buffer, and be 16 byte (128 bit) aligned. This value is adjusted to move the cursor off the top edge of the display. See the CUR_VERT_OFF description. NOTE: Bits 3:0 of this field are hardwired to ZERO
(reserved)	30:25		
CUR_LOCK	31	0	Locks the CUR_OFFSET, CUR_HORZ_VERT_POSN and CUR_HORZ_VERT_OFF registers to allow tear free atomic updating of the cursor shape and/or position. Moving the cursor around on the top and/or left edges, or changing the shape, requires multiple register writes. If these were done without setting CUR_LOCK, then flicker could occur. 0 = Unlocked 1 = Locked

Description:

Location of the hardware cursor image.

CUR_HORZ_VERT_POSN [RW] 32-bits Access: 8/16/32			MMR: 264, MMR_1: 264, IND: 264
Field Name	**Bits**	**Default**	**Description**
CUR_VERT_POSN	10:0	0	Cursor vertical position. To move the cursor off the top edge set CUR_VERT_POSN = 0 and see the CUR_VERT_OFF description.
(reserved)	15:11		
CUR_HORZ_POSN	26:16	0	Cursor horizontal position. To move the cursor off the left edge set CUR_HORZ_POSN = 0 and see the CUR_HORZ_OFF description.

(Continued)

CUR_HORZ_VERT_POSN [RW] 32-bits Access: 8/16/32			MMR: 264, MMR_1: 264, IND: 264
Field Name	**Bits**	**Default**	**Description**
(reserved)	30:27		
CUR_LOCK	31	0	Locks the CUR_OFFSET, CUR_HORZ_VERT_POSN and CUR_HORZ_VERT_OFF registers to allow tear free atomic updating of the cursor shape and/or position. Moving the cursor around on the top and/or left edges, or changing the shape, requires multiple register writes. If these were done without setting CUR_LOCK, then flicker could occur. 0 = Unlocked 1 = Locked

Description:

Sets the screen position of the top left pixel of the visible part of the hardware cursor.

CUR_HORZ_VERT_OFF [RW] 32-bits Access: 8/16/32			MMR: 268, MMR_1: 268, IND: 268
Field Name	**Bits**	**Default**	**Description**
CUR_VERT_OFF	5:0	0	Cursor vertical offset. Height of cursor is (64-CUR_VERT_OFF). To move the cursor off the top of the display, set CUR_VERT_POSN to 0, add 16*(number of lines to move off the top) to CUR_OFFSET, and increase CUR_VERT_OFF by the same number of lines.
(reserved)	15:6		

(Continued)

CUR_HORZ_VERT_OFF [RW] 32-bits Access: 8/16/32			MMR: 268, MMR_1: 268, IND: 268
Field Name	**Bits**	**Default**	**Description**
CUR_HORZ_OFF	21:16	0	Cursor horizontal offset. Width of the cursor is always 64 pixels. CUR_HORZ_OFF controls how far into the cursor map from the left is 'pixel 0'.The horizontal position on the display of 'pixel 0' is set by CUR_HORZ_POSN. Therefore to move the cursor off the left edge of the display, set the CUR_HORZ_POSN to zero, and increase the CUR_HORZ_OFF by the number of pixels off the left edge.
(reserved)	30:22		
CUR_LOCK	31	0	Locks the CUR_OFFSET, CUR_HORZ_VERT_POSN and CUR_HORZ_VERT_OFF registers to allow tear free atomic updating of the cursor shape and/or position. Moving the cursor around on the top and/or left edges, or changing the shape, requires multiple register writes. If these were done without setting CUR_LOCK, then flicker could occur. 0 = Unlocked 1 = Locked

Description:

Controls the size of the hardware cursor mask in memory, and used to move the cursor off the top and/or left edges of the display.

CUR_CLR0 [RW] 32-bits Access: 8/16/32			MMR: 26C, MMR_1: 26C, IND: 26C
Field Name	**Bits**	**Default**	**Description**
CUR_CLR0_B	7:0	0	Blue cursor color 0, to internal DAC
CUR_CLR0_G	15:8	0	Green cursor color 0, to internal DAC
CUR_CLR0_R	23:16	0	Red cursor color 0, to internal DAC
(reserved)	31:24		

Description:

Hardware cursor color 0. Always 24bpp, independent of graphics mode.

CUR_CLR1 [RW] 32-bits Access: 8/16/32			MMR: 270, MMR_1: 270, IND: 270
Field Name	**Bits**	**Default**	**Description**
CUR_CLR1_B	7:0	0	Blue cursor color 1, to internal DAC
CUR_CLR1_G	15:8	0	Green cursor color 1, to internal DAC
CUR_CLR1_R	23:16	0	Red cursor color 1, to internal DAC
(reserved)	31:24		

Description:

Hardware cursor color 1. Always 24bpp, independent of graphics mode.

6.4 GenLocking Registers

MEM_ADDR_CONFIG [RW] 32-bits Access: 8/16/32			MMR: 148, MMR_1: 148, IND: 148
Field Name	**Bits**	**Default**	**Description**
MEM_ADDR_MAPPING	3:0	0	Row/Column/Banks address mapping of target memory. 0 = 9 row bits x 8 col bits x 2 banks 1 = 10 row bits x 8 col bits x 2 banks 2 = 11 row bits x 8 col bits x 2 banks 3 = 12 row bits x 8 col bits x 2 banks: CS2 = A12 4 = 13 row bits x 8 col bits x 2 banks: CS2 = A12, CS3 = A13 8 = 9 row bits x 8 col bits x 4 banks 9 = 10 row bits x 7 col bits x 4 banks 10 = 10 row bits x 8 col bits x 4 banks 11 = 11 row bits x 7 col bits x 4 banks: CS2 = A12 12 = 11 row bits x 8 col bits x 4 banks: CS2 = A12 13 = 12 row bits x 8 col bits x 4 banks: CS2 = A12, CS3 = A13
MEM_AP_MAPPING	6:4	0	Address bit used for auto-precharge function. 0 = Address bit 8 1 = Address bit 9 2 = Address bit 10 3 = Address bit 11 4 = Address bit 12
(reserved)	7		
MEM_BUS_WIDTH	8	0	Memory Data bus width. 0=64 bits 1=128 bits
(reserved)	15:9		
MEM_CHECKBOARD	17:16	0	Address bit to 'twiddle' in order to get desired checkerboard pattern of tiled memory surfaces. 0 = twiddle byte address bit 10 1 = twiddle byte address bit 11 2 = twiddle byte address bit 12 3 = twiddle byte address bit 13
(reserved)	19:18		

(Continued)

MEM_ADDR_CONFIG [RW] 32-bits Access: 8/16/32			MMR: 148, MMR_1: 148, IND: 148
Field Name	**Bits**	**Default**	**Description**
MEM_BLKWR_MODE	21:20	0	Level of block write support of the memory. 0 = Block write disabled 1 = Block write disabled 2 = Block write enabled without column byte mask 3 = Block write enabled with column byte mask
(reserved)	31:22		

Description:

Configuration of memory interface.

SNAPSHOT_VH_COUNTS [R] 32-bits Access: 8/16/32			MMR: 240, MMR_1: 240, IND: 240
Field Name	**Bits**	**Default**	**Description**
SNAPSHOT_HCOUNT	8:0	0	Snapshot of CRTC vertical count value.
(reserved)	15:9		
SNAPSHOT_VCOUNT	26:16	0	Snapshot of CRTC horizontal count value.
(reserved)	31:27		

SNAPSHOT_F_COUNT [R] 32-bits Access: 8/16/32			MMR: 244, MMR_1: 244, IND: 244
Field Name	**Bits**	**Default**	**Description**
SNAPSHOT_F_COUNT	20:0	0	Snapshot of CRTC frame count value.
(reserved)	31:21		

N_VIF_COUNT [RW] 32-bits Access: 8/16/32			MMR: 248, MMR_1: 248, IND: 248
Field Name	**Bits**	**Default**	**Description**
N_VIF_COUNT_VAL	9:0	0	Programmable N-video-in-field count value which is used to generate a snapshot interrupt when this N-count value is equal to the count value of the lower 10-bit SNAPSHOT_VIF_COUNT (See also CRTC_INT_CNTL register [8:7] - 0_06 for the snapshot interrupt specification).
(reserved)	30:10		
GENLOCK_SOURCE_SEL	31	0	<No Description>

SNAPSHOT_VIF_COUNT [RW] 32-bits Access: 8/16/32			MMR: C4, MMR_1: C4, IOR: C4, IND: C4
Field Name	**Bits**	**Default**	**Description**
LSNAPSHOT_VIF_COUNT **(R)**	9:0	0	Lower Snapshot of Video-in-field count value (lower 10-bit [9:0] indicate the current number of frames).
USNAPSHOT_VIF_COUNT **(R)**	20:10	0	Upper Snapshot of Video-in-field count value (upper 11-bit [20:10] indicate the number of N-frames).
(reserved)	23:21		
AUTO_SNAPSHOT_TAKEN_WR **(W)**	24	0	<No Description>
AUTO_SNAPSHOT_TAKEN_RD **(R)**	24	0	<No Description>
MANUAL_SNAPSHOT_NOW	25	0	1 = Snapshot taken immediately (writing '1' to this bit prevents all auto-snapshot taking until a write of '0' to the AUTO_SNAPSHOT_TAKEN bit that will re-enable the auto-snapshot taking).
(reserved)	31:26		

6.5 Memory Control Registers

MEM_VGA_WP_SEL [RW] 32-bits Access: 8/16/32			MMR: 38, MMR_1: 38, IOR: 38, IND: 38
Field Name	**Bits**	**Default**	**Description**
MEM_VGA_WPS0	9:0	0	<No Description>
(reserved)	15:10		
MEM_VGA_WPS1	25:16	0	<No Description>
(reserved)	31:26		

MEM_VGA_RP_SEL [RW] 32-bits Access: 8/16/32			MMR: 3C, MMR_1: 3C, IOR: 3C, IND: 3C
Field Name	**Bits**	**Default**	**Description**
MEM_VGA_RPS0	9:0	0	<No Description>
(reserved)	15:10		
MEM_VGA_RPS1	25:16	0	<No Description>
(reserved)	31:26		

MEM_CNTL [RW] 32-bits Access: 8/16/32			MMR: 140, MMR_1: 140, IND: 140
Field Name	**Bits**	**Default**	**Description**
MEM_CFG_TYPE	1:0	0	Configuration type of memory interface. 0 = SDR SGRAM (1:1) 1 = SDR SGRAM (2:1) 2 = DDR SGRAM
(reserved)	2		
MEM_BW_COL	3	0	Number of columns written by block write command. 0 = 8 columns 1 = 16 columns

(Continued)

MEM_CNTL [RW] 32-bits Access: 8/16/32			MMR: 140, MMR_1: 140, IND: 140
Field Name	**Bits**	**Default**	**Description**
MEM_ERST_CNTL	5:4	0	Delay of internal ERST signal after read command. Only relevant when memory configuration type is SDR2:1 or DDR. 0 = (CL-1) clocks 1 = (CL-1/2) clocks 2 = CL clocks 3 = Always enabled
MEM_DREN_CNTL	7:6	0	Delay of internal DRAN signal after read command. Only relevant when memory configuration type is SDR2:1 or DDR. 0 = (CL-1) clocks 1 = (CL-1/2) clocks 2 = CL clocks 3 = Always enabled
MEM_LATENCY	10:8	0	Memory read data latching delay from CASE (typically the same setting as MEM_CAS_LATENCY). 0 = 1 clock 1 = 2 clocks 2 = 3 clocks 3 = 4 clocks
(reserved)	11		
MEM_WR_LATENCY	13:12	0	Latency of write data after write command. 0 = 0 clocks 1 = 1/2 clocks 2 = 1 clock
MEM_WDOE_CNTL	15:14	0	Control of when to drive write data bus relative to write command. 0 = 1 clock before 1 = 1/2 clock before 2 = 0 clocks before 3 = 1/2 clock after
MEM_OPER_MODE	19:16	0	Operating mode of the sequencer. 0 = Normal 1 = Page Hiding Disabled
MEM_CTLR_STATUS **(R)**	20	0	Memory Controller busy indicator. 0 = Idle 1 = Busy

(Continued)

MEM_CNTL [RW] 32-bits Access: 8/16/32			MMR: 140, MMR_1: 140, IND: 140
Field Name	**Bits**	**Default**	**Description**
MEM_SEQNCR_STATUS **(R)**	21	0	Memory Controller's sequencer busy indicator. 0 = Idle 1 = Busy
MEM_ARBITER_STATUS (R)	22	0	Memory Controller's arbiter busy indicator. 0 = Idle 1 = Busy
MEM_REQ_LOCK	23	0	Locks out new client requests from being accepted by the memory controller. 0 = Unlocked 1 = Lock Out Requestors
MEM_EXTND_ERST	24	0	Extend internal ERST signal an additional clock cycle. Only relevant when memory configuration type is SDR2:1 or DDR. 0 = No Extension 1 = Extend
MEM_EXTND_DREN	25	0	Extend internal DREN signal an additional clock cycle. Only relevant when memory configuration type is SDR2:1 or DDR. 0 = No Extension 1 = Extend
MEM_DQM_RD_DIS	26	0	Disable assertion of DQM for read commands. 0 = enabled 1 = disabled
MEM_REFRESH_DIS	27	1	Disable refresh cycles. Must be turned OFF in shared configurations! 0 = Enable 1 = Disable

(Continued)

MEM_CNTL [RW] 32-bits Access: 8/16/32			MMR: 140, MMR_1: 140, IND: 140
Field Name	**Bits**	**Default**	**Description**
MEM_REFRESH_RATE	31:28	0	Refresh cycle rate set depending on XCLK frequency. 0 = 10 MHz - 50 MHz (1 refresh every 156 XCLK's) 1 = 50 MHz - 66 MHz (1 refresh every 781 XCLK's) 2 = 66 MHz - 75 MHz (1 refresh every 1031 XCLK's) 3 = 75 MHz - 83 MHz (1 refresh every 1172 XCLK's) 4 = 83 MHz - 90 MHz (1 refresh every 1297 XCLK's) 5 = 90 MHz - 95 MHz (1 refresh every 1406 XCLK's) 6 = 95 MHz - 100 MHz (1 refresh every 1484 XCLK's) 7 =100 MHz - 105 MHz (1 refresh every 1563 XCLK's) 8 =105 MHz - 110 MHz (1 refresh every 1641 XCLK's) 9 =110 MHz - 115 MHz (1 refresh every 1719 XCLK's) 10 =115 MHz - 120 MHz (1 refresh every 1797 XCLK's) 11 =120 MHz - 125 MHz (1 refresh every 1875 XCLK's) 12 =125 MHz and above (1 refresh every 1953 XCLK's)

Description:

Memory Control Register.

EXT_MEM_CNTL [RW] 32-bits Access: 8/16/32			MMR: 144, MMR_1: 144, IND: 144
Field Name	**Bits**	**Default**	**Description**
MEM_TRP	1:0	3	RAS precharge time, or PRE to ACTV time: 0 = 1 clock 1 = 2 clocks 2 = 3 clocks 3 = 4 clocks
MEM_TRCD	3:2	3	RAS to CAS delay, or ACTV to CMD time: 0 = 1 clock 1 = 2 clocks 2 = 3 clocks 3 = 4 clocks
MEM_TRAS	6:4	7	RAS low minimum pulse width, or ACTV to PRE same bank: 0 = 1 clock 1 = 2 clocks 2 = 3 clocks 3 = 4 clocks 4 = 5 clocks 5 = 6 clocks 6 = 7 clocks 7 = 8 clocks
(reserved)	7		
MEM_TRRD	9:8	2	RAS to RAS delay, or ACTV to ACTV delay: 1 = 1 clock 2 = 2 clocks 3 = 3 clocks
MEM_TR2W	11:10	1	Read to write data turnaround clock cycles: 1 = 1 clock 2 = 2 clocks
MEM_TWR	13:12	1	Write recovery time: 0 = 0 clocks 1 = 1 clock 2 = 2 clocks 3 = 3 clocks
MEM_TBWC	14	1	Block write cycle time: 0 = 1 clock 1 = 2 clocks
MEM_TSML	15	1	Special mode register write latency: 0 = 1 clock 1 = 2 clocks

(Continued)

EXT_MEM_CNTL [RW] 32-bits Access: 8/16/32			MMR: 144, MMR_1: 144, IND: 144
Field Name	**Bits**	**Default**	**Description**
MEM_TR2R	17:16	0	Read to read data turnaround time of 2 different memory parts driving the same MD signals: 0 = 0 clocks 1 = 1 clock 2 = 2 clocks 3 = 3 clocks
(reserved)	27:18		
MEM_TW2R_MODE	28	0	Write to Read command delay: 0 = 1 clock 1 = Use MEM_TWR for write to read command delay
MEM_TEST_MODE	30:29	0	Test mode for memory controller. No test mode was actually implemented. 0 = Normal
(reserved)	31		

Description:

Extended Memory Control Register.

MEM_INTF_CNTL [RW] 32-bits Access: 8/16/32			MMR: 14C, MMR_1: 14C, IND: 14C
Field Name	**Bits**	**Default**	**Description**
MEM_SSTL_EN	0	0	LVTTL/SSTL interface. 0 = LVTTL interface 1 = SSTL interface
MEM_MA_YCLK	1	0	Propagate memory address signals off of the falling edge of YCLK. 0 = propagate off of XCLK 1 = propagate off of YCLKb
MEM_CNTL_YCLK	2	0	Propagate RAS/CAS/WE/DSF signals off of the falling edge of YCLK. 0 = propagate off of XCLK 1 = propagate off of YCLKb

(Continued)

MEM_INTF_CNTL [RW] 32-bits Access: 8/16/32			MMR: 14C, MMR_1: 14C, IND: 14C
Field Name	**Bits**	**Default**	**Description**
MEM_CS_YCLK	3	0	Propagate CS signals off of the falling edge of YCLK. 0 = propagate off of XCLK 1 = propagate off of YCLKb
MEM_HCLK0_DRIVE	4	0	Drive strength of HCLK0 pin. 0 = low drive strength 1 = high drive strength
MEM_HCLK1_DRIVE	5	0	Drive strength of HCLK1 pin. 0 = low drive strength 1 = high drive strength
MEM_MA_DRIVE	6	0	Drive strength of memory address pins. 0 = low drive strength 1 = high drive strength
MEM_CNTL_DRIVE	7	0	Drive strength of RAS/CAS/WE/DSF pins. 0 = low drive strength 1 = high drive strength
MEM_CS_DRIVE	8	0	Drive strength of CS pins. 0 = low drive strength 1 = high drive strength
MEM_QS_DRIVE	9	0	Drive strengths of QS pins. 0 = low drive strength 1 = high drive strength
MEM_DQML_DRIVE	10	0	Drive strength of DQM(7:0) pins. 0 = low drive strength 1 = high drive strength
MEM_DQMU_DRIVE	11	0	Drive strength of DQM(15:8) pins. 0 = low drive strength 1 = high drive strength
MEM_MDLE_DRIVE	12	0	Drive strength of even pins of MD(63:0). 0 = low drive strength 1 = high drive strength
MEM_MDLO_DRIVE	13	0	Drive strength of odd pins of MD(63:0). 0 = low drive strength 1 = high drive strength
MEM_MDUE_DRIVE	14	0	Drive strength of even pins of MD(127:64). 0 = low drive strength 1 = high drive strength

(Continued)

MEM_INTF_CNTL [RW] 32-bits Access: 8/16/32			MMR: 14C, MMR_1: 14C, IND: 14C
Field Name	**Bits**	**Default**	**Description**
MEM_MDUO_DRIVE	15	0	Drive strength of odd pins of MD(127:64). 0 = low drive strength 1 = high drive strength
MEM_QS_REC	16	0	Receiver mode of QS pins. 0 = hysteresis receiver 1 = differential receiver
MEM_MD_REC	17	0	Receiver mode of MD pins. 0 = hysteresis receiver 1 = differential receiver
(reserved)	31:18		

Description:

Memory Interface Control Signals.

MEM_STR_CNTL [RW] 32-bits Access: 8/16/32			MMR: 150, MMR_1: 150, IND: 150
Field Name	**Bits**	**Default**	**Description**
STR0_SEL	2:0	0	Strobe signal for MD(31:0) 0 = positive edge of XCLK 1 = negative edge of XCLK 2 = HCLK0 feedback 3 = HCLK1 feedback 4 = HCLK0 feedback by 2 5 = HCLK1 feedback by 2 6 = QS0 delayed 7 = QS0 direct from pad
(reserved)	3		

(Continued)

MEM_STR_CNTL [RW] 32-bits Access: 8/16/32			MMR: 150, MMR_1: 150, IND: 150
Field Name	**Bits**	**Default**	**Description**
STR1_SEL	6:4	0	Strobe signal for MD(63:32). 0 = positive edge of XCLK 1 = negative edge of XCLK 2 = HCLK0 feedback 3 = HCLK1 feedback 4 = HCLK0 feedback by 2 5 = HCLK1 feedback by 2 6 = QS1 delayed 7 = QS1 direct from pad
(reserved)	7		
STR2_SEL	10:8	0	Strobe signal for MD(95:64). 0=positive edge of XCLK 1=negative edge of XCLK 2=HCLK0 feedback 3=HCLK1 feedback
(reserved)	11		
STR3_SEL	14:12	0	Strobe signal for MD(127:96). 0 = positive edge of XCLK 1 = negative edge of XCLK 2 = HCLK0 feedback 3 = HCLK1 feedback
(reserved)	15		
HCLK0_FB_SKEW	18:16	0	Programmable delay of feedback signal selected by HCLK0_FB_SEL. Only has effect when HCLK0 feedback or HCLK0 feedback by 2 is selected as a read data strobe.
(reserved)	19		Pin to use as HCLK0 feedback signal. 0 = HCLK0 pin 1 = QS0 pin
HCLK1_FB_SKEW	22:20	0	Programmable delay of feedback signal selected by HCLK1_FB_SEL. Only has effect when HCLK1 feedback or HCLK1 feedback by 2 is selected as a read data strobe.
HCLK1_FB_SEL	23	0	Pin to use as HCLK1 feedback signal. 0 = HCLK1 pin 1 = QS1 pin
(reserved)	31:24	0	

Description:

Memory Read Data Strobe Control.

MEM_INIT_LAT_TIMER [RW] 32-bits Access: 8/16/32			MMR: 154, MMR_1: 154, IND: 154
Field Name	**Bits**	**Default**	**Description**
MEM_PC0R_INIT_LAT	5:0	0	Initial Latency for PC0R request.
MEM_PC0W_INIT_LAT	11:6	0	Initial Latency for PC0W request.
MEM_PC1R_INIT_LAT	17:12	0	Initial Latency for PC1R request.
MEM_PC1W_INIT_LAT	23:18	0	Initial Latency for PC1W request.
MEM_TEXEL_INIT_LAT	29:24	0	Initial Latency for Texel request.
(reserved)	31:30		

Description:

Initial latency timer for memory controller arbiter.

MEM_SDRAM_MODE_REG [RW] 32-bits Access: 8/16/32			MMR: 158, MMR_1: 158, IND: 158
Field Name	**Bits**	**Default**	**Description**
MEM_MODE_REG	13:0	0	Value programmed into SDRAM mode register when SDRAM reset sequence is initiated.
(reserved)	15:14		
MEM_BURST_LENGTH	18:16	0	SDRAM burst length. 1 = 2 2 = 4 3 = 8
MEM_BURST_MODE	19	0	SDRAM burst mode. 0 = Sequential 1 = Interleaved

(Continued)

MEM_SDRAM_MODE_REG [RW] 32-bits Access: 8/16/32			MMR: 158, MMR_1: 158, IND: 158
Field Name	**Bits**	**Default**	**Description**
MEM_CAS_LATENCY	22:20	3	SDRAM CAS Latency. 2 = 2 3 = 3
(reserved)	30:23		
MEM_SDRAM_RESET	31	0	Initiate SDRAM reset sequence on a 0 to 1 transition of this register bit. 0 = Normal 1 = Reset

Description:

SDRAM Mode Register Control.

6.6 DAC Control Registers

DAC_CNTL [RW] 32-bits Access: 8/16/32			MMR: 58, MMR_1: 58, IOR: 58
Field Name	**Bits**	**Default**	**Description**
DAC_RANGE_CNTL	1:0	0	DAC control bits. Should be set to '10' by default. BIOS will modify if needed.
DAC_BLANKING	2	0	Controls use of DAC blanking pedestal during horizontal and vertical blanks. 0 = 0 IRE blanking pedestal 1 = Enable 7.5 IRE blanking pedestal. Increases display brightness relative to blanking regions.
DAC_COMP_EN	3	0	<No Description>
(reserved)	6:4		
DAC_CMP_OUTPUT **(R)**	7	0	DAC comparator output. 0 = At least 1 comparator > ~0.373V. 1 = All 3 comparators < ~0.373V. The comparators are used for monitor detection by sensing if the termination on the R,G&B lines is 75 ohms (no monitor) or 37.5 ohms (monitor present). This can also determine if the attached monitor is monochrome or color. To use this register the driver must ensure the raster is currently in the active display area. Reading multiple times is recommended. To test if Green is terminated, set Red and Blue to 0 and set Green to 5A (post palette). If the Green line is terminated, then DAC_CMP_OUTPUT will read back '1' when the raster is on the above color. See the programmers manual for more details on the monitor detection algorithm
DAC_8BIT_EN	8	0	Enables 8 bit DAC operation. 8 bit is normal, 6 bit used for VGA emulation. When in 6 bit writes and reads to DAC_DATA and PALETTE_DATA are affected. Writes shift 6 bits left by 2 to make 8 bits in the palette memory. Reads shift 8 bit palette data right by 2 to give 6 MSBs to the host. 0 = 6 bit 1 = 8 bit

(Continued)

DAC_CNTL [RW] 32-bits Access: 8/16/32			MMR: 58, MMR_1: 58, IOR: 58
Field Name	**Bits**	**Default**	**Description**
DAC_4BPP_PIX_ORDER	9	0	Selects the order of pixel nibbles within bytes for 4 bpp extended (non-VGA) display modes. 0 = Most significant nibble is the left pixel. 1 = Least significant nibble is the left pixel.
DAC_TVO_EN	10	0	Enables generation of TV output byte stream for use by ImpacTV/Ripper encoder. Use the MPP_TB_TVO_EN or MPP_GP_TVO_EN bits to control which MPP port drives out the display data. This depends on the board design. The display clock generation must also be programmed for TV out to get an image on the TV. 0 = Disable 1 = Enable
DAC_TVO_OVR_EXCL	11	0	Used when TV out active to suppress overscan on the CRT monitor. Overscan is used by the TV out circuitry for frame synchronization. 0 = CRT & TVO overscan 1 = TVO overscan only
DAC_TVO_16BPP_DITH_EN	12	0	Selects method of encoding TV out data when using 565 mode. Dither method is one dimensional error diffusion. 0 = Disable 1 = Enable Dithering on 16BPP TV Output
DAC_VGA_ADR_EN	13	0	Enables access of the palette (DAC) at the VGA I/O DAC addresses when in extended display modes (non-VGA, or CRTC_EXT_DISP_EN=1).
(reserved)	14		
DAC_PDWN	15	0	Power down internal DAC (DAC macro only). This does not affect the digital outputs (TV or flat panel). The DAC is automatically powered down when the PMI_POWER_STATE register is not in the D0 state. This should save about 56 mA.
(reserved)	18:16		

(Continued)

DAC_CNTL [RW] 32-bits Access: 8/16/32			MMR: 58, MMR_1: 58, IOR: 58
Field Name	**Bits**	**Default**	**Description**
DAC_CRC_EN	19	0	Enables the CRC signature check on the data going to the DAC macro. This is what appears on the screen, and includes graphics, HW cursor, video overlay, sub-picture, and overscan. 0 = Disable. Reset before using. 1 = Enable. CRC will start in next vertical blank, and run for one field/frame. **NOTE:** There is no hardware control of whether the CRC occurs on evan or odd frames in interlaced modes. This can be done by software polling the CRTC_FRAME and CRTC_CRNT_VLINE registers before enabling the CRC. The CRC's for even and odd frames will be different.
(reserved)	23:20		
DAC_MASK	31:24	ff	Masks off usage of individual palette index bits before pixel index is looked-up in the palette. 0 = do not use this bit of the index 1 = use this bit of the index This is a mirror of the VGA DAC_MASK register. It only has an effect in VGA emulation modes (CRTC_EXT_DISP_EN=0), not for VESA modes or extended display modes.

Description:

General control for the RGB DAC and palette.

DAC_CRC_SIG [R] 32-bits Access: 8/16/32			MMR: 2CC, MMR_1: 2CC, IND: 2CC
Field Name	**Bits**	**Default**	**Description**
DAC_CRC_SIG	23:0	0	DAC CRC signature value. Use DAC_CRC_EN to initiated a field or frame analysis. After completion of the field/frame the DAC_CRC_SIG will remain constant until DAC_CRC_EN is cleared and set again. Only the even or odd field of interlaced displays is CRC'ed at one time. This is the code for the CRC signature: CRCB(7:0) <= 0;CRCG(7:0) <= 0;CRCR(7:0) <= 0; While in frame to capture and not blank do once per pixel: CRCB(7:1) <= Blue(7:1) x or CRCB(6:0); CRCB(0) <= (Blue(0) x or CRCB(0)) x or (CRCB(7) x or CRCG(7)); CRCG(7:1) <= Green(7:1) x or CRCG(6:0); CRCG(0) <= (Green(0) x or CRCG(0)) x or (CRCG(7) x or CRCR(7)); CRCR(7:1) <= Red(7:1) x or CRCR(6:0); CRCR(0) <= Red(0) x or (CRCR(0) x or CRCR(7)); End do; DAC_CRC_SIG(23:0) <= CRCB(7:0) & CRCG(7:0) & CRCR(7:0);
(reserved)	31:24		

Description:

CRC signature value.

PALETTE_INDEX [RW] 32-bits Access: 8/16/32			MMR: B0, MMR_1: B0, IOR: B0, IND: B0
Field Name	**Bits**	**Default**	**Description**
PALETTE_W_INDEX	7:0	0	Write: Sets starting index for palette writes. Auto-increments on each write to PALETTE_DATA. Read: Indicates index where next write to PALETTE_DATA will be written.
(reserved)	15:8		

(Continued)

PALETTE_INDEX [RW] 32-bits Access: 8/16/32			MMR: B0, MMR_1: B0, IOR: B0, IND: B0
Field Name	**Bits**	**Default**	**Description**
PALETTE_R_INDEX	23:16	0	Write: Sets starting index for palette reads. Auto-increments on each read from PALETTE_DATA. Read: Indicates index where next read from PALETTE_DATA will be read.
(reserved)	31:24		

Description:

Display palette read and write index setting. Recommend using byte writes to set either read mode or write mode for the palette.

PALETTE_DATA [RW] 32-bits Access: 8/16/32			MMR: B4, MMR_1: B4, IOR: B4, IND: B4
Field Name	**Bits**	**Default**	**Description**
PALETTE_DATA_B	7:0	0	Blue palette data.
PALETTE_DATA_G	15:8	0	Green palette data.
PALETTE_DATA_R	23:16	0	Red palette data.
(reserved)	31:24		

Description:

Display palette data read/write.

7.1 Destination Registers

DST_OFFSET [RW] 32-bits Access: 32			MMR: 1404, MMR_1: 1404, IND: 1404
Field Name	**Bits**	**Default**	**Description**
DST_OFFSET	25:0	0	Byte-aligned destination offset address. This is a virtual address. The lower 32MB maps to frame buffer, the upper 32MB to AGP_BASE + DST_OFFSET(24:0). Note that this register is contained to 128 bit alignment. NOTE: Bits 3:0 of this field are hardwired to ZERO
(reserved)	31:26		

DST_PITCH_OFFSET [W] 32-bits Access: 32			MMR: 142C, MMR_1: 142C, IND: 142C
Field Name	**Bits**	**Default**	**Description**
DST_OFFSET	20:0	0	32 byte-aligned destination offset address.
DST_PITCH	30:21	0	Destination pitch in pixels*8. Note that for monochrome modes the destination pitch must be a multiple of 64 pixels.
DST_TILE	31	0	Destination tile bit.

DST_PITCH [RW] 32-bits Access: 32			MMR: 1408, MMR_1: 1408, IND: 1408
Field Name	**Bits**	**Default**	**Description**
DST_PITCH	9:0	0	Destination pitch in pixels*8. Note that for monochrome modes the destination pitch must be a multiple of 64 pixels.
(reserved)	15:10		
DST_TILE	16	0	Denotes whether the destination surface is in 'tiled' format.
DST_PITCH_ADJ	18:17	0	Denotes that DST_PITCH should be multiplied prior to use:
(reserved)	31:19		

DST_X [RW] 32-bits Access: 32			MMR: 141C, MMR_1: 141C, IND: 141C
Field Name	**Bits**	**Default**	**Description**
DST_X	13:0	0	Destination X co-ordinate (range -8192 to 8192) If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.12.0 bit integer with bits 14:13 reserved.
(reserved)	31:14		

DST_Y [RW] 32-bits Access: 32			MMR: 1420, MMR_1: 1420, IND: 1420
Field Name	**Bits**	**Default**	**Description**
DST_Y	13:0	0	Destination Y coordinate (range -8192 to 8192) - Bits 15:14 should be copies of bit 13 (i.e. sign extended). If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.14.0 integer.
(reserved)	31:14		

DST_X_Y [W] 32-bits Access: 32			MMR: 1594, MMR_1: 1594, IND: 1594
Field Name	**Bits**	**Default**	**Description**
DST_Y	13:0	0	Destination Y coordinate (range -8192 to 8192) - Bits 15:14 should be copies of bit 13 (i.e. sign extended). If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.14.0 integer.
(reserved)	15:14		
DST_X	29:16	0	Destination X co-ordinate (range -8192 to 8192) If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.12.0 bit integer with bits 14:13 reserved.
(reserved)	31:30		

DST_Y_X [W] 32-bits Access: 32			MMR: 1438, MMR_1: 1438, IND: 1438
Field Name	**Bits**	**Default**	**Description**
DST_X	13:0	0	If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.12.0 bit integer with bits 14:13 reserved.
(reserved)	15:14		
DST_Y	29:16	0	If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.14.0 integer.
(reserved)	31:30		

DST_WIDTH [RW] 32-bits Access: 32			MMR: 140C, MMR_1: 140C, IND: 140C
Field Name	**Bits**	**Default**	**Description**
DST_WIDTH	13:0	0	Destination width. Only bits 12:0 are used for rectangle draws. Bit 15 is write ONLY and will always read back as '0'. Bits [15:13] are aliased to DST_BRES_LENGTH[15:13] and are used for trapezoid draw operations.
(reserved)	31:14		

DST_HEIGHT [RW] 32-bits Access: 32			MMR: 1410, MMR_1: 1410, IND: 1410
Field Name	**Bits**	**Default**	**Description**
DST_HEIGHT	13:0	0	<No Description>
(reserved)	31:14		

DST_HEIGHT_WIDTH [W] 32-bits Access: 32			MMR: 143C, MMR_1: 143C, IND: 143C
Field Name	**Bits**	**Default**	**Description**
DST_WIDTH	13:0	0	<No Description>
(reserved)	15:14		
DST_HEIGHT	29:16	0	<No Description>
(reserved)	31:30		

DST_HEIGHT_WIDTH_BW [W] 32-bits Access: 32			MMR: 15B4, MMR_1: 15B4, IND: 15B4
Field Name	**Bits**	**Default**	**Description**
DST_WIDTH	13:0	0	<No Description>
(reserved)	15:14		
DST_HEIGHT	29:16	0	<No Description>
(reserved)	31:30		

Usage:

A write to this register indicates all alignment conditions (x, width, scissors,...) have been met to do a block write fill. It is valid for all memory types, but it is of most value (e.g., Z-clears) in non-byte-maskable memories where, block writes are disabled except when writes to this register occur.Note: this is an initiator register.

DST_WIDTH_HEIGHT [W] 32-bits Access: 32			MMR: 1598, MMR_1: 1598, IND: 1598
Field Name	**Bits**	**Default**	**Description**
DST_HEIGHT	13:0	0	Destination height (bits 12:0 aliased to TRAIL_X@DST_BRES_LNTH)
(reserved)	15:14		
DST_WIDTH	29:16	0	Destination width. Only bits 12:0 are used for rectangle draws. Bit 15 is write ONLY and will always read back as ‘0’. Bits [15:13] are aliased to DST_BRES_LENGTH[15:13] and are used for trapezoid draw operations.
(reserved)	31:30		

DST_HEIGHT_WIDTH_8 [W] 32-bits Access: 32			MMR: 158C, MMR_1: 158C, IND: 158C
Field Name	**Bits**	**Default**	**Description**
(reserved)	15:0		
DST_WIDTH	23:16	0	<No Description>
DST_HEIGHT	31:24	0	<No Description>

DST_HEIGHT_Y [W] 32-bits Access: 32			MMR: 15A0, MMR_1: 15A0, IND: 15A0
Field Name	**Bits**	**Default**	**Description**
DST_Y	13:0	0	<No Description>
(reserved)	15:14		
DST_HEIGHT	29:16	0	<No Description>
(reserved)	31:30		

DST_WIDTH_X [W] 32-bits Access: 32			MMR: 1588, MMR_1: 1588, IND: 1588
Field Name	**Bits**	**Default**	**Description**
DST_X	13:0	0	Destination X coordinate. If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.12.0 bit integer with bits 14:13 reserved.
(reserved)	15:14		
DST_WIDTH	29:16	0	Destination width. Only bits 12:0 are used for rectangle draws. Bit 15 is write ONLY and will always read back as ‘0’. Bits [15:13] are aliased to DST_BRES_LENGTH[15:13] and are used for trapezoid draw operations.
(reserved)	31:30		

DST_WIDTH_X_INCY [W] 32-bits Access: 32			MMR: 159C, MMR_1: 159C, IND: 159C
Field Name	**Bits**	**Default**	**Description**
DST_X	13:0	0	Destination X coordinate. If SUB_PIX_ON is set, this field is interpreted as a S.12.2 number. Otherwise it is a S.12.0 bit integer with bits 14:13 reserved.
(reserved)	15:14		
DST_WIDTH	29:16	0	Destination width. Only bits 12:0 are used for rectangle draws. Bit 15 is write ONLY and will always read back as '0'. Bits [15:13] are aliased to DST_BRES_LENGTH[15:13] and are used for trapezoid draw operations.
(reserved)	31:30		

DST_BRES_LNTH [W] 32-bits Access: 32			MMR: 1634, MMR_1: 1634, IND: 1634
Field Name	**Bits**	**Default**	**Description**
DST_BRES_LNTH	13:0	0	Bresenham line, and Trapezoid leading edge length. This field is aliased with DST_WIDTH[14:0].
(reserved)	31:14		

DST_BRES_ERR [RW] 32-bits Access: 32			MMR: 1628, MMR_1: 1628, IND: 1628
Field Name	**Bits**	**Default**	**Description**
DST_BRES_ERR	19:0	0	Bresenham error term for line and Trapezoid leading edge
(reserved)	31:20		

DST_BRES_INC [RW] 32-bits Access: 32			MMR: 162C, MMR_1: 162C, IND: 162C
Field Name	**Bits**	**Default**	**Description**
DST_BRES_INC	19:0	0	Bresenham increment for line and Trapezoid leading edge
(reserved)	31:20		

DST_BRES_DEC [RW] 32-bits Access: 32			MMR: 1630, MMR_1: 1630, IND: 1630
Field Name	**Bits**	**Default**	**Description**
DST_BRES_DEC	19:0	0	Bresenham decrement for line and Trapezoid leading edge
(reserved)	31:20		

DST_X_SUB [RW] 32-bits Access: 32			MMR: 15A4, MMR_1: 15A4, IND: 15A4
Field Name	**Bits**	**Default**	**Description**
LEAD_X_FRACT	3:0	0	<No Description>
LEAD_X	17:4	0	<No Description>
(reserved)	31:18		

DST_Y_SUB [RW] 32-bits Access: 32			MMR: 15A8, MMR_1: 15A8, IND: 15A8
Field Name	**Bits**	**Default**	**Description**
LEAD_Y_FRACT	3:0	0	<No Description>
LEAD_Y	17:4	0	<No Description>
(reserved)	31:18		

DST_WIDTH_BW [W] 32-bits Access: 32			MMR: 15B4, MMR_1: 15B4, IND: 15B4
Field Name	**Bits**	**Default**	**Description**
DST_WIDTH	31:0	0	Destination width
(reserved)	31:14		

Description:

This is an initiator register. A write to this register indicates that all alignment conditions to do a block write fill (x, width, scissors,...) have been met. It is valid for all memory types, but it is of most value (e.g., Z-clears) in non-byte-maskable memories where block writes are disabled except when writes to this register occur.

DST_BRES_LNTH_SUB [RW] 32-bits Access: 32			MMR: 1638, MMR_1: 1638, IND: 1638
Field Name	**Bits**	**Default**	**Description**
DST_BRES_LNTH_SUB	3:0	0	<No Description>
DST_BRES_LNTH	17:4	0	Bresenham line, and Trapezoid leading edge length. This field is aliased with DST_WIDTH[14:0].
(reserved)	31:18		

COMPOSITE_SHADOW_ID [RW] 32-bits Access: 32			MMR: 1A0C, MMR_1: 1A0C, IND: 1A0C
Field Name	**Bits**	**Default**	**Description**
COMPOSITE_SHADOW_ID	23:0	0	This field is a count of 3D primitives executed. It is used as part of the shadow ID algorithm, but may also be used as a general counter for performance purposes.
(reserved)	31:24		

Desription:

Triangle count for shadow algorithm.

DST_PITCH_OFFSET_C [W] 32-bits Access: 32			MMR: 1C80, MMR_1: 1C80, IND: 1C80
Field Name	**Bits**	**Default**	**Description**
DST_OFFSET	20:0	0	<No Description>
DST_PITCH	30:21	0	<No Description>
DST_TILE	31	0	<No Description>

LEAD_BRES_ERR [W] 32-bits Access: 32			MMR: 1600, MMR_1: 1600, IND: 1600
Field Name	**Bits**	**Default**	**Description**
LEAD_BRES_ERR	19:0	0	Bresenham error term for trapezoid leading edge.
(reserved)	31:20		

LEAD_BRES_INC [W] 32-bits Access: 32			MMR: 1604, MMR_1: 1604, IND: 1604
Field Name	**Bits**	**Default**	**Description**
LEAD_BRES_INC	19:0	0	Bresenham increment for trapezoid leading edge.
(reserved)	31:20		

LEAD_BRES_DEC [W] 32-bits Access: 32			MMR: 1608, MMR_1: 1608, IND: 1608
Field Name	**Bits**	**Default**	**Description**
LEAD_BRES_DEC	19:0	0	Bresenham decrement for trapezoid leading edge.
(reserved)	31:20		

LEAD_BRETH_LNTH [W] 32-bits Access: 32			MMR: 161C, MMR_1: 161C, IND: 161C
Field Name	**Bits**	**Default**	**Description**
LEAD_BRES_LNTH	13:0	0	Trapezoid leading edge length.
(reserved)	31:14		

TRAIL_BRES_ERR [RW] 32-bits Access: 32			MMR: 160C, MMR_1: 160C, IND: 160C
Field Name	**Bits**	**Default**	**Description**
TRAIL_BRES_ERR	19:0	0	Bresenham error term for trapezoid trailing edge.
(reserved)	31:20		

TRAIL_BRES_INC [RW] 32-bits Access: 32			MMR: 1610, MMR_1: 1610, IND: 1610
Field Name	**Bits**	**Default**	**Description**
TRAIL_BRES_INC	19:0	0	Bresenham increment for line and Trapezoid trailing edge.
(reserved)	31:20		

TRAIL_BRES_DEC [RW] 32-bits Access: 32			MMR: 1614, MMR_1: 1614, IND: 1614
Field Name	**Bits**	**Default**	**Description**
TRAIL_BRES_DEC	19:0	0	Bresenham decrement for line and Trapezoid trailing edge.
(reserved)	31:20		

TRAIL_X [RW] 32-bits Access: 32			MMR: 1618, MMR_1: 1618, IND: 1618
Field Name	**Bits**	**Default**	**Description**
TRAIL_X	13:0	0	X for trapezoid trailing edge.
(reserved)	31:14		

TRAIL_X_SUB [RW] 32-bits Access: 32			MMR: 1620, MMR_1: 1620, IND: 1620
Field Name	**Bits**	**Default**	**Description**
TRAIL_X_FRACT	3:0	0	Sub pixel bits of TRAIL_X coordinate. Note that when TRAIL_X is written these bits are set to 1000 (one half).
TRAIL_X	17:4	0	Trailing edge X coordinate: range -8192 to 8191. Aliased to TRAIL_X[13:0].
(reserved)	31:18		

LEAD_BRETH_LNTH_SUB [RW] 32-bits Access: 32			MMR: 1624, MMR_1: 1624, IND: 1624
Field Name	**Bits**	**Default**	**Description**
LEAD_BRES_LNTH_SUB	3:0	0	Trapezoid leading edge length.
LEAD_BRES_LNTH	17:4	0	Trapezoid leading edge length. Aliased to DST_BRES_LNTH[13:0].
(reserved)	31:18		

7.2 GIU Source Registers

SRC_OFFSET [RW] 32-bits Access: 32			MMR: 15AC, MMR_1: 15AC, IND: 15AC
Field Name	**Bits**	**Default**	**Description**
SRC_OFFSET	25:0	0	Source offset address in terms of 64 bit words. NOTE: Bits 3:0 of this field are hardwired to ZERO
(reserved)	31:26		

SRC_PITCH_OFFSET [W] 32-bits Access: 32			MMR: 1428, MMR_1: 1428, IND: 1428
Field Name	**Bits**	**Default**	**Description**
SRC_OFFSET	20:0	0	Source offset address in terms of 64 bit words.
SRC_PITCH	30:21	0	Source pitch in pixelsx8. Note that in monochrome mode the source pitch must be a multiple of 64 pixels
SRC_TILE	31	0	<No Description>

SRC_PITCH [RW] 32-bits Access: 32			MMR: 15B0, MMR_1: 15B0, IND: 15B0
Field Name	**Bits**	**Default**	**Description**
SRC_PITCH	9:0	0	Source pitch in pixelsx8. Note that in monochrome mode the source pitch must be a multiple of 64 pixel.
(reserved)	15:10		
SRC_TILE	16	0	<No Description>
(reserved)	31:17		

SRC_X [RW] 32-bits Access: 32			MMR: 1414, MMR_1: 1414, IND: 1414
Field Name	**Bits**	**Default**	**Description**
SRC_X	13:0	0	Source X coordinate
(reserved)	31:14		

SRC_Y [RW] 32-bits Access: 32			MMR: 1418, MMR_1: 1418, IND: 1418
Field Name	**Bits**	**Default**	**Description**
SRC_Y	13:0	0	Source Y coordinate
(reserved)	31:14		

SRC_X_Y [W] 32-bits Access: 32			MMR: 1590, MMR_1: 1590, IND: 1590
Field Name	**Bits**	**Default**	**Description**
SRC_Y	13:0	0	Source Y coordinate
(reserved)	15:14		
SRC_X	29:16	0	Source X coordinate
(reserved)	31:30		

SRC_Y_X [W] 32-bits Access: 32			MMR: 1434, MMR_1: 1434, IND: 1434
Field Name	**Bits**	**Default**	**Description**
SRC_X	13:0	0	Source X coordinate
(reserved)	15:14		
SRC_Y	29:16	0	Source Y coordinate
(reserved)	31:30		

SRC_SC_RIGHT [RW] 32-bits Access: 32			MMR: 1654, MMR_1: 1654, IND: 1654
Field Name	**Bits**	**Default**	**Description**
SRC_SC_RIGHT	13:0	0	<No Description>
(reserved)	31:14		

SRC_SC_BOTTOM [RW] 32-bits Access: 32			MMR: 165C, MMR_1: 165C, IND: 165C
Field Name	**Bits**	**Default**	**Description**
SRC_SC_BOTTOM	13:0	0	<No Description>
(reserved)	31:14		

SRC_SC_BOTTOM_RIGHT [W] 32-bits Access: 32			MMR: 16F4, MMR_1: 16F4, IND: 16F4
Field Name	**Bits**	**Default**	**Description**
SRC_SC_RIGHT	13:0	0	<No Description>
(reserved)	15:14		
SRC_SC_BOTTOM	29:16	0	<No Description>
(reserved)	31:30		

7.3 GUI Host Data Registers

The host data registers provide pixel data which are utilized in the current drawing operation. The pixel data may be used as a monochrome pixel source or color pixel source. For rectangular drawing operations the pixel data may be either packed from one horizontal line to the next or unpacked. Sixteen 32 bit host data registers are provided. All registers are treated identically and data is fed to the engine in the order in which it is written to any of the host data registers. Up to sixteen host data registers are provided to allow block data moves of variable length up to the depth of the parameter FIFO.

Note: This table represents 8 registers: HOST_DATA0 to HOST_DATA7

HOST_DATA[7:0] [W] 32-bits Access: 32			MMR: 17C0-17DC, MMR_1: 17C0-17DC, IND: 17C0-17DC
Field Name	**Bits**	**Default**	**Description**
HOST_DATA[7:0]	31:0	0	Host data register

Desription:

Host data register.

HOST_DATA_LAST [RW] 32-bits Access: 32			MMR: 17E0, MMR_1: 17E0, IND: 17E0
Field Name	**Bits**	**Default**	**Description**
HOST_DATA_LAST	31:0	0	

7.4 Pattern Registers

Pattern registers 0 - 63. Pattern register 0 is used for 32x1, 8x1, 1x8 mono cases. Pattern registers 0-1 are used for 8x8 mono case. Pattern registers 0-15 are used for 8bpp color. Pattern registers 0-31 are used for 16bpp color, 32x32 mono cases. Pattern registers 0-63 are used for 24/32bpp color (24bpp not packed). Pattern registers 0-1 are used for 8x1, 1x8 8bpp color. Pattern registers are used 0-3 for 8x1, 1x8 16bpp color. Pattern registers 0-7 are used for 8x1, 1x8 24/32bpp color (24bpp not packed).

BRUSH_DATA0 [RW] 32-bits Access: 32			MMR: 1480, MMR_1: 1480, IND: 1480
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA0	31:0	0	<No Description>

BRUSH_DATA1 [RW] 32-bits Access: 32			MMR: 1484, MMR_1: 1484, IND: 1484
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA1	31:0	0	<No Description>

BRUSH_DATA2 [RW] 32-bits Access: 32			MMR: 1488, MMR_1: 1488, IND: 1488
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA2	31:0	0	<No Description>

BRUSH_DATA3 [RW] 32-bits Access: 32			MMR: 148C, MMR_1: 148C, IND: 148C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA3	31:0	0	<No Description>

BRUSH_DATA4			MMR: 1490, MMR_1: 1490,
[RW] 32-bits Access: 32			IND: 1490
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA4	31:0	0	<No Description>

BRUSH_DATA5			MMR: 1494, MMR_1: 1494,
[RW] 32-bits Access: 32			IND: 1494
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA5	31:0	0	<No Description>

BRUSH_DATA6			MMR: 1498, MMR_1: 1498,
[RW] 32-bits Access: 32			IND: 1498
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA6	31:0	0	<No Description>

BRUSH_DATA7			MMR: 149C, MMR_1: 149C,
[RW] 32-bits Access: 32			IND: 149C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA7	31:0	0	<No Description>

BRUSH_DATA8			MMR: 14A0, MMR_1: 14A0,
[RW] 32-bits Access: 32			IND: 14A0
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA8	31:0	0	<No Description>

BRUSH_DATA9			MMR: 14A4, MMR_1: 14A4,
[RW] 32-bits Access: 32			IND: 14A4
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA9	31:0	0	<No Description>

BRUSH_DATA10			MMR: 14A8, MMR_1: 14A8,
[RW] 32-bits Access: 32			IND: 14A8
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA10	31:0	0	<No Description>

BRUSH_DATA11			MMR: 14AC, MMR_1: 14AC,
[RW] 32-bits Access: 32			IND: 14AC
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA11	31:0	0	<No Description>

BRUSH_DATA12			MMR: 14B0, MMR_1: 14B0,
[RW] 32-bits Access: 32			IND: 14B0
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA12	31:0	0	<No Description>

BRUSH_DATA13			MMR: 14B4, MMR_1: 14B4,
[RW] 32-bits Access: 32			IND: 14B4
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA13	31:0	0	<No Description>

BRUSH_DATA14			MMR: 14B8, MMR_1: 14B8,
[RW] 32-bits Access: 32			IND: 14B8
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA14	31:0	0	<No Description>

BRUSH_DATA15			MMR: 14BC, MMR_1: 14BC,
[RW] 32-bits Access: 32			IND: 14BC
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA15	31:0	0	<No Description>

BRUSH_DATA16			MMR: 14C0, MMR_1: 14C0,
[RW] 32-bits Access: 32			IND: 14C0
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA16	31:0	0	<No Description>

BRUSH_DATA17			MMR: 14C4, MMR_1: 14C4,
[RW] 32-bits Access: 32			IND: 14C4
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA17	31:0	0	<No Description>

BRUSH_DATA18			MMR: 14C8, MMR_1: 14C8,
[RW] 32-bits Access: 32			IND: 14C8
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA18	31:0	0	<No Description>

BRUSH_DATA19			MMR: 14CC, MMR_1: 14CC,
[RW] 32-bits Access: 32			IND: 14CC
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA19	31:0	0	<No Description>

BRUSH_DATA20			MMR: 14D0, MMR_1: 14D0,
[RW] 32-bits Access: 32			IND: 14D0
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA20	31:0	0	<No Description>

BRUSH_DATA21			MMR: 14D4, MMR_1: 14D4,
[RW] 32-bits Access: 32			IND: 14D4
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA21	31:0	0	<No Description>

BRUSH_DATA22			MMR: 14D8, MMR_1: 14D8,
[RW] 32-bits Access: 32			IND: 14D8
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA22	31:0	0	<No Description>

BRUSH_DATA23			MMR: 14DC, MMR_1: 14DC,
[RW] 32-bits Access: 32			IND: 14DC
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA23	31:0	0	<No Description>

BRUSH_DATA24			MMR: 14E0, MMR_1: 14E0,
[RW] 32-bits Access: 32			IND: 14E0
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA24	31:0	0	<No Description>

BRUSH_DATA25			MMR: 14E4, MMR_1: 14E4,
[RW] 32-bits Access: 32			IND: 14E4
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA25	31:0	0	<No Description>

BRUSH_DATA26			MMR: 14E8, MMR_1: 14E8,
[RW] 32-bits Access: 32			IND: 14E8
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA26	31:0	0	<No Description>

BRUSH_DATA27			MMR: 14EC, MMR_1: 14EC,
[RW] 32-bits Access: 32			IND: 14EC
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA27	31:0	0	<No Description>

BRUSH_DATA28			MMR: 14F0, MMR_1: 14F0,
[RW] 32-bits Access: 32			IND: 14F0
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA28	31:0	0	<No Description>

BRUSH_DATA29			MMR: 14F4, MMR_1: 14F4,
[RW] 32-bits Access: 32			IND: 14F4
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA29	31:0	0	<No Description>

BRUSH_DATA30			MMR: 14F8, MMR_1: 14F8
[RW] 32-bits Access: 32			IND: 14F8
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA30	31:0	0	<No Description>

BRUSH_DATA31			MMR: 14FC, MMR_1: 14FC,
[RW] 32-bits Access: 32			IND: 14FC
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA31	31:0	0	<No Description>

BRUSH_DATA32			MMR: 1500, MMR_1: 1500,
[RW] 32-bits Access: 32			IND: 1500
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA32	31:0	0	<No Description>

BRUSH_DATA33			MMR: 1504, MMR_1: 1504,
[RW] 32-bits Access: 32			IND: 1504
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA33	31:0	0	<No Description>

BRUSH_DATA34			MMR: 1508, MMR_1: 1508,
[RW] 32-bits Access: 32			IND: 1508
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA34	31:0	0	<No Description>

BRUSH_DATA35			MMR: 150C, MMR_1: 150C,
[RW] 32-bits Access: 32			IND: 150C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA35	31:0	0	<No Description>

BRUSH_DATA36			MMR: 1510, MMR_1: 1510,
[RW] 32-bits Access: 32			IND: 1510
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA36	31:0	0	<No Description>

BRUSH_DATA37			MMR: 1514, MMR_1: 1514,
[RW] 32-bits Access: 32			IND: 1514
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA37	31:0	0	<No Description>

BRUSH_DATA38			MMR: 1518, MMR_1: 1518,
[RW] 32-bits Access: 32			IND: 1518
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA38	31:0	0	<No Description>

BRUSH_DATA39			MMR: 151C, MMR_1: 151C,
[RW] 32-bits Access: 32			IND: 151C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA39	31:0	0	<No Description>

BRUSH_DATA40			MMR: 1520, MMR_1: 1520,
[RW] 32-bits Access: 32			IND: 1520
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA40	31:0	0	<No Description>

BRUSH_DATA41			MMR: 1524, MMR_1: 1524,
[RW] 32-bits Access: 32			IND: 1524
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA41	31:0	0	<No Description>

BRUSH_DATA42			MMR: 1528, MMR_1: 1528,
[RW] 32-bits Access: 32			IND: 1528
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA42	31:0	0	<No Description>

BRUSH_DATA43			MMR: 152C, MMR_1: 152C,
[RW] 32-bits Access: 32			IND: 152C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA43	31:0	0	<No Description>

BRUSH_DATA44			MMR: 1530, MMR_1: 1530,
[RW] 32-bits Access: 32			IND: 1530
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA44	31:0	0	<No Description>

BRUSH_DATA45			MMR: 1534, MMR_1: 1534,
[RW] 32-bits Access: 32			IND: 1534
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA45	31:0	0	<No Description>

BRUSH_DATA46			MMR: 1538, MMR_1: 1538,
[RW] 32-bits Access: 32			IND: 1538
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA46	31:0	0	<No Description>

BRUSH_DATA47			MMR: 153C, MMR_1: 153C,
[RW] 32-bits Access: 32			IND: 153C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA47	31:0	0	<No Description>

BRUSH_DATA48			MMR: 1540, MMR_1: 1540,
[RW] 32-bits Access: 32			IND: 1540
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA48	31:0	0	<No Description>

BRUSH_DATA49			MMR: 1544, MMR_1: 1544,
[RW] 32-bits Access: 32			IND: 1544
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA49	31:0	0	<No Description>

BRUSH_DATA50			MMR: 1548, MMR_1: 1548,
[RW] 32-bits Access: 32			IND: 1548
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA50	31:0	0	<No Description>

BRUSH_DATA51			MMR: 154C, MMR_1: 154C,
[RW] 32-bits Access: 32			IND: 154C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA51	31:0	0	<No Description>

BRUSH_DATA52			MMR: 1550, MMR_1: 1550,
[RW] 32-bits Access: 32			IND: 1550
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA52	31:0	0	<No Description>

BRUSH_DATA53			MMR: 1554, MMR_1: 1554,
[RW] 32-bits Access: 32			IND: 1554
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA53	31:0	0	<No Description>

BRUSH_DATA54			MMR: 1558, MMR_1: 1558,
[RW] 32-bits Access: 32			IND: 1558
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA54	31:0	0	<No Description>

BRUSH_DATA55			MMR: 155C, MMR_1: 155C,
[RW] 32-bits Access: 32			IND: 155C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA55	31:0	0	<No Description>

BRUSH_DATA56			MMR: 1560, MMR_1: 1560,
[RW] 32-bits Access: 32			IND: 1560
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA56	31:0	0	<No Description>

BRUSH_DATA57			MMR: 1564, MMR_1: 1564,
[RW] 32-bits Access: 32			IND: 1564
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA57	31:0	0	<No Description>

BRUSH_DATA58			MMR: 1568, MMR_1: 1568,
[RW] 32-bits Access: 32			IND: 1568
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA58	31:0	0	<No Description>

BRUSH_DATA59			MMR: 156C, MMR_1: 156C,
[RW] 32-bits Access: 32			IND: 156C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA59	31:0	0	<No Description>

BRUSH_DATA60			MMR: 1570, MMR_1: 1570,
[RW] 32-bits Access: 32			IND: 1570
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA60	31:0	0	<No Description>

BRUSH_DATA61			MMR: 1574, MMR_1: 1574,
[RW] 32-bits Access: 32			IND: 1574
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA61	31:0	0	<No Description>

BRUSH_DATA62			MMR: 1578, MMR_1: 1578,
[RW] 32-bits Access: 32			IND: 1578
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA62	31:0	0	<No Description>

BRUSH_DATA63			MMR: 157C, MMR_1: 157C,
[RW] 32-bits Access: 32			IND: 157C
Field Name	**Bits**	**Default**	**Description**
BRUSH_DATA63	31:0	0	<No Description>

BRUSH_Y_X [RW] 32-bits Access: 32			MMR: 1474, MMR_1: 1474, IND: 1474
Field Name	**Bits**	**Default**	**Description**
BRUSH_X	4:0	0	Brush X used for alignment purposes only.
(reserved)	7:5		
BRUSH_Y	12:8	0	Brush Y used for alignment purposes only.
(reserved)	15:13		
BRUSH_X_START	20:16	0	Initial value used for BRUSH_X pointer during Lines. When POLY_LINE is off, it is reloaded from BRUSH_X at the end of the line. When POLY_LINE is on, it is reloaded from the current Brush pointer at the end of the live. Whenever BRUSH_X is updated, the field should be written with the same value.
(reserved)	31:21		

BRUSH_SCALE [RW] 32-bits Access: 32			MMR: 1470, MMR_1: 1470, IND: 1470
Field Name	**Bits**	**Default**	**Description**
BRUSH_SCALE	7:0	0	Used to change the scale of a pattern when drawing 3D lines. This register applies to LINES only. It indicates the number of pixels to draw before incrementing the current BRUSH_X. This scale capability is required for OPEN GL patterned lines. If no scale is required, this register should be programmed to 1. This field only applies to 3D lines. Otherwise it is assumed to be 1.A value of 00 is interpreted as 256.
BRUSH_SCALE_START	15:8	0	Initial value used for BRUSH_SCALE counter. When POLY_LINE is off, it is reloaded from BRUSH_SCALE at the end of the line. When POLY_LINE is on, it is reloaded from the current Scale counter at the end of the line. Whenever BRUSH_SCALE is updated, the field should be with the same value. This field only applies to 3D lines. Otherwise it is assumed to be 1.
(reserved)	31:16		

7.5 Datapath Registers

DP_BRUSH_BKGD_CLR [RW] 32-bits Access: 32			MMR: 1478, MMR_1: 1478, IND: 1478
Field Name	**Bits**	**Default**	**Description**
DP_BRUSH_BKGD_CLR	31:0	0	Background color

DP_BRUSH_FRGD_CLR [RW] 32-bits Access: 32			MMR: 147C, MMR_1: 147C, IND: 147C
Field Name	**Bits**	**Default**	**Description**
DP_BRUSH_FRGD_CLR	31:0	0	Foreground color

DP_SRC_FRGD_CLR [RW] 32-bits Access: 32			MMR: 15D8, MMR_1: 15D8, IND: 15D8
Field Name	**Bits**	**Default**	**Description**
DP_SRC_FRGD_CLR	31:0	0	Foreground color. When color compare src eq flip is enabled, a '1' in bit location n means enable flipping on bit n.

DP_SRC_BKGD_CLR [RW] 32-bits Access: 32			MMR: 15DC, MMR_1: 15DC, IND: 15DC
Field Name	**Bits**	**Default**	**Description**
DP_SRC_BKGD_CLR	31:0	0	Background color

DP_CNTL [RW] 32-bits Access: 32			MMR: 16C0, MMR_1: 16C0, IND: 16C0
Field Name	**Bits**	**Default**	**Description**
DST_X_DIR	0	0	Destination X direction. This bit is written during setup engine initiated operations. This bit is set to '1' by a GUI_MASTER_CNTL write. 0 = right to left 1 = left to right
DST_Y_DIR	1	0	Destination Y direction. This bit is written during setup engine initiated operations. Note that this bit is assumed to be '1' for all triangles. This bit is set to '1' by a GUI_MASTER_CNTL write. 0 = bottom to top 1 = top to bottom
DST_Y_MAJOR	2	0	Destination Y major axis flag for bresenham lines. This bit is written during setup engine initiated operations. This bit is assumed to be '1' for all triangles. 0 = X major line 1 = Y Major line
DST_X_TILE	3	0	Enables rectangular tiling in the X direction. 0 = rectangular tiling in the X direction disabled 1 = rectangular tiling in the X direction enabled
DST_Y_TILE	4	0	Enables rectangular tiling in the Y direction. 0 = rectangular tiling in the X direction disabled 1 = rectangular tiling in the X direction enabled
DST_LAST_PEL	5	0	Destination last pel enable for lines. This bit is written during Setup engine operations. 0 = Destination last pel disabled 1 = Destination last pel enabled
TRAIL_X_DIR	6	0	Trapezoid trailing edge direction. This bit is written during setup engine initiated operations. 0 = right to left 1 = left to right
TRAIL_FILL_DIR	7	0	Trapezoid fill direction. 0 = right to left (trailing edge is to the left of the leading edge); 1 = left to right (trailing edge is to the right of the leading edge). This bit is written during setup engine initiated operations.

(Continued)

DP_CNTL [RW] 32-bits Access: 32			MMR: 16C0, MMR_1: 16C0, IND: 16C0
Field Name	**Bits**	**Default**	**Description**
BRES_SIGN	8	0	Bresenham sign. For Trapezoids with sub-pixel addressing, this bit is changed to include pixels on the top/left of the triangle. This bit is automatically set during setup engine operations. 0 = Zero error term is positive 1 = Zero error term is positive 1 (X Major lines and Y_DIR is 0 2. Y Major lines and X_DIR is 0)
(reserved)	14:9		
POLY_LINE	15	0	Indicates whether the current line is not the last line of a poly line. This bit implies BRUSH tiling. This bit is written during Setup engine operations. This bit is written to '1' by a DP_GUI_MASTER_CNTL write. 0 = Last or independent line 1 = Non-last line of polyline
DP_RASTER_STALL	16	0	If set, stall all DST operations until either: a) The Raster has passed the current destination location or b) No Display Offset writes are pending. 0 = Raster stall disabled 1 = Raster stall enabled
DP_TRI_DIS	17	0	If set, the edgewalker will accept a triangle from the setup engine, but only issue a single span, representing no pixels. Also denotes that DP_POLY_EDGE should not mask out Z writes. 0 = Draw triangles normally 1 = Draw no pixels for triangles
DP_POLY_EDGE	18	0	Denotes that the line to be drawn is an anti-aliased edge of a polygon. Sub-pixel adjust to first pixel center in the direction of the line, and always mask out Z writes. Always draw last pixel of the line. This bit only applies to 3D texture and shading operations. This bit is written by the Setup engine.
ANTI_ALIAS_INV_DMAJOR	22:19	0	Mantissa of the inverse of DMAJOR in normalized (1.xxxx) format. This field is written be the Setup engine.

(Continued)

DP_CNTL [RW] 32-bits Access: 32			MMR: 16C0, MMR_1: 16C0, IND: 16C0
Field Name	**Bits**	**Default**	**Description**
ANTI_ALIAS_SHIFT	27:23	0	Number of right shifts to do to the Bresenham Error term for anti-aliased lines to produce an error term between 0 and 15. This should be programmed with (1 - (exponent of the inverse of DMAJOR)). This field is written by the Setup engine.
ANTI_ALIAS_SLOPE	31:28	0	MSBs of absolute value of the slope. (DMINOR/DMAJOR) F represents 45 degrees. This field is written by the setup engine.

DP_DATATYPE [RW] 32-bits Access: 32			MMR: 16C4, MMR_1: 16C4, IND: 16C4
Field Name	**Bits**	**Default**	**Description**
DP_DST_DATATYPE	3:0	0	Destination datapath pixel width. Note: choices 7-15 only valid in 3D mode. 2 = 8 bpp pseudo-color 3 = 16 bpp aRGB 1555 4 = 16 bpp RGB 565 5 = 24 bpp RGB 6 = 32 aRGB 8888 7 = 8 bpp RGB 332 8 = Y8 greyscale 9 = RGB8 greyscale (8 bit intensity, duplicated for all 4 channels. Red channel is used on writes) 11 = YUV 422 packed (VYUY) 12 = YUV 422 packed(YVYU) 14 = aYUV 444(8:8:8:8) 15 = aRGB4444 (intermediate format only. Not understood by the Display Controller)
(reserved)	7:4		

(Continued)

DP_DATATYPE [RW] 32-bits Access: 32			MMR: 16C4, MMR_1: 16C4, IND: 16C4
Field Name	**Bits**	**Default**	**Description**
DP_BRUSH_DATATYPE	11:8	0	Brush datapath pixel type: 0 = 8X8 mono pattern (expanded to frgd, bkgd) 1 = 8X8 mono pattern (expanded to frgd, leave_alone) 2 = 8X1 mono pattern (expanded to frgd, leave_alone) 3 = 8X1 mono pattern (expanded to frgd, leave_alone) 4 = 1X8 mono pattern (expanded to frgd, bkgd) 5 = 1X8 mono pattern for line (expanded to frgd, leave_alone) 6 = 32X1 mono pattern for lines (expanded to frgd, bkgd) 7 = 32X1 mono pattern for line (expanded to frgd, leave_alone) 8 = 32X32 mono pattern for OPEN GL support (expanded to frgd, bkgd) 9 = 32X32 mono pattern for OPEN GL support (expanded to frgd, leave_alone) 10 = 8X8 color (pixel type same as DST) 11 = 8X1 color (pixel type same as DST) 12 = 1X8 color (pixel type same as DST) 13 = solid color (use frgd) 15 = Reserved for ProMo4 Parser. Must not be used by anyone else. Treat as 13, but really means no brush data is to be used
(reserved)	15:12		
DP_SRC_DATATYPE	17:16	0	Source datapath pixel type (If 3D/Scaler operations are in progress, this field is ignored and assumed to be 3). 0 = mono (expanded to frgd, bkgd) 1 = mono (expanded to frgd, leave_alone) 3 = color (pixel type same as DST)
(reserved)	28:18		
HOST_BIG_ENDIAN_EN	29	0	Enables big endian data translation for 15 bpp, 16 bpp, and 32 bpp pixel width. In 15 bpp and 16 bpp modes the bytes within each word are swapped. In 32 bpp mode the order of the four bytes within each dword is reversed. 0 = big endian data translation disabled 1 = big endian data translation enabled

(Continued)

DP_DATATYPE [RW] 32-bits Access: 32			MMR: 16C4, MMR_1: 16C4, IND: 16C4
Field Name	**Bits**	**Default**	**Description**
DP_BYTE_PIX_ORDER	30	0	Reverses the pixel order within each byte in monochrome modes: 0 = pixel order from MSBit to LSBit 1 = pixel order from LSBit to MSBit
DP_CONVERSION_TEMP	31	0	YUV to RGB conversion temperature: 0 = red@6500 K, GB@9300 K 1 = RGB@9300K

DP_CNTL_XDIR_YDIR_YMAJOR [RW] 32-bits Access: 32			MMR: 16D0, MMR_1: 16D0, IND: 16D0
Field Name	**Bits**	**Default**	**Description**
(reserved)	1:0		
DST_Y_MAJOR	2	0	Destination Y major axis flag for bresenham lines: 0 = X major line; 1 = Y major line. **Note:** Can we eliminate this bit and assume everything will be draw Y major? NO. Accuracy problem in polylines.
(reserved)	14:3		
DST_Y_DIR	15	0	Destination Y direction. 0 = bottom to top 1 = top to bottom
(reserved)	30:16		
DST_X_DIR	31	0	Destination X direction. 0 = right to left 1 = left to right

DP_MIX [RW] 32-bits Access: 32			MMR: 16C8, MMR_1: 16C8, IND: 16C8
Field Name	**Bits**	**Default**	**Description**
(reserved)	7:2		

DP_MIX [RW] 32-bits Access: 32			MMR: 16C8, MMR_1: 16C8, IND: 16C8
Field Name	**Bits**	**Default**	**Description**
(reserved)	7:0		
DP_SRC_SOURCE	10:8	0	SRC source. Note that during 3D/Scaler Operations (whenever SCALE_3D_FCN is non-zero) the DP_SRC_SOURCE field is ignored and data is always loaded from the 3D/Scaler pipeline 2 = loaded from memory (rectangular trajectory 3 = loaded through hostdata (linear trajectory) 4 = loaded through hostdata (linear trajectory & byte-aligned)
(reserved)	15:11		
DP_ROP3	23:16	0	Windows 3.1 ROP3 code 0 = ROP3 function
(reserved)	31:24	0	

DP_WRITE_MSK [RW] 32-bits Access: 32			MMR: 16CC, MMR_1: 16CC, IND: 16CC
Field Name	**Bits**	**Default**	**Description**
DP_WRITE_MSK	31:0		Write mask

DP_GUI_MASTER_CNTL [RW] 32-bits Access: 32			MMR: 146C, MMR_1: 146C, IND: 146C
Field Name	**Bits**	**Default**	**Description**
GMC_SRC_PITCH_OFFSET_CNTL	0	0	Control of SRC_OFFSET, SRC_PITCH: 0 = SRC_OFFSET=DEFAULT_OFFSET, SRC_PITCH=DEFAULT_PITCH 1 = Leave Alone
GMC_DST_PITCH_OFFSET_CNTL	1	0	Control of DST_OFFSET, DST_PITCH: 0 = DST_OFFSET=DEFAULT_OFFSET, DST_PITCH=DEFAULT_PITCH 1 = Leave Alone

(Continued)

DP_GUI_MASTER_CNTL [RW] 32-bits Access: 32			MMR: 146C, MMR_1: 146C, IND: 146C
Field Name	**Bits**	**Default**	**Description**
GMC_SRC_CLIPPING	2	0	Control of SRC scissors: 0 = (SRC_SC_RIGHT, SRC_SC_BOTTOM) = (DEFAULT_SC_BOTTOM_RIGHT) 1 = no default (leave alone)
GMC_DST_CLIPPING	3	0	Control of DST scissors: 0 = (SC_LEFT, SC_TOP) = (0,0), (SC_BOTTOM, SC_RIGHT) = DEF_SC_BOTTOM_RIGHT) 1 = no default (leave alone)
GMC_BRUSH_DATATYPE	7:4	0	Brush type to use: See DP_BRUSH_DATATYPE in DP_DATATYPE 0 = 8X8 mono pattern (expanded to frgd, bkgd) 1 = 8X8 mono pattern (expanded to frgd, leave_alone) 2 = 8X1 mono pattern (expanded to frgd, leave_alone) 3 = 8X1 mono pattern (expanded to frgd,leave_alone) 4 = 1X8 mono pattern (expanded to frgd, bkgd) 5 = 1X8 mono pattern for line (expanded to frgd,leave_alone) 6 = 32X1 mono pattern for lines (expanded to frgd, bkgd) 7 = 32X1 mono pattern for line (expanded to frgd,leave_alone) 8 = 32X32 mono pattern for OPEN GL support (expanded to frgd, bkgd) 9 = 32X32 mono pattern for OPEN GL support (expanded to frgd,leave_alone) 10 = 8X8 color (pixel type same as DST) 11 = 8X1 color (pixel type same as DST) 12 = 1X8 color (pixel type same as DST) 13 = solid color (use frgd) 15 = Reserved for ProMo4 Parser. Must not be used by anyone else. Treat as 13, but really means no brush data is to be used

(Continued)

DP_GUI_MASTER_CNTL [RW] 32-bits Access: 32			MMR: 146C, MMR_1: 146C, IND: 146C
Field Name	**Bits**	**Default**	**Description**
GMC_DST_DATATYPE	11:8	0	Dst type to use: See DP_DST_DATATYPE in DP_DATATYPE 2 = 8 bpp pseudo-color 3 = 16 bpp aRGB 1555 4 = 16 bpp RGB 565 5 = 24 bpp RGB 6 = 32 aRGB 8888 7 = 8 bpp RGB 332 8 = Y8 greyscale 9 = RGB8 greyscale (8 bit intensity, duplicated for all 4 channels. Red channel is used on writes) 11 = YUV 422 packed (VYUY) 12 = YUV 422 packed(YVYU) 14 = aYUV 444(8:8:8:8) 15 = aRGB4444 (intermediate format only. Not understood by the Display Controller)
GMC_SRC_DATATYPE	13:12	0	Src type to use:See DP_SRC_DATATYPE in DP_DATATYPE: 0 = mono (expanded to frgd, bkgd) 1 = mono (expanded to frgd, leave_alone) 2 = color (pixel type same as DST)=3
GMC_BYTE_PIX_ORDER	14	0	Mapped to DP_BYTE_PIX_ORDER in DP_DATATYPE: 0 = pixel order from MSBit to LSBit 1 = pixel order from LSBit to MSBit
GMC_CONVERSION_TEMP	15	0	Mapped to DP_CONVERSION_TEMP in DP_DATATYPE: 0 = red@6500 K, GB@9300 K 1 = RGB@9300K \
GMC_ROP3	23:16	0	Mapped to DP_ROP3 in DP_MIX 0 = ROP3 function
DP_SRC_SOURCE	26:24	0	Mapped to DP_SRC_SOURCE in DP_MIX: 2 = loaded from memory (rectangular trajectory 3 = loaded through hostdata (linear trajectory) 4 = loaded through hostdata (linear trajectory & byte-aligned)
GMC_3D_FCN_EN	27	0	0 = clear SCALE_3D_FCN, Z_EN, STENCIL_EN 1 = leave alone

(Continued)

DP_GUI_MASTER_CNTL [RW] 32-bits Access: 32			MMR: 146C, MMR_1: 146C, IND: 146C
Field Name	**Bits**	**Default**	**Description**
GMC_CLR_CMP_CNTL_DIS	28	0	0 = leave alone 1 = clear CLR_CMP_FCN_DST, CLR_CMP_FCN_SRC
GMC_AUX_CLIP_DIS	29	0	0 = leave alone 1 = clear all AUXn_SC_ENB bits
GMC_WR_MSK_DIS	30	0	0 = leave alone 1 = set DP_WRITE_MSK/CLR_CMP_MSK to ffffffff
GMC_LD_BRUSH_Y_X	31		0 = BRUSH_Y_X DWORD not contained in PROMO4 packet 1 = BRUSH_Y_X DWORD contained in PROMO4 packet

DP_GUI_MASTER_CNTL_C [RW] 32-bits Access: 32			MMR: 1C84, MMR_1: 1C84, IND: 1C84
Field Name	**Bits**	**Default**	**Description**
GMC_SRC_PITCH_OFFSET_CNTL	0	0	See same field in register DP_GUI_MASTER_CNTL: 0 = SRC_OFFSET=DEFAULT_OFFSET, SRC_PITCH=DEFAULT_PITCH 1 = Leave Alone
GMC_DST_PITCH_OFFSET_CNTL	1	0	See same field in register DP_GUI_MASTER_CNTL: 0 = DST_OFFSET=DEFAULT_OFFSET, DST_PITCH=DEFAULT_PITCH 1 = Leave Alone
GMC_SRC_CLIPPING	2	0	See same field in register DP_GUI_MASTER_CNTL: 0 = (SRC_SC_RIGHT, SRC_SC_BOTTOM) = (DEFAULT_SC_BOTTOM_RIGHT) 1 = no default (leave alone)

(Continued)

DP_GUI_MASTER_CNTL_C [RW] 32-bits Access: 32			MMR: 1C84, MMR_1: 1C84, IND: 1C84
Field Name	**Bits**	**Default**	**Description**
GMC_DST_CLIPPING	3	0	See same field in register DP_GUI_MASTER_CNTL: 0 = (SC_LEFT, SC_TOP) = (0,0), (SC_BOTTOM, SC_RIGHT) = DEF_SC_BOTTOM_RIGHT) 1 = no default (leave alone)
GMC_BRUSH_DATATYPE	7:4	0	See same field in register DP_GUI_MASTER_CNTL: 0 = 8X8 mono pattern (expanded to frgd, bkgd) 1 = 8X8 mono pattern (expanded to frgd, leave_alone) 2 = 8X1 mono pattern (expanded to frgd, leave_alone) 3 = 8X1 mono pattern (expanded to frgd,leave_alone) 4 = 1X8 mono pattern (expanded to frgd, bkgd) 5 = 1X8 mono pattern for line (expanded to frgd,leave_alone) 6 = 32X1 mono pattern for lines (expanded to frgd, bkgd) 7 = 32X1 mono pattern for line (expanded to frgd,leave_alone) 8 = 32X32 mono pattern for OPEN GL support (expanded to frgd, bkgd) 9 = 32X32 mono pattern for OPEN GL support (expanded to frgd,leave_alone) 10 = 8X8 color (pixel type same as DST) 11 = 8X1 color (pixel type same as DST) 12 = 1X8 color (pixel type same as DST) 13 = solid color (use frgd) 15 = Reserved for ProMo4 Parser. Must not be used by anyone else. Treat as 13, but really means no brush data is to be used

(Continued)

DP_GUI_MASTER_CNTL_C [RW] 32-bits Access: 32			MMR: 1C84, MMR_1: 1C84, IND: 1C84
Field Name	**Bits**	**Default**	**Description**
GMC_DST_DATATYPE	11:8	0	See same field in register DP_GUI_MASTER_CNTL: 2 = 8 bpp pseudo-color 3 = 16 bpp aRGB 1555 4 = 16 bpp RGB 565 5 = 24 bpp RGB 6 = 32 aRGB 8888 7 = 8 bpp RGB 332 8 = Y8 greyscale 9 = RGB8 greyscale (8 bit intensity, duplicated for all 4 channels. Red channel is used on writes) 11 = YUV 422 packed (VYUY) 12 = YUV 422 packed(YVYU) 14 = aYUV 444(8:8:8:8) 15 = aRGB4444 (intermediate format only. Not understood by the Display Controller)
GMC_SRC_DATATYPE	13:12	0	See same field in register DP_GUI_MASTER_CNTL: 0 = mono (expanded to frgd, bkgd) 1 = mono (expanded to frgd, leave_alone) 2 = color (pixel type same as DST)=3
GMC_BYTE_PIX_ORDER	14	0	See same field in register DP_GUI_MASTER_CNTL: 0 = pixel order from MSBit to LSBit 1 = pixel order from LSBit to MSBit
GMC_CONVERSION_TEMP	15	0	See same field in register DP_GUI_MASTER_CNTL: 0 = red@6500 K, GB@9300 K 1 = RGB@9300K \
GMC_ROP3	23:16	0	See same field in register DP_GUI_MASTER_CNTL: 0 = ROP3 function
DP_SRC_SOURCE	26:24	0	See same field in register DP_GUI_MASTER_CNTL: 2 = loaded from memory (rectangular trajectory 3 = loaded through hostdata (linear trajectory) 4 = loaded through hostdata (linear trajectory & byte-aligned)

(Continued)

DP_GUI_MASTER_CNTL_C [RW] 32-bits Access: 32			MMR: 1C84, MMR_1: 1C84, IND: 1C84
Field Name	**Bits**	**Default**	**Description**
GMC_3D_FCN_EN	27	0	See same field in register DP_GUI_MASTER_CNTL: 0 = clear SCALE_3D_FCN, Z_EN, STENCIL_EN 1 = leave alone
GMC_CLR_CMP_CNTL_DIS	28	0	<No Description>
GMC_AUX_CLIP_DIS	29	0	See same field in register DP_GUI_MASTER_CNTL
GMC_WR_MSK_DIS	30	0	See same field in register DP_GUI_MASTER_CNTL
(reserved)	31		

Desription:

Aliased to register DP_GUI_MASTER_CNTL.

DEFAULT_OFFSET [RW] 32-bits Access: 32			MMR: 16E0, MMR_1: 16E0, IND: 16E0
Field Name	**Bits**	**Default**	**Description**
DEFAULT_OFFSET	25:0	0	Default destination offset address for DP_GUI_MASTER_CNTL operations. Bits 3:0 of this field are hardwired to ZERO. See also description of DST_OFFSET.
(reserved)	31:26		

DEFAULT_PITCH [RW] 32-bits Access: 32			MMR: 16E4, MMR_1: 16E4, IND: 16E4
Field Name	**Bits**	**Default**	**Description**
DEFAULT_PITCH	9:0	0	Default destination pitch for DP_GUI_MASTER_CNTL operations. See also description of DST_PITCH.

DEFAULT_PITCH [RW] 32-bits Access: 32			MMR: 16E4, MMR_1: 16E4, IND: 16E4
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:10		

DEFAULT_SC_BOTTOM_RIGHT [RW] 32-bits Access: 32			MMR: 16E8, MMR_1: 16E8, IND: 16E8
Field Name	**Bits**	**Default**	**Description**
DEFAULT_SC_RIGHT	13:0	0	Default right scissor for DP_GUI_MASTER_CNTL
(reserved)	15:14		
DEFAULT_SC_BOTTOM	29:16	0	Default bottom scissor for DP_GUI_MASTER_CNTL
(reserved)	31:30		

7.6 Scissor Registers

The scissor registers define the rectangular region within which data is drawn. Left and right scissor registers are within the range -4096 to +4095. Top and bottom scissor registers are within the range -16384 to +16383. Polylines which follow a trajectory to the left of the left scissor register will result in a line drawn along the left scissor coordinate.

SC_LEFT [RW] 32-bits Access: 32			MMR: 1640, MMR_1: 1640, IND: 1640
Field Name	**Bits**	**Default**	**Description**
SC_LEFT	13:0	0	Destination left scissor: range -8192 to 8191
(reserved)	31:14		

SC_RIGHT [RW] 32-bits Access: 32			MMR: 1644, MMR_1: 1644, IND: 1644
Field Name	**Bits**	**Default**	**Description**
SC_RIGHT	13:0	0	Destination right scissor: range -8192 to 8191
(reserved)	31:14		

SC_TOP [RW] 32-bits Access: 32			MMR: 1648, MMR_1: 1648, IND: 1648
Field Name	**Bits**	**Default**	**Description**
SC_TOP	13:0	0	Destination top scissor: range -8192 to 8191
(reserved)	31:14		

SC_BOTTOM [RW] 32-bits Access: 32			MMR: 164C, MMR_1: 164C, IND: 164C
Field Name	**Bits**	**Default**	**Description**
SC_BOTTOM	13:0	0	Destination bottom scissor: range -8192 to 8191
(reserved)	31:14		

AUX_SC_CNTL [RW] 32-bits Access: 32			MMR: 1660, MMR_1: 1660, IND: 1660
Field Name	**Bits**	**Default**	**Description**
AUX1_SC_ENB	0	0	Enable for Auxiliary 1 scissors: 0 = Off 1 = On. This bit is set to 0 on Chip Reset.
AUX1_SC_MODE	1	0	Auxiliary Scissors can function in 1 of 2 modes: 0 = Additive. Combine with other destination SCISSORs with 'OR'. 1 = Subtractive. Combine with other destination SCISSORs with 'AND NOT'.
AUX2_SC_ENB	2	0	Enable for Auxiliary 2 scissors: 0 = Off 1 = On. This bit is set to 0 on Chip Reset
AUX2_SC_MODE	3	0	Auxiliary Scissors can function in 1 of 2 modes: 0 = Additive. Combine with other destination SCISSORs with 'OR'. 1 = Subtractive. Combine with other destination SCISSORs with 'AND NOT'.
AUX3_SC_ENB	4	0	Enable for Auxiliary 3 scissors: 0 = Off 1 = On. This bit is set to 0 on Chip Reset
AUX3_SC_MODE	5	0	Auxiliary Scissors can function in 1 of 2 modes: 0 = Additive. Combine with other destination SCISSORs with 'OR'. 1 = Subtractive. Combine with other destination SCISSORs with 'AND NOT'.
(reserved)	31:6		

AUX1_SC_LEFT [RW] 32-bits Access: 32			MMR: 1664, MMR_1: 1664, IND: 1664
Field Name	**Bits**	**Default**	**Description**
AUX1_SC_LEFT	13:0	0	Auxiliary 1 left scissor: range -8192 to 8191.
(reserved)	31:14		

AUX1_SC_RIGHT [RW] 32-bits Access: 32			MMR: 1668, MMR_1: 1668, IND: 1668
Field Name	**Bits**	**Default**	**Description**
AUX1_SC_RIGHT	13:0	0	Auxiliary 1 right scissor: range -8192 to 8191.
(reserved)	31:14		

AUX1_SC_TOP [RW] 32-bits Access: 32			MMR: 166C, MMR_1: 166C, IND: 166C
Field Name	**Bits**	**Default**	**Description**
AUX1_SC_TOP	13:0	0	Auxiliary 1 top scissor: range -8192 to 8191.
(reserved)	31:14		

AUX1_SC_BOTTOM [RW] 32-bits Access: 32			MMR: 1670, MMR_1: 1670, IND: 1670
Field Name	**Bits**	**Default**	**Description**
AUX1_SC_BOTTOM	13:0	0	Auxiliary 2 bottom scissor: range -8192 to 8191.
(reserved)	31:14		

AUX2_SC_LEFT [RW] 32-bits Access: 32			MMR: 1674, MMR_1: 1674, IND: 1674
Field Name	**Bits**	**Default**	**Description**
AUX2_SC_LEFT	13:0	0	Auxiliary 2 left scissor: range -8192 to 8191.
(reserved)	31:14		

AUX2_SC_RIGHT [RW] 32-bits Access: 32			MMR: 1678, MMR_1: 1678, IND: 1678
Field Name	**Bits**	**Default**	**Description**
AUX2_SC_RIGHT	13:0	0	Auxiliary 2 right scissor: range -8192 to 8191.
(reserved)	31:14		

AUX2_SC_TOP [RW] 32-bits Access: 32			MMR: 167C, MMR_1: 167C, IND: 167C
Field Name	**Bits**	**Default**	**Description**
AUX2_SC_TOP	13:0	0	Auxiliary 2 top scissor: range -8192 to 8191.
(reserved)	31:14		

AUX2_SC_BOTTOM [RW] 32-bits Access: 32			MMR: 1680, MMR_1: 1680, IND: 1680
Field Name	**Bits**	**Default**	**Description**
AUX2_SC_BOTTOM	13:0	0	Auxiliary 2 bottom scissor: range -8192 to 8191.
(reserved)	31:14		

AUX3_SC_LEFT [RW] 32-bits Access: 32			MMR: 1684, MMR_1: 1684, IND: 1684
Field Name	**Bits**	**Default**	**Description**
AUX3_SC_LEFT	13:0	0	Auxiliary 3 left scissor: range -8192 to 8191.
(reserved)	31:14		

AUX3_SC_RIGHT [RW] 32-bits Access: 32			MMR: 1688, MMR_1: 1688, IND: 1688
Field Name	**Bits**	**Default**	**Description**
AUX3_SC_RIGHT	13:0	0	Auxiliary 3 right scissor: range -8192 to 8191.
(reserved)	31:14		

AUX3_SC_TOP [RW] 32-bits Access: 32			MMR: 168C, MMR_1: 168C, IND: 168C
Field Name	**Bits**	**Default**	**Description**
AUX3_SC_TOP	13:0	0	Auxiliary 3 top scissor: range -8192 to 8191.
(reserved)	31:14		

AUX3_SC_BOTTOM [RW] 32-bits Access: 32			MMR: 1690, MMR_1: 1690, IND: 1690
Field Name	**Bits**	**Default**	**Description**
AUX3_SC_BOTTOM	13:0	0	Auxiliary 3 bottom scissor: range -8192 to 8191.
(reserved)	31:14		

SC_TOP_LEFT [W] 32-bits Access: 32			MMR: 16EC, MMR_1: 16EC, IND: 16EC
Field Name	**Bits**	**Default**	**Description**
SC_LEFT	13:0	0	Left scissor
(reserved)	15:14		
SC_TOP	29:16	0	Top scissor
(reserved)	31:30		

Desription:

Destination SC_TOP_LEFT.

SC_BOTTOM_RIGHT [W] 32-bits Access: 32			MMR: 16F0, MMR_1: 16F0, IND: 16F0
Field Name	**Bits**	**Default**	**Description**
SC_RIGHT	13:0	0	Right scissor
(reserved)	15:14		
SC_BOTTOM	29:16	0	Bottom scissor
(reserved)	31:30		

Desription:

Destination BOTTOM_RIGHT.

SC_TOP_LEFT_C [W] 32-bits Access: 32			MMR: 1C88, MMR_1: 1C88, IND: 1C88
Field Name	**Bits**	**Default**	**Description**
SC_LEFT	13:0	0	Right scissor
(reserved)	15:14		
SC_TOP	29:16	0	Bottom scissor
(reserved)	31:30		

SC_BOTTOM_RIGHT_C [W] 32-bits Access: 32			MMR: 1C8C, MMR_1: 1C8C, IND: 1C8C
Field Name	**Bits**	**Default**	**Description**
SC_RIGHT	13:0	0	Right scissor
(reserved)	15:14		
SC_BOTTOM	29:16	0	Bottom scissor
(reserved)	31:30		

7.7 Color Compare Registers

CLR_CMP_CLR_SRC [RW] 32-bits Access: 32			MMR: 15C4, MMR_1: 15C4, IND: 1590
Field Name	**Bits**	**Default**	**Description**
CLR_CMP_CLR_SRC	31:0	0	Color comparison color of source

CLR_CMP_CLR_DST [RW] 32-bits Access: 32			MMR: 15C8, MMR_1: 15C8, IND: 15C8
Field Name	**Bits**	**Default**	**Description**
CLR_CMP_CLR_DSP	31:0	0	Color comparison color of destination

CLR_CMP_CNTL [RW] 32-bits Access: 32			MMR: 15C0, MMR_1: 15C0, IND: 15C0
Field Name	**Bits**	**Default**	**Description**
CLR_CMP_FN_SRC	2:0	0	Color comparison function (Mnemonic, action): 0 = False (CMP_FALSE, always draw) 1 = True (CMP_TRUE, never draw) 2-3 = (reserved) 4 = SRC_CLR!= CLR_CMP_CLR_SRC (CMP_EQ_COLOR, draw when eq) 5 = SRC_CLR = CLR_CMP_CLR_SRC (CMP_NEQ_COLOR, draw when neq) 6 = (reserved) 7 = SRC_CLR = CLR_CMP_CLR_SRC (CMP_EQ_FLIP, flip using expanded SRC_FRGD_CLR as flip mask when eq)
(reserved)	7:3		
CLR_CMP_FN_DST	10:8	0	Color comparison function (Mnemonic, action): 0 = False (CMP_FALSE, always draw) 1 = True (CMP_TRUE, never draw) 2-3 = (reserved) 4 = DST_CLR!= CLR_CMP_CLR_DST (CMP_EQ_COLOR, draw when eq) 5 = DST_CLR = CLR_CMP_CLR_DST (CMP_NEQ_COLOR, draw when neq) 6-7 = (reserved)

CLR_CMP_CNTL [RW] 32-bits Access: 32			MMR: 15C0, MMR_1: 15C0, IND: 15C0
Field Name	**Bits**	**Default**	**Description**
(reserved)	23:11		
CLR_CMP_SRC	25:24	0	Defines source for color keying: 0 = Destination 1 = Source 2 = Src and Dst 3 = (reserved)
(reserved)	31:26		

CLR_CMP_MSK [RW] 32-bits Access: 32			MMR: 15CC, MMR_1: 15CC, IND: 15CC
Field Name	**Bits**	**Default**	**Description**
CLR_CMP_MSK	31:0	0	Color comparison color mask

CLR_CMP_CLR_3D [RW] 32-bits Access: 32			MMR: 1A24, MMR_1: 1A24, IND: 1A24
Field Name	**Bits**	**Default**	**Description**
CLR_CMP_CLR_3D	23:0	0	Color comparison color
(reserved)	31:24		

CLR_CMP_MSK_3D [RW] 32-bits Access: 32			MMR: 1A28, MMR_1: 1A28, IND: 1A28
Field Name	**Bits**	**Default**	**Description**
CLR_CMP_MSK_3D	23:0	0	Color compare mask
(reserved)	31:24		

7.8 2D Engine Control Registers

WAIT_UNTIL [RW] 32-bits Access: 32			MMR: 1720, MMR_1: 1720, IND: 1720
Field Name	**Bits**	**Default**	**Description**
EVENT_CRTC_OFFSET	0	0	Used to stall until the display has started displaying the last new CRTC_OFFSET value written. (i.e. used to do page flips.) Write 0: No effect. Write 1: Stall CMDFIFO until CRTC_GUI_TRIG_OFFSET = 0. See also CRTC_OFFSET register.
EVENT_RE_CRTC_VLINE	1	0	Used to stall until the display has reached the start of a specific range of raster lines. Write 0: No effect. Write 1: Stall CMDFIFO until CRTC_GUI_TRIG_VLINE has a rising edge. See also CRTC_GUI_TRIG_VLINE register.
EVENT_FE_CRTC_VLINE	2	0	Used to stall until the display has reached the end of a specific range of raster lines. Write 0: No effect. Write 1: Stall CMDFIFO until CRTC_GUI_TRIG_VLINE has a falling edge. See CRTC_GUI_TRIG_VLINE register.
EVENT_CRTC_VLINE	3	0	Used to stall until the display has reached anywhere in a specific range of raster lines. Write 0: No effect. Write 1: Stall CMDFIFO until CRTC_GUI_TRIG_VLINE = 1. See also CRTC_GUI_TRIG_VLINE register.
EVENT_BM_VIP0_IDLE	4	0	Write 0: No effect Write 1: Stall CMDFIFO until BM_IDLE for this channel.
EVENT_BM_VIP1_IDLE	5	0	Write 0: No effect Write 1: Stall CMDFIFO until BM_IDLE for this channel.
EVENT_BM_VIP2_IDLE	6	0	Write 0: No effect Write 1: Stall CMDFIFO until BM_IDLE for this channel.
EVENT_BM_VIP3_IDLE	7	0	Write 0: No effect Write 1: Stall CMDFIFO until BM_IDLE for this channel.

(Continued)

WAIT_UNTIL [RW] 32-bits Access: 32			MMR: 1720, MMR_1: 1720, IND: 1720
Field Name	**Bits**	**Default**	**Description**
EVENT_BM_VIDCAP_IDLE	8	0	Write 0: No effect Write 1: Stall CMDFIFO until BM_IDLE for this channel.
EVENT_BM_GUI_IDLE	9	0	Write 0: No effect Write 1: Stall CMDFIFO until BM_IDLE for this channel.
EVENT_CMDFIFO	10	0	Write 0: No effect Write 1: Stall CMDFIFO until number of entries specified in EVENT_CMDFIFO_ENTRIES is met.
EVENT_OV0_FLIP	11	0	Write 0: No effect. Write 1: Stall CMDFIFO until OV0_FLIP='1'. The intent here is for the overlay to be able to tell the GUI that it is using the surface that the GUI wants to render to. The overlay will send an 'OV0_FLIP' signal to the GUI. It will make this signal go low when there is a danger of front buffer overwrite as determined by software. If software wants to stall the GUI, then it will set OV0_STALL_GUI_UNTIL_FLIP when it locks, updates, and unlocks overlay and subpicture registers. OV0_FLIP will go low at unlock and then high during VBlank (when the hardware double buffering flips the registers). The behavior of OV0_FLIP is undefined if OV0_STALL_GUI_UNTIL_FLIP is written to when the lock bit is not set. OV0_FLIP is not an event signal. If it is low the WaitUntilEvent command must stall the GUI until it is high. It does not wait until the signal transitions from low to high. (i.e. If it is already high, there is no stall).
(reserved)	19:12		
EVENT_CMDFIFO_ENTRIES	26:20	0	Number of CMDFIFO entries to trigger on.
(reserved)	31:27		

Description:

Enables stalling the processing of commands out of the command FIFO until the selected trigger condition is reached. This is done to delay the processing of further contents of the Promo4 stream until certain engines in the chip have reached certain milestones. Stall CMDFIFO based on 'AND' of all set triggers.

7.9 2D Engine Status Registers

GIU_SCRATCH_REG0			MMR: 15E0, MMR_1: 15E0,
[RW] 32-bits Access: 32			IND: 15E0
Field Name	**Bits**	**Default**	**Description**
GUI_SCRATCH_REG0	31:0	0	FIFO'd scratch register.

GIU_SCRATCH_REG1			MMR: 15E4, MMR_1: 15E4,
[RW] 32-bits Access: 32			IND: 15E0
Field Name	**Bits**	**Default**	**Description**
GUI_SCRATCH_REG1	31:0	0	FIFO'd scratch register.

GIU_SCRATCH_REG2			MMR: 15E8, MMR_1: 15E8,
[RW] 32-bits Access: 32			IND: 15E8
Field Name	**Bits**	**Default**	**Description**
GUI_SCRATCH_REG2	31:0	0	FIFO'd scratch register.

GIU_SCRATCH_REG3			MMR: 15EC, MMR_1: 15EC,
[RW] 32-bits Access: 32			IND: 15EC
Field Name	**Bits**	**Default**	**Description**
GUI_SCRATCH_REG3	31:0	0	FIFO'd scratch register.

GIU_SCRATCH_REG4 [RW] 32-bits Access: 32			MMR: 15F0, MMR_1: 15F0, IND: 15F0
Field Name	**Bits**	**Default**	**Description**
GUI_SCRATCH_REG4	31:0	0	FIFO'd scratch register.

GIU_SCRATCH_REG5 [RW] 32-bits Access: 32			MMR: 15F4, MMR_1: 15F4, IND: 15F4
Field Name	**Bits**	**Default**	**Description**
GUI_SCRATCH_REG5	31:0	0	FIFO'd scratch register.

GUI_STAT [RW] 32-bits Access: 32			MMR: 1740, MMR_1: 1740, IND: 1740
Field Name	**Bits**	**Default**	**Description**
GUI_FIFOCNT	11:0	40	Number of free CMDFIFO entries
(reserved)	15:12		
PM4_BUSY	16	0	State of PROMO_4 engine
MICRO_BUSY	17	0	State of the micro engine
FPU_BUSY	18	0	State of the pre-setup engine
VC_BUSY	19	0	State of the Vertex controller engine
IDCT_BUSY	20	0	State of the IDCT engine
ENG_EV_BUSY	21	0	State of the event engine
SETUP_BUSY	22	0	State of the setup engine
EDGEWALK_BUSY	23	0	State of the edgewalker pipeline
ADDRESSING_BUSY	24	0	State of the texel/Destination addressing pipeline
ENG_3D_BUSY	25	0	State of the eng_3d data pipeline
ENG_2D_SM_BUSY	26	0	State of the eng_2d engine
ENG_2D_BUSY	27	0	State of the eng_2d pipeline
GUI_WB_BUSY	28	0	State of the gui write buffer

GUI_STAT [RW] 32-bits Access: 32			MMR: 1740, MMR_1: 1740, IND: 1740
Field Name	**Bits**	**Default**	**Description**
CACHE_BUSY	29	0	State of the pixel cache
(reserved)	30		
GUI_ACTIVE	31	0	'OR' of the above bits

GUI_DEBUG0 [RW] 32-bits Access: 32			MMR: 16A0, MMR_1: 16A0, IND: 16A0
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

Description:

These are the 2D engine debug bits. These bits can only be written when the GUI is idle and are written through the command FIFO.

GUI_DEBUG1 [RW] 32-bits Access: 32			MMR: 16A4, MMR_1: 16A4, IND: 16A4
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

GUI_DEBUG2 [RW] 32-bits Access: 32			MMR: 16A8, MMR_1: 16A8, IND: 16A8
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

GUI_DEBUG3 [RW] 32-bits Access: 32			MMR: 16AC, MMR_1: 16AC, IND: 16AC
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

GUI_DEBUG4 [RW] 32-bits Access: 32			MMR: 16B0, MMR_1: 16B0, IND: 16B0
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

GUI_DEBUG5 [RW] 32-bits Access: 32			MMR: 16B4, MMR_1: 16B4, IND: 16B4
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

GUI_DEBUG6 [RW] 32-bits Access: 32			MMR: 16B8, MMR_1: 16B8, IND: 16B8
Field Name	**Bits**	**Default**	**Description**
(reserved)	31:0		

GUI_PROBE [RW] 32-bits Access: 32			MMR: 16BC, MMR_1: 16BC, IND: 16BC
Field Name	**Bits**	**Default**	**Description**
GUI_STATE	2:0	0	Eng_2d state machine: 000 = idle 001 = DO_COLOR_WRITE 010 = DO_SPAN_EVEN 011 = DO_SPAN 100-101 = (undefined) 110 = WAIT_FOR_SPAN 111 = WAIT_FOR_PIPE_EMPTY.
GUI_PROBE_DUMMY3	3	0	<No Description>
GUI_PROBE_DUMMY4	4	0	<No Description>
GUI_PROBE_DUMMY5	5	0	<No Description>
GUI_PROBE_DUMMY6	6	0	<No Description>
GUI_PROBE_DUMMY7	7	0	<No Description>
GUI_SPAN_REQ	8	0	Span FIFO request
GUI_SPAN_RDY	9	0	Span FIFO ready
GUI_REQ_SRCS	10	0	All required sources present (stage 1 write)
GUI_HOST_REQ	11	0	HOST_DATA request
GUI_HOST_RDY	12	0	HOST_DATA ready
GUI_SRC_REQ	13	0	SRC/Z request
GUI_SRC_RDY	14	0	SRC/Z ready
GUI_E3D_REQ	15	0	3D data request
GUI_E3D_RDY	16	0	3D data ready
GUI_DST_REQ	17	0	DST request
GUI_DST_RDY	18	0	DST ready
GUI_WRT_REQ	19	0	DST write request
GUI_WRT_ZS_REQ	20	0	Z write request
GUI_WRT_RDY	21	0	Write ready
GUI_PROBE_DUMMY22	22	0	<No Description>
GUI_PROBE_DUMMY23	23	0	<No Description>

(Continued)

GUI_PROBE [RW] 32-bits Access: 32			MMR: 16BC, MMR_1: 16BC, IND: 16BC
Field Name	**Bits**	**Default**	**Description**
GUI_PROBE_DUMMY24	24	0	<No Description>
GUI_PROBE_DUMMY25	25	0	<No Description>
GUI_PROBE_DUMMY26	26	0	<No Description>
GUI_PROBE_DUMMY27	27	0	<No Description>
GUI_PROBE_DUMMY28	28	0	<No Description>
GUI_PROBE_DUMMY29	29	0	<No Description>
GUI_PROBE_DUMMY30	30	0	<No Description>
GUI_PROBE_DUMMY31	31	0	<No Description>

Description:

Probe of internal 2D draw engine signals.

FLUSH_1 [RW] 32-bits Access: 32			MMR: 1704, MMR_1: 1704, IND: 1704
Field Name	**Bits**	**Default**	**Description**
FLUSH_1	31:0	0	Block FIFO'd writes until level 1 engines are idle.

FLUSH_2 [RW] 32-bits Access: 32			MMR: 1708, MMR_1: 1708, IND: 1708
Field Name	**Bits**	**Default**	**Description**
FLUSH_2	31:0	0	Block FIFO'd writes until level 2 engines are idle.

FLUSH_3 [RW] 32-bits Access: 32			MMR: 170C, MMR_1: 170C, IND: 170C
Field Name	**Bits**	**Default**	**Description**
FLUSH_3	31:0	0	Block FIFO'd writes until level 3 engines are idle.

FLUSH_4 [RW] 32-bits Access: 32			MMR: 1710, MMR_1: 1710, IND: 1710
Field Name	**Bits**	**Default**	**Description**
FLUSH_4	31:0	0	Block FIFO'd writes until level 4 engines are idle.

FLUSH_5 [RW] 32-bits Access: 32			MMR: 1714, MMR_1: 1714, IND: 1714
Field Name	**Bits**	**Default**	**Description**
FLUSH_5	31:0	0	Block FIFO'd writes until level 5 engines are idle.

FLUSH_6 [RW] 32-bits Access: 32			MMR: 1718, MMR_1: 1718, IND: 1718
Field Name	**Bits**	**Default**	**Description**
FLUSH_6	31:0	0	Block FIFO'd writes until level 6 engines are idle.

FLUSH_7 [RW] 32-bits Access: 32			MMR: 171C, MMR_1: 171C, IND: 171C
Field Name	**Bits**	**Default**	**Description**
FLUSH_7	31:0	0	Block FIFO'd writes until level 7 engines are idle.

This page intentionally left blank.

Chapter 8
Miscellaneous Registers

8.1 Miscellaneous Registers

Miscellaneous RegistFunction: Overlay RegistersActive Field: OV0_SCALE_Y2R_DIS: Divide by 10x0: Divide by 1.

GEN_INT_CNTL [RW] 32-bits Access: 32			MMR: 40, MMR_1: 40, IOR: 40, IND: 40
Field Name	**Bits**	**Default**	**Description**
CRTC_VBLANK_INT_EN	0	0	Vertical blank interrupt enable. 0 = Disable 1 = Enable
CRTC_VLINE_INT_EN	1	0	Vertical line interrupt enable. 0 = Disable 1 = Enable
CRTC_VSYNC_INT_EN	2	0	Vertical sync interrupt enable. 0 = Disable 1 = Enable
SNAPSHOT_INT_EN	3	0	Snapshot interrupt enable. 0 = Disable 1 = Enable
(reserved)	15:4		
BUSMASTER_EOL_INT_EN	16	0	Bus master end-of-system-list interrupt enable. 0 = Disable 1 = Enable
I2C_INT_EN	17	0	I^2C interrupt enable. 0 = Disable 1 = Enable
MPP_GP_INT_EN	18	0	<No Description> 0 = Disable 1 = Enable
GUI_IDLE_INT_EN	19	0	<No Description> 0 = Disable 1 = Enable
(reserved)	23:20		

(Continued)

GEN_INT_CNTL [RW] 32-bits Access: 32			MMR: 40, MMR_1: 40, IOR: 40, IND: 40
Field Name	**Bits**	**Default**	**Description**
VIPH_INT_EN	24	0	<No Description> 0 = Disable 1 = Enable
(reserved)	31:25		

Description:

Interrupt enables. Setting bits allows corresponding status bit to generate an interrupt signal to the system. No effect if strapped to interrupt disable.

GEN_INT_STATUS [RW] 32-bits Access: 8/16/32			MMR: 44, MMR_1: 44, IOR: 44, IND: 44
Field Name	**Bits**	**Default**	**Description**
CRTC_VBLANK_INT **(R)**	0	0	Vertical blank started since last cleared. 0 = No event 1 = Event has occurred, interrupting if enabled
CRTC_VBLANK_INT_AK **(W)**	0	0	Write '1' clears CRTC_VBLANK_INT status. 0 = No effect 1 = Clear status
CRTC_VLINE_INT_AK **(W)**	1	0	Write '1' clears CRTC_VLINE_INT status. 0 = No effect 1 = Clear status
CRTC_VLINE_INT **(R)**	1	0	Vertical line trigger point reached since last cleared. 0 = No event 1 = Event has occurred, interrupting if enabled
CRTC_VSYNC_INT_AK **(W)**	2	0	Write '1' clears CRTC_VSYNC_INT status. 0 = No effect 1 = Clear status
CRTC_VSYNC_INT **(R)**	2	0	Vertical sync started since last cleared. 0 = No event 1 = Event has occurred, interrupting if enabled

(Continued)

GEN_INT_STATUS **[RW] 32-bits Access: 8/16/32**			**MMR: 44, MMR_1: 44,** **IOR: 44, IND: 44**
Field Name	**Bits**	**Default**	**Description**
SNAPSHOT_INT_AK **(W)**	3	0	Write '1' clears SNAPSHOT_INT status. 0 = No effect 1 = Clear status
SNAPSHOT_INT **(R)**	3	0	Snapshot taken since last cleared. 0 = No event 1 = Event has occurred, interrupting if enabled
(reserved)	7:4		
CAP0_INT_ACTIVE **(R)**	8	0	Indicates capture port 0 is generating an interrupt. 0 = Capture port 0 not source of any active interrupt 1 = Capture port 0 has active interrupt(s)
CAP1_INT_ACTIVE **(R)**	9	0	Indicates capture port 1 is generating an interrupt. 0 = Capture port 1 not source of any active interrupt 1 = Capture port 1 has active interrupt(s)
(reserved)	15:10		
BUSMASTER_EOL_INT_AK **(W)**	16	0	<No Description> 0 = No effect 1 = Clear status
BUSMASTER_EOL_INT **(R)**	16	0	<No Description> 0 = No event 1 = Event has occurred, interrupting if enabled
I2C_INT_AK **(W)**	17	0	<No Description> 0 = No effect 1 = Clear status
I2C_INT **(R)**	17	0	<No Description> 0 = No event 1 = Event has occurred, interrupting if enabled
MPP_GP_INT_AK **(W)**	18	0	<No Description> 0 = No effect 1 = Clear status
MPP_GP_INT **(R)**	18	0	<No Description> 0 = No event 1 = Event has occurred, interrupting if enabled
GUI_IDLE_INT_AK **(W)**	19	0	<No Description> 0 = No effect 1 = Clear status

(Continued)

GEN_INT_STATUS [RW] 32-bits Access: 8/16/32			MMR: 44, MMR_1: 44, IOR: 44, IND: 44
Field Name	**Bits**	**Default**	**Description**
GUI_IDLE_INT **(R)**	19	0	<No Description> 0 = No event 1 = Event has occurred, interrupting if enabled
(reserved)	23:20		
VIPH_INT_AK **(W)**	24	0	<No Description> 0 = No effect 1 = Clear status
VIPH_INT **(R)**	24	0	<No Description> 0 = No event 1 = Event has occurred, interrupting if enabled
(reserved)	31:25		

Description:

Interrupt & Status indicators. Read shows current states. Write of 1 clears states. Note each field may be used for polling, even if not enabled to generate an interrupt.

GEN_RESET_CNTL [RW] 32-bits Access: 8/16/32			MMR: F0, MMR_1: F0, IOR: F0, IND: F0
Field Name	**Bits**	**Default**	**Description**
SOFT_RESET_GUI	0	0	<No Description> 0 = Not reset 1 = Reset
(reserved)	7:1		
SOFT_RESET_VCLK	8	0	<No Description> 0 = Not reset 1 = Reset
SOFT_RESET_PCLK	9	0	<No Description> 0 = Not reset 1 = Reset

(Continued)

GEN_RESET_CNTL [RW] 32-bits Access: 8/16/32			MMR: F0, MMR_1: F0, IOR: F0, IND: F0
Field Name	**Bits**	**Default**	**Description**
SOFT_RESET_ECP	10	0	<No Description> 0 = Not reset 1 = Reset
SOFT_RESET_DISPENG_XCLK	11	0	<No Description> 0 = Not reset 1 = Reset
SOFT_RESET_MEMCTLR_XCLK	12	0	<No Description> 0 = Not reset 1 = Reset
(reserved)	31:13		

Description:

Soft reset controls for various blocks.

SW_SEMAPHORE [RW] 32-bits Access: 8/16/32			MMR: 13C, MMR_1: 13C, IND: 13C
Field Name	**Bits**	**Default**	**Description**
SW_SEMAPHORE	15:0	0	Scratch register for use by software to implement status flags and semaphores. No affect on the hardware.
(reserved)	31:16		

Description:

Scratch register.

AMCGPIO_MASK [RW] 32-bits Access: 8/16/32			MMR: 194, MMR_1: 194, IND: 194
Field Name	**Bits**	**Default**	**Description**
AMCGPIO_MASK	31:0	0	Each bit in this register makes the same bit in the AMCGPIO_A_REG and AMCGPIO_EN_REG effective.

MDGPIO_MASK [RW] 32-bits Access: 8/16/32			MMR: 198, MMR_1: 198, IND: 198
Field Name	**Bits**	**Default**	**Description**
MDGPIO_MASK	31:0	0	Each bit in this register makes the same bit in the MDGPIO_A_REG and MDGPIO_EN_REG effective.

AMCGPIO_A_REG [RW] 32-bits Access: 8/16/32			MMR: 1A0, MMR_1: 1A0, IND: 1A0
Field Name	**Bits**	**Default**	**Description**
AMCGPIO_A	31:0	0	This register controls the 'a' pin of 26 pads, but each bit inside this register is effective only if the same bit inside the AMCGPIO_MASK is turned on. The register bits are mapped to pins as follows: AMCGPIO_A(3:0) - Address/Data for MPP AMCGPIO_A(7:4) - Address/Data for MPP or VIP HAD(7:4) AMCGPIO_A(8) - MPP AS or VIPCLK AMCGPIO_A(9) - MPP DS or VIP HCTL AMCGPIO_A(10) - MPP SRDY or VIP HAD(0) AMCGPIO_A(11) - BUS_CLK_SEL_STRAP / Clock for EPROM flops / VIP HAD(1) AMCGPIO_A(19:12) - DVS data in AMCGPIO_A(20) - DVS clock in AMCGPIO_A(21) - BYTCLK AMCGPIO_A(22) - I2C SDA / VIP HAD(2) AMCGPIO_A(23) - I2C SCL / VIP HAD(3) / VIP interrupt AMCGPIO_A(24) - LCDCLK AMCGPIO_A(25) - LCDCDE

AMCGPIO_Y_REG [RW] 32-bits Access: 8/16/32			MMR: 1A4, MMR_1: 1A4, IND: 1A4
Field Name	**Bits**	**Default**	**Description**
AMCGPIO_Y **(R)**	31:0	0	Reading from this register gives the logic value on the 'p' pin of the corresponding pad.

AMCGPIO_EN_REG [RW] 32-bits Access: 8/16/32			MMR: 1A8, MMR_1: 1A8, IND: 1A8
Field Name	**Bits**	**Default**	**Description**
AMCGPIO_EN	31:0	0	This register controls the output enable of 26 pads, but each bit inside this register is effective only if the same bit inside the AMCGPIO_MASK is turned on. Turning on the enable will make that pad an output from the chip, turning it off makes that pad input. The register bits are mapped to pins as follows: AMCGPIO_A(3:0) - Address/Data for MPP AMCGPIO_A(7:4) - Address/Data for MPP or VIP HAD(7:4) AMCGPIO_A(8) - MPP AS or VIPCLK AMCGPIO_A(9) - MPP DS or VIP HCTL AMCGPIO_A(10) - MPP SRDY or VIP HAD(0) AMCGPIO_A(11) - BUS_CLK_SEL_STRAP / Clock for EPROM flops / VIP HAD(1) AMCGPIO_A(19:12) - DVS data in AMCGPIO_A(20) - DVS clock in AMCGPIO_A(21) - BYTCLK AMCGPIO_A(22) - I2C SDA / VIP HAD(2) AMCGPIO_A(23) - I2C SCL / VIP HAD(3) / VIP interrupt AMCGPIO_A(24) - LCDCLK AMCGPIO_A(25) -LCDCDE

MDGPIO_A_REG			**MMR: 1AC, MMR_1: 1AC,**
[RW] 32-bits Access: 8/16/32			**IND: 1AC**
Field Name	**Bits**	**Default**	**Description**
MDGPIO_A	31:0	0	This register controls the 'a' pin of the DQ pads, but each bit inside this register is effective only if the same bit inside the MDGPIO_MASK is turned on. These register bits are mapped to pins as follows: MDGPIO_A(7:6) - DQ(71:70) ZV Control Port MDGPIO_A(15:8) - DQ(79:72) Extended VIP / DVS port / ZV data in MDGPIO_A(22) - DQ(86) DS for MPP2/I2C SDA MDGPIO_A(23) - DQ(87) AS for MPP2/I2C SCL MDGPIO_A(31:24) - DQ(95:88) Address / Data for MPP2

MDGPIO_EN_REG			**MMR: 1B0, MMR_1: 1B0,**
[RW] 32-bits Access: 8/16/32			**IND: 1B0**
Field Name	**Bits**	**Default**	**Description**
MDGPIO_EN	31:0	0	Turning on the enable will make that pad an output from the chip, turning it off makes that pad input.

MDGPIO_Y_REG			**MMR: 1B4, MMR_1: 1B4,**
[RW] 32-bits Access: 8/16/32			**IND: 1B4**
Field Name	**Bits**	**Default**	**Description**
MDGPIO_Y **(R)**	31:0	0	Reading from this register gives the logic value on the 'p' pin of the corresponding pad.

VID_BUFFER_CONTROL [RW] 32-bits Access: 8/16/32			MMR: 900, MMR_1: 900, IND: 900
Field Name	**Bits**	**Default**	**Description**
CAP0_BUFFER_WATER_MARK	4:0	1	<No Description>
(reserved)	7:5		
CAP1_BUFFER_WATER_MARK	12:8	1	<No Description>
(reserved)	15:13		
FULL_BUFFER_EN	16	0	<No Description> 0 = Disable 1 = Enable
(reserved)	19:17		
VID_BUFFER_RESET	20	0	<No Description> 0 = Not reset 1 = Reset
(reserved)	23:21		
CAP0_BUFFER_EMPTY **(R)**	24	0	<No Description> 0 = Empty 1 = Not empty
(reserved)	27:25		
CAP1_BUFFER_EMPTY **(R)**	28	0	<No Description> 0 = Empty 1 = Not empty
(reserved)	31:29		

DESTINATION_3D_CLR_CMP_VAL [RW] 32-bits Access: 32			MMR: 1820, MMR_1: 1820, IND: 1820
Field Name	**Bits**	**Default**	**Description**
DST_3D_CLR_CMP_CLR	23:0	0	<No Description>
(reserved)	31:24		

DESTINATION_3D_CLR_CMP_MSK [RW] 32-bits Access: 32			MMR: 1824, MMR_1: 1824, IND: 1824
Field Name	**Bits**	**Default**	**Description**
DST_3D_CLR_CMP_MSK	23:0	0	<No Description>
(reserved)	31:24		

MISC_3D_STATE_CNTL_REG [RW] 32-bits Access: 32			MMR: 1CA0, MMR_1: 1CA0, IND: 1CA0
Field Name	**Bits**	**Default**	**Description**
REF_ALPHA	7:0	0	Alpha reference value used when alpha compare enabled.
SCALE_3D_FN	9:8	0	The SCALE_3D_FCN encodes the operation(s) to be performed by the 3D / Scaling pipe. 0 = No operation 1 = Scaling 2 = Texture Mapping/Shading 3 = (Reserved). Note that if this field is set to 0, many 3D/Front-End Scaler/Setup Engine registers are NOT writable. Hence this field should be written to a non-zero value prior to trying to write any other 3D/Front-End Scaler registers. This field is set to 0 on Chip Reset.
SCALE_PIX_REP	10		During Scaling operations, replicate pixels rather than linear blend. 0 = Blend pixels during scale 1 = Replicate pixels during scale
(reserved)	11		
ALPHA_COMB_FCN	13:12	0	Allows modification of how the ALPHA_BLND_SRC and ALPHA_BLND_DST are combined: 0 = Add and Clamp 1 = Add but no Clamp 2 = Subtract Dst from Src and clamp 3 = Subtract Dst from Src but don't clamp
FOG_TABLE_EN	14	0	<No Description> 0=Use Vertex Fog 1=Use Fog Table based on Z interpolator value

(Continued)

MISC_3D_STATE_CNTL_REG [RW] 32-bits Access: 32			MMR: 1CA0, MMR_1: 1CA0, IND: 1CA0
Field Name	**Bits**	**Default**	**Description**
(reserved)	15		
ALPHA_BLND_SRC	19:16	0	Determines the type of SRC alpha blending to use: 0 = BLEND_ZERO. Blend factor is (0,0,0,0) 1 = BLEND_ONE. Blend factor is (1,1,1,1) 2 = BLEND_SRCCOLOUR. Blend factor is (RS,GD,BD,AD) 3 = BLEND_INVSRCCOLOUR. Blend factor is (1-RD,1-GD,1-BD,1-AD) 4 = BLEND_SRCALPHA. Blend factor is As AS, AS) 5 = BLEND_INVSRCALPHA. Blend factor is (1-AS,1-AS,1- AS) 6 = BLEND_DESTALPHA. Blend factor is Add Ad, Ad) 7 = BLEND_INVDESTALPHA. Blend factor is (1-Ad,1- Ad,1- Ad,1- Ad) 8 = BLEND_DESTCOLOUR. Blend factor is (Rd,Gd,Bd,Ad) 9 = BLEND_INVDESTCOLOUR. Blend factor is (1-Rd,1-Gd,1-Bd,1-Ad) 0a = BLEND_SRCALPHASAT. Blend factor is (f,f,f,1); f = min(AS, 1-Ad) 0b = BLRND_BOTHSRCALPHA 0c = BLEND_BOTHINVSRCALPHA 0d-0f = Reserved 11 = SRC Blend factor is (AS,AS,AS,AS), force DST Blend factor to (1-AS,1-AS,1-AS, 1-AS) 12 = SRC Blend factor is (1-AS,1-AS,1-AS,1-AS), force DST Blend factor to (AS,AS,AS, AS)

(Continued)

MISC_3D_STATE_CNTL_REG [RW] 32-bits Access: 32			MMR: 1CA0, MMR_1: 1CA0, IND: 1CA0
Field Name	**Bits**	**Default**	**Description**
ALPHA_BLND_DST	23:20	0	Determines the type of DEST alpha blending to use: 0 = BLEND_ZERO. Blend factor is (0,0,0,0) 1 = BLEND_ONE. Blend factor is (1,1,1,1) 2 = BLEND_SRCCOLOUR. Blend factor is (RS,GS,BS,AS) 3 = BLEND_INVSRCCOLOUR. Blend factor is (1-RS,1-GS,1-BS,1-AS) 4 = BLEND_SRCALPHA. Blend factor is (AS, AS) 5 = BLEND_INVSRCALPHA. Blend factor is (1-AS,1-AS,1- AS) 6 = BLEND_DESTALPHA. Blend factor is (AD, Ad, Ad) 7 = BLEND_INVDESTALPHA. Blend factor is (1-Ad,1- Ad,1- Ad,1- Ad) 8 = BLEND_DESTCOLOUR. Blend factor is (Rd,Gd,Bd,Ad) 9 = BLEND_INVDESTCOLOUR. Blend factor is (1-Rd,1-Gd,1-Bd,1-Ad) 0a = BLEND_SRCALPHASAT. 0b-0f = Reserved
ALPHA_TEST_OP	26:24	0	Specifies what function to use when comparing the SRC Alpha value against a specified Alpha value: 0 = Never Pass 1 = Src < Ref 2 = Src <= Ref 3 = Src == Ref 4 = Src >= Ref 5 = Src > Ref 6 = Src!= Ref 7 = Always Pass
(reserved)	29:27		
CLR_CMP_FCN_3D	31:30	0	**NOTE:** This type of color keying is unavailable when using the old texture interface (execute buffer, DrawPrimitv etc). When the new multi-texture API is used, the APP must use the texel alpha. This is what MS is advocating. Aliased to CLR_CMP_CNTL_3D bits 1:0. 0 = False 1 = True 2 =Texel!= CLR_CMP_CLR_3D 3 = Texel = CLR_CMP_CLR_3D

CONSTANT_COLOR_C [RW] 32-bits Access: 32			MMR: 1D34, MMR_1: 1D34, IND: 1D34
Field Name	**Bits**	**Default**	**Description**
CONSTANT_BLUE	7:0	0	Blue component of constant color that can be used by texture combining.
CONSTANT_GREEN	15:8	0	Green component of constant color that can be used by texture combining.
CONSTANT_RED	23:16	0	Red component of constant color that can be used by texture combining.
CONSTANT_ALPHA	31:24	0	Alpha component of constant color that can be used by texture combining.

PLANE_3D_MASK_C [RW] 32-bits Access: 32			MMR: 1D44, MMR_1: 1D44, IND: 1D44
Field Name	**Bits**	**Default**	**Description**
PLANE_3D_MASK	31:0	0	This MASK is used to perform bitmask operations on color planes. I.E. This value would be written into DP_WRITE_MASK.

8.2 Scratch Pad Registers

BIOS_0_SCRATCH [RW] 32-bits Access: 8/16/32			MMR: 10, MMR_1: 10, IOR: 10, IND: 10
Field Name	**Bits**	**Default**	**Description**
BIOS_0_SCRATCH	31:0	0	Scratch memory for use by video BIOS.

Description:

BIOS Scratch 0.

BIOS_1_SCRATCH [RW] 32-bits Access: 8/16/32			MMR: 14, MMR_1: 14, IOR: 14, IND: 14
Field Name	**Bits**	**Default**	**Description**
BIOS_1_SCRATCH	31:0	0	Scratch memory for use by video BIOS.

Description:

BIOS Scratch 1.

BIOS_2_SCRATCH [RW] 32-bits Access: 8/16/32			MMR: 18, MMR_1: 18, IOR: 18, IND: 18
Field Name	**Bits**	**Default**	**Description**
BIOS_2_SCRATCH	31:0	0	Scratch memory for use by video BIOS.

Description:

BIOS Scratch 2.

BIOS_3_SCRATCH [RW] 32-bits Access: 8/16/32			MMR: 1C, MMR_1: 1C, IOR: 1C, IND: 1C
Field Name	**Bits**	**Default**	**Description**
BIOS_3_SCRATCH	31:0	0	Scratch memory for use by video BIOS.

Description:

BIOS Scratch 3.

This page intentionally left blank.

Appendix A
Reference Table

The following table sorts by mnemonics the RAGE 128 registers described in this reference manual.

Table 8-1 Register Listing

Register Name	Address	Page
ADAPTER_ID	F2C	4-6
ADAPTER_ID_W	4C	4-8
AGP_APER_OFFSET	178	4-24
AGP_BASE	170	4-23
AGP_CNTL	174	4-23
AGP_COMMAND	58	4-22
AGP_PLL_CNTL	10	3-27
AGP_STATUS	54	4-22
AMCGPIO_A_REG	1A0	8-6
AMCGPIO_EN_REG	1A8	8-7
AMCGPIO_MASK	194	8-6
AMCGPIO_Y_REG	1A4	8-7
ATTR10	10	5-44
ATTR11	11	5-45
ATTR12	12	5-45
ATTR13	13	5-46
ATTR14	14	5-47
ATTR[0F:00]	00	5-44
ATTRDR	VGA_IO_3C1	5-43
ATTRDW	VGA_IO_3C0	5-43
ATTRX	VGA_IO_3C0	5-43
AUX1_SC_BOTTOM	1670	7-48
AUX1_SC_LEFT	1664	7-48
AUX1_SC_LEFT	1664	7-48
AUX1_SC_RIGHT	1668	7-48

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
AUX1_SC_RIGHT	1668	7-48
AUX1_SC_TOP	166C	7-48
AUX1_SC_TOP	166C	7-48
AUX2_SC_BOTTOM	1680	7-49
AUX2_SC_BOTTOM	1680	7-49
AUX2_SC_LEFT	1674	7-49
AUX2_SC_LEFT	1674	7-49
AUX2_SC_RIGHT	1678	7-49
AUX2_SC_RIGHT	1678	7-49
AUX2_SC_TOP	167C	7-49
AUX2_SC_TOP	167C	7-49
AUX3_SC_BOTTOM	1690	7-50
AUX3_SC_BOTTOM	1690	7-50
AUX3_SC_LEFT	1684	7-50
AUX3_SC_LEFT	1684	7-50
AUX3_SC_RIGHT	1688	7-50
AUX3_SC_RIGHT	1688	7-50
AUX3_SC_TOP	168C	7-50
AUX3_SC_TOP	168C	7-50
AUX_SC_CNTL	1660	7-47
AUX_SC_CNTL	1660	7-47
BASE_CODE	F0B	4-4
BIOS_0_SCRATCH	10	8-14
BIOS_1_SCRATCH	14	8-14
BIOS_2_SCRATCH	18	8-14
BIOS_3_SCRATCH	1C	8-15
BIOS_ROM	F30	4-6
BIST	F0F	4-5
BRUSH_DATA0	1480	7-18
BRUSH_DATA1	1484	7-18
BRUSH_DATA10	14A8	7-20
BRUSH_DATA11	14AC	7-20

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
BRUSH_DATA12	14B0	7-20
BRUSH_DATA13	14B4	7-20
BRUSH_DATA14	14B8	7-21
BRUSH_DATA15	14BC	7-21
BRUSH_DATA16	14C0	7-21
BRUSH_DATA17	14C4	7-21
BRUSH_DATA18	14C8	7-21
BRUSH_DATA19	14CC	7-22
BRUSH_DATA2	1488	7-18
BRUSH_DATA20	14D0	7-22
BRUSH_DATA21	14D4	7-22
BRUSH_DATA22	14D8	7-22
BRUSH_DATA23	14DC	7-22
BRUSH_DATA24	14E0	7-23
BRUSH_DATA25	14E4	7-23
BRUSH_DATA26	14E8	7-23
BRUSH_DATA27	14EC	7-23
BRUSH_DATA28	14F0	7-23
BRUSH_DATA29	14F4	7-24
BRUSH_DATA3	148C	7-18
BRUSH_DATA30	14F8	7-24
BRUSH_DATA31	14FC	7-24
BRUSH_DATA32	1500	7-24
BRUSH_DATA33	1504	7-24
BRUSH_DATA34	1508	7-25
BRUSH_DATA35	150C	7-25
BRUSH_DATA36	1510	7-25
BRUSH_DATA37	1514	7-25
BRUSH_DATA38	1518	7-25
BRUSH_DATA39	151C	7-26
BRUSH_DATA4	1490	7-19
BRUSH_DATA40	1520	7-26

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
BRUSH_DATA41	1524	7-26
BRUSH_DATA42	1528	7-26
BRUSH_DATA43	152C	7-26
BRUSH_DATA44	1530	7-27
BRUSH_DATA45	1534	7-27
BRUSH_DATA46	1538	7-27
BRUSH_DATA47	153C	7-27
BRUSH_DATA48	1540	7-27
BRUSH_DATA49	1544	7-28
BRUSH_DATA5	1494	7-19
BRUSH_DATA50	1548	7-28
BRUSH_DATA51	154C	7-28
BRUSH_DATA52	1550	7-28
BRUSH_DATA53	1554	7-28
BRUSH_DATA54	1558	7-29
BRUSH_DATA55	155C	7-29
BRUSH_DATA56	1560	7-29
BRUSH_DATA57	1564	7-29
BRUSH_DATA58	1568	7-29
BRUSH_DATA59	156C	7-30
BRUSH_DATA6	1498	7-19
BRUSH_DATA60	1570	7-30
BRUSH_DATA61	1574	7-30
BRUSH_DATA62	1578	7-30
BRUSH_DATA63	157C	7-30
BRUSH_DATA7	149C	7-19
BRUSH_DATA8	14A0	7-19
BRUSH_DATA9	14A4	7-20
BRUSH_SCALE	1470	7-31
BRUSH_Y_X	1474	7-31
BUS_CNTL	30	4-25
BUS_CNTL1	34	3-45

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
CACHE_LINE	F0C	4-4
CAPABILITIES_ID	F50	4-8
CAPABILITIES_PTR	F34	4-7
CLK_PIN_CNTL	01	3-12
CLOCK_CNTL_DATA	0C	3-12
CLOCK_CNTL_INDEX	08	3-12
CLR_CMP_CLR_3D	1A24	7-54
CLR_CMP_CLR_3D	1A24	7-54
CLR_CMP_CLR_DST	15C8	7-53
CLR_CMP_CLR_SRC	15C4	7-53
CLR_CMP_CNTL	15C0	7-53
CLR_CMP_CNTL	15C0	7-53
CLR_CMP_MSK	15CC	7-54
CLR_CMP_MSK	15CC	7-54
CLR_CMP_MSK_3D	1A28	7-54
CLR_CMP_MSK_3D	1A28	7-54
COMMAND	F04	4-1
COMPOSITE_SHADOW_ID	1A0C	7-10
COMPOSITE_SHADOW_ID	1A0C	7-10
CONFIG_APER_0_BASE	100	4-11
CONFIG_APER_1_BASE	104	4-11
CONFIG_APER_SIZE	108	4-13
CONFIG_BONDS	E8	4-10
CONFIG_CNTL	E0	4-9
CONFIG_MEMSIZE	F8	4-11
CONFIG_MEMSIZE_EMBEDDED	114	4-15
CONFIG_REG_1_BASE	10C	4-13
CONFIG_REG_APER_SIZE	110	4-14
CONFIG_XSTRAP	E4	4-9
CONSTANT_COLOR_C	1D34	8-13
CRC_CMDFIFO_ADDR	740	3-11
CRC_CMDFIFO_DOUT	744	3-11

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
CRT00	00	5-13
CRT00_S	40	5-28
CRT01	01	5-14
CRT01_S	41	5-29
CRT02	02	5-14
CRT02_S	42	5-29
CRT03	03	5-15
CRT04	04	5-15
CRT04_S	44	5-29
CRT05	05	5-16
CRT05_S	45	5-30
CRT06	06	5-16
CRT06_S	46	5-30
CRT07	07	5-17
CRT07_S	47	5-30
CRT08	08	5-18
CRT08_S	48	5-31
CRT09	09	5-18
CRT09_S	49	5-31
CRT0A	0A	5-19
CRT0A_S	4A	5-31
CRT0B	0B	5-20
CRT0B_S	4B	5-32
CRT0C	0C	5-20
CRT0C_S	4C	5-32
CRT0D	0D	5-21
CRT0D_S	4D	5-32
CRT0E	0E	5-21
CRT0E_S	4E	5-32
CRT0F	0F	5-22
CRT0F_S	4F	5-32
CRT10	10	5-22

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
CRT10_S	50	5-33
CRT11	11	5-23
CRT11_S	51	5-33
CRT12	12	5-23
CRT12_S	52	5-33
CRT13	13	5-24
CRT13_S	53	5-33
CRT14	14	5-24
CRT14_S	54	5-34
CRT15	15	5-25
CRT15_S	55	5-34
CRT16	16	5-25
CRT16_S	56	5-34
CRT17	17	5-26
CRT17_S	57	5-34
CRT18	18	5-27
CRT18_S	58	5-35
CRT1E	1E	5-27
CRT1E_S	5E	5-35
CRT1F	1F	5-28
CRT1F_S	5F	5-35
CRT22	22	5-28
CRT22_S	CRT_62	5-36
CRTC8_DATA	VGA_IO_3B5_3D5	5-13
CRTC8_IDX	VGA_IO_3B4_3D4	5-13
CRTC_CRNT_FRAME	214	6-13
CRTC_DEBUG	21C	5-36
CRTC_EXT_CNTL	54	6-3
CRTC_GEN_CNTL	50	6-1
CRTC_GUI_TRIG_VLINE	218	6-9
CRTC_H_SYNC_STRT_WID	204	6-7
CRTC_H_TOTAL_DISP	200	6-6

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
CRTC_OFFSET	224	6-10
CRTC_OFFSET_CNTL	228	6-11
CRTC_PITCH	22C	6-13
CRTC_STATUS	5C	6-6
CRTC_V_SYNC_STRT_WID	20C	6-8
CRTC_V_TOTAL_DISP	208	6-7
CRTC_VLINE_CRNT_VLINE	210	6-8
CUR_CLR0	26C	6-20
CUR_CLR1	270	6-21
CUR_HORZ_VERT_OFF	268	6-19
CUR_HORZ_VERT_POSN	264	6-18
CUR_OFFSET	260	6-18
DAC_CNTL	58	6-36
DAC_CRC_SIG	2CC	6-39
DAC_DATA	VGA_IO_3C9	5-7
DAC_MASK	VGA_IO_3C6	5-7
DAC_R_INDEX	VGA_IO_3C7	5-8
DAC_W_INDEX	VGA_IO_3C8	5-8
DDA_CONFIG	2E0	6-14
DDA_ON_OFF	2E4	6-14
DEFAULT_OFFSET	16E0	7-44
DEFAULT_PITCH	16E4	7-44
DEFAULT_SC_BOTTOM_RIGHT	16E8	7-45
DESTINATION_3D_CLR_CMP_MSK	1824	8-10
DESTINATION_3D_CLR_CMP_VAL	1820	8-9
DEVICE_ID	F02	4-1
DP_BRUSH_BKGD_CLR	1478	7-32
DP_BRUSH_FRGD_CLR	147C	7-32
DP_CNTL	16C0	7-33
DP_CNTL_XDIR_YDIR_YMAJOR	16D0	7-37
DP_DATATYPE	16C4	7-35
DP_DATATYPE	16C4	7-35

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
DP_GUI_MASTER_CNTL	146C	7-38
DP_GUI_MASTER_CNTL	146C	7-38
DP_GUI_MASTER_CNTL	146C	7-41
DP_GUI_MASTER_CNTL	146C	7-41
DP_MIX	16C8	7-37
DP_MIX	16C8	7-37
DP_SRC_BKGD_CLR	15DC	7-32
DP_SRC_BKGD_CLR	15DC	7-32
DP_SRC_FRGD_CLR	15D8	7-32
DP_SRC_FRGD_CLR	15D8	7-32
DP_WRITE_MSK	16CC	7-38
DST_BRES_DEC	1630	7-8
DST_BRES_DEC	1630	7-8
DST_BRES_ERR	1628	7-7
DST_BRES_ERR	1628	7-7
DST_BRES_INC	162C	7-8
DST_BRES_INC	162C	7-8
DST_BRES_LNTH	1634	7-7
DST_BRES_LNTH	1634	7-7
DST_BRES_LNTH_SUB	1638	7-9
DST_BRES_LNTH_SUB	1638	7-9
DST_HEIGHT	1410	7-4
DST_HEIGHT_WIDTH	143C	7-4
DST_HEIGHT_WIDTH	143C	7-4
DST_HEIGHT_WIDTH_8	158C	7-6
DST_HEIGHT_WIDTH_8	158C	7-6
DST_HEIGHT_WIDTH_BW	15B4	7-5
DST_HEIGHT_Y	15A0	7-6
DST_HEIGHT_Y	15A0	7-6
DST_OFFSET	1404	7-1
DST_PITCH	1408	7-2
DST_PITCH	1408	7-2

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
DST_PITCH_OFFSET	142C	7-1
DST_PITCH_OFFSET	142C	7-1
DST_PITCH_OFFSET_C	1C80	7-10
DST_PITCH_OFFSET_C	1C80	7-10
DST_PITCH_OFFSET_C	1C80	7-10
DST_WIDTH	140C	7-4
DST_WIDTH	140C	7-4
DST_WIDTH_BW	15B4	7-9
DST_WIDTH_HEIGHT	1598	7-5
DST_WIDTH_X	1588	7-6
DST_WIDTH_X_INCY	159C	7-7
DST_X	141C	7-2
DST_X	141C	7-2
DST_X_SUB	15A4	7-8
DST_X_Y	1594	7-3
DST_X_Y	1594	7-3
DST_X_Y	1594	7-3
DST_Y	1420	7-2
DST_Y_SUB	15A8	7-8
DST_Y_X	1438	7-3
EXT_MEM_CNTL	144	6-29
FCP_CNTL	12	3-27
FLUSH_1	1704	7-62
FLUSH_2	1708	7-62
FLUSH_3	170C	7-63
FLUSH_4	1710	7-63
FLUSH_5	1714	7-63
FLUSH_6	1718	7-63
FLUSH_7	171C	7-63
GEN_INT_CNTL	40	8-1
GEN_INT_STATUS	44	8-2
GEN_RESET_CNTL	F0	8-4

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
GENENB	VGA_IO_3C3	5-6
GENFC_RD	VGA_IO_3CA	5-4
GENFC_WT	VGA_IO_3BA_3DA	5-4
GENMO_RD	VGA_IO_3CC	5-2
GENMO_WT	VGA_IO_3C2	5-1
GENS0	VGA_IO_3C2	5-5
GENS1	VGA_IO_3BA_3DA	5-5
GIU_SCRATCH_REG0	15E0	7-57
GIU_SCRATCH_REG1	15E4	7-57
GIU_SCRATCH_REG2	15E8	7-57
GIU_SCRATCH_REG3	15EC	7-57
GIU_SCRATCH_REG4	15F0	7-58
GIU_SCRATCH_REG5	15F4	7-58
GPIO_MONID	68	3-1
GPIO_MONIDB	6C	3-2
GRA00	GRPH_00	5-37
GRA01	GRPH_01	5-38
GRA02	GRPH_02	5-38
GRA03	GRPH_03	5-39
GRA04	GRPH_04	5-39
GRA05	GRPH_05	5-40
GRA06	GRPH_06	5-41
GRA07	GRPH_07	5-41
GRA08	GRPH_08	5-42
GRPH8_DATA	VGA_IO_3CF	5-37
GRPH8_IDX	VGA_IO_3CE	5-37
GUI_DEBUG0	16A0	7-59
GUI_DEBUG1	16A4	7-59
GUI_DEBUG2	16A8	7-59
GUI_DEBUG3	16AC	7-60
GUI_DEBUG4	16B0	7-60
GUI_DEBUG5	16B4	7-60

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
GUI_DEBUG6	16B8	7-60
GUI_PROBE	16BC	7-61
GUI_STAT	1740	7-58
HEADER	F0E	4-5
HOST_DATA[7:0]	17C0-17DC	7-17
HOST_DATA[7:0]	17C0_17DC	7-17
HOST_DATA_LAST	17E0	7-17
HOST_PATH_CNTL	130	3-4
HTOTAL_CNTL	09	3-19
HW_DEBUG	128	3-10
INTERRUPT_LINE	3C	4-7
INTERRUPT_PIN	F3D	4-7
IO_BASE	F14	4-6
LATENCY	F0D	4-5
LEAD_BRES_DEC	1608	7-11
LEAD_BRES_ERR	1600	7-10
LEAD_BRES_INC	1604	7-11
LEAD_BRETH_LNTH	161C	7-11
LEAD_BRETH_LNTH_SUB	1624	7-13
MAX_LATENCY	F3F	4-8
MCLK_CNTL	0F	3-26
MDGPIO_A_REG	1AC	8-8
MDGPIO_EN_REG	1B0	8-8
MDGPIO_MASK	198	8-6
MDGPIO_Y_REG	1B4	8-8
MEM_ADDR_CONFIG	148	6-22
MEM_BASE	F10	4-5
MEM_CNTL	140	6-25
MEM_INIT_LAT_TIMER	154	6-34
MEM_INTF_CNTL	14C	6-30
MEM_SDRAM_MODE_REG	158	6-34
MEM_STR_CNTL	150	6-32

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
MEM_VGA_RP_SEL	3C	6-25
MEM_VGA_WP_SEL	38	6-25
MIN_GRANT	F3E	4-7
MISC_3D_STATE_CNTL_REG	1CA0	8-10
MM_DATA	04	4-15
MM_INDEX	00	4-15
MPLL_CNTL	0E	3-25
N_VIF_COUNT	248	6-24
OVR_CLR	230	6-16
OVR_WID_LEFT_RIGHT	234	6-16
OVR_WID_TOP_BOTTOM	238	6-17
PALETTE_DATA	B4	6-40
PALETTE_INDEX	B0	6-39
PC_DEBUG_MODE	1760	3-41
PC_GUI_CTLSTAT	1748	3-33
PC_NGUI_CTLSTAT	184	3-37
PC_NGUI_MODE	180	3-31
PCI_GART_PAGE	17C	4-12
PLANE_3D_MASK_C	1D44	8-13
PLL_DIV_[3:0]	04_07	3-16
PLL_TEST_CNTL	13	3-28
PMI_CAP_ID	5C	3-42
PMI_DATA	F63	3-44
PMI_NXT_CAP_PTR	5D	3-43
PMI_PMC_REG	F5E	3-43
PMI_PMCSR_REG	F60	3-44
PMI_REGISTER	F5C	3-42
PPLL_CNTL	02	3-14
PPLL_REF_DIV	03	3-14
PWR_MNGMT_CNTL_STATUS	F60	4-8
REG_BASE	F18	4-6
REGPROG_ID	F09	4-4

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
REVISION_ID	F08	4-3
SC_BOTTOM	164C	7-47
SC_BOTTOM	164C	7-47
SC_BOTTOM_RIGHT	16F0	7-51
SC_BOTTOM_RIGHT	16F0	7-51
SC_BOTTOM_RIGHT	16F0	7-52
SC_BOTTOM_RIGHT	16F0	7-52
SC_BOTTOM_RIGHT	16F0	7-52
SC_BOTTOM_RIGHT	16F0	7-52
SC_BOTTOM_RIGHT_C	1C8C	7-52
SC_LEFT	1640	7-46
SC_RIGHT	1644	7-46
SC_RIGHT	1644	7-46
SC_TOP	1648	7-46
SC_TOP	1648	7-46
SC_TOP_LEFT	16EC	7-51
SC_TOP_LEFT	16EC	7-51
SC_TOP_LEFT_C	1C88	7-52
SEQ00	SEQ_00	5-9
SEQ01	SEQ_01	5-10
SEQ02	SEQ_02	5-11
SEQ03	SEQ_03	5-11
SEQ04	SEQ_04	5-12
SEQ8_DATA	VGA_IO_3C5	5-9
SEQ8_IDX	VGA_IO_3C4	5-9
SNAPSHOT_F_COUNT	244	6-23
SNAPSHOT_VH_COUNTS	240	6-23
SNAPSHOT_VIF_COUNT	C4	6-24
SRC_OFFSET	15AC	7-14
SRC_OFFSET	15AC	7-14
SRC_PITCH	15B0	7-14
SRC_PITCH_OFFSET	1428	7-14

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
SRC_PITCH_OFFSET	1428	7-14
SRC_SC_BOTTOM	165C	7-16
SRC_SC_BOTTOM	165C	7-16
SRC_SC_BOTTOM_RIGHT	16F4	7-16
SRC_SC_BOTTOM_RIGHT	16F4	7-16
SRC_SC_BOTTOM_RIGHT	16F4	7-16
SRC_SC_RIGHT	1654	7-16
SRC_SC_RIGHT	1654	7-16
SRC_X	1414	7-15
SRC_X_Y	1590	7-15
SRC_Y	1418	7-15
SRC_Y	1418	7-15
SRC_Y_X	1434	7-15
STATUS	F06	4-2
SUB_CLASS	F0A	4-4
SURFACE0_INFO	B0C	4-18
SURFACE0_LOWER_BOUND	B04	4-16
SURFACE0_UPPER_BOUND	B08	4-17
SURFACE1_INFO	B1C	4-19
SURFACE1_LOWER_BOUND	B14	4-16
SURFACE1_UPPER_BOUND	B18	4-17
SURFACE2_INFO	B2C	4-20
SURFACE2_LOWER_BOUND	B24	4-16
SURFACE2_UPPER_BOUND	B28	4-18
SURFACE3_INFO	B3C	4-21
SURFACE3_LOWER_BOUND	B34	4-17
SURFACE3_UPPER_BOUND	B38	4-18
SURFACE_DELAY	B00	4-16
SW_SEMAPHORE	13C	8-5
TEST_DEBUG_CNTL	120	3-6
TEST_DEBUG_MUX	124	3-8
TEST_DEBUG_OUT	12C	3-9

Table 8-1 Register Listing (Continued)

Register Name	Address	Page
TRAIL_BRES_DEC	1614	7-12
TRAIL_BRES_ERR	160C	7-11
TRAIL_BRES_INC	1610	7-12
TRAIL_X	1618	7-12
TRAIL_X_SUB	1620	7-12
VCLK_ECP_CNTL	08	3-17
VENDOR_ID	F00	4-1
VGA_DDA_0N_OFF	2EC	6-15
VID_BUFFER_CONTROL	900	8-9
VIDEOMUX_CNTL	190	3-3
WAIT_UNTIL	1720	7-55
X_MPLL_REF_FB_DIV	0A	3-21
XCLK_CNTL	0D	3-24
XDLL_CNTL	0C	3-22
XPLL_CNTL	0B	3-22