M76
Register Reference Guide

Technical Reference Manual
Rev 1.01o

P/N: 42590_m76_rrg_1.01o

Advanced Micro Devices, Inc., will not provide any indemnity, pay any royalty, nor provide any license/sublicense to any:

(a) Intellectual property rights relating to any of the following: (i) Macrovision for its Analog Protection System ("APS") technologies; (ii) Advanced Television Systems Committee (ATSC) standard and related technologies; or (iii) the High Definition Multimedia Interface (HDMI) standard and related technologies; or

(b) Audio and/or video codecs or any industry standard technology (e.g., technology or specifications promulgated by any standards development organization, consortium, trade association, special interest group or like entity) .

Trademarks

AMD, the AMD Arrow logo, Athlon, and combinations thereof, ATI, ATI logo, and Radeon are trademarks of Advanced Micro Devices, Inc.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Product Notice:

This device is protected by U.S. patent numbers 4,631,603; 4,577,216; and 4,819,098; and 6,516,132; and other intellectual property rights. The use of Macrovision's copy protection technology in the device must be authorized by Macrovision and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by Macrovision. Reverse engineering or disassembly is prohibited.

Protection Notice:

This device may only be sold or distributed to: (i) a Macrovision Authorized Buyer, (ii) a customer (PMA Customer) who has executed a Proprietary Materials Agreement (PMA) with Macrovision that is still in effect, (iii) a contract manufacturer approved by Macrovision to purchase this device on behalf of a Macrovision Authorized Buyer or a PMA Customer, or (iv) a distributor who has executed a Macrovision-specified distribution agreement with ATI.

Disclaimer

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel, or otherwise, to any intellectual property rights are granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Table of Contents

Chapter 1: Introduction

1.1 About this Manual 1-1
1.2 Nomenclature and Conventions 1-1
1.2.1 Numeric Representations 1-1
1.2.2 Register Description 1-1

Chapter 2: Registers Description

2.1 Memory Controller Registers 2-2
2.2 Bus Interface Registers 2-51
2.3 PCI-E Registers 2-54
2.4 Clock Generator Registers 2-89
2.5 VIP/I2C Registers 2-97
2.5.1 I2C Registers 2-97
2.5.2 Video Interface Port Host Port Registers 2-110
2.5.3 Capture Registers 2-116
2.5.4 VIP Host Port DMA Registers 2-126
2.5.5 General Purpose I/O Data and Control Registers 2-132
2.5.6 VIP Miscellaneous Registers 2-133
2.6 Video Graphics Array (VGA) Registers 2-139
2.6.1 VGA Control/Status Registers 2-139
2.6.2 VGA DAC Control Registers 2-141
2.6.3 VGA Sequencer Registers 2-142
2.6.4 VGA CRT Registers 2-143
2.6.5 VGA Graphics Registers 2-151
2.6.6 VGA Attribute Registers 2-153
2.6.7 VGA Miscellaneous Registers 2-159
2.7 Display Controller Registers 2-170
2.7.1 Primary Display Graphics Control Registers 2-170
2.7.2 Primary Display Video Overlay Control Registers 2-176
2.7.3 Primary Display Video Overlay Transform Registers 2-181
2.7.4 Primary Display Video Overlay Gamma Correction Registers 2-184
2.7.5 Primary Display Graphics and Overlay Blending Registers 2-188
2.7.6 Primary Display Color Matrix Transform Registers 2-192
2.7.7 Primary Display Subsampling Registers 2-196
2.7.8 Primary Display Realtime Overlay Registers 2-196
2.7.9 Primary Display Hardware Cursor Registers 2-198
2.7.10 Primary Display Hardware Icon Registers 2-200
2.7.11 Primary Display Multi-VPU Control Registers 2-202
2.7.12 Secondary Display Graphics Control Registers 2-208
2.7.13 Secondary Display Video Overlay Control Registers 2-215
2.7.14 Secondary Display Video Overlay Transform Registers 2-219
2.7.15 Secondary Display Video Overlay Gamma Correction Registers 2-223
2.7.16 Secondary Display Graphics and Overlay Blending Registers 2-227
2.7.17 Secondary Display Color Matrix Transform Registers 2-230
2.7.18 Secondary Display Subsampling Registers 2-233

2.7.19 Secondary Display Realtime Overlay Registers 2-234
2.7.20 Secondary Display Hardware Cursor Registers 2-235
2.7.21 Secondary Display Hardware Icon Registers 2-238
2.7.22 Secondary Display Multi-VPU Control Registers 2-240
2.7.23 Display Look Up Table Control Registers 2-241
2.7.24 Display Controller Look Up Table A Registers 2-244
2.7.25 Display Controller Look Up Table B Registers 2-247
2.7.26 Display Controller CRC Registers 2-250
2.7.27 Display/Memory Interface Control and Status Registers 2-251
2.7.28 MCIF Control Registers 2-253
2.7.29 Display Controller to Line Buffer Control Registers 2-254
2.7.30 Multi VPU Control Registers 2-254
2.8 CRTC Control Registers 2-255
2.8.1 Primary Display CRTC Control Registers 2-255
2.8.2 Secondary Display CRTC Control Registers 2-271
2.9 Display Output Registers 2-288
2.9.1 Digital to Analog Converter (DAC) Registers 2-288
2.9.2 Display Output Control Registers 2-342

Appendix A: Cross Referenced Index

A.1 Quick Cross-Reference Index A-1
A.2 Configuration Registers Sorted by Name A-2
A.3 Configuration Registers Sorted by Address A-5
A.4 Clock Registers Sorted by Name A-8
A.5 Clock Registers Sorted by Address A-9
A.6 Display Controller Registers Stored by Name A-10
A.7 Display Controller Registers Stored by Address A-26
A.8 Host Interface Decode Space Registers Sorted by Name A-42
A.9 Memory Controller Registers Sorted By Name A-43
A.10 Memory Controller Registers Sorted By Address A-47
A.11 PCIE Registers Sorted By Name A-51
A.12 PCIE Registers Sorted By Address A-53
A.13 VIP Registers Sorted By Name A-55
A.14 VIP Registers Sorted By Address A-58
A.15 VGA ATTR Registers Sorted By Name A-61
A.16 VGA CRT Registers Sorted By Name A-62
A.17 VGA GRPH Registers Sorted By Name A-63
A.18 VGA SEQ Registers Sorted By Name A-64
A.19 All Registers Sorted by Name A-65

Appendix B: Revision History

Chapter 1
Introduction

1.1 About this Manual

This manual serves as a register reference guide to the M76 graphics controller.

- *Chapter 1* outlines the notations and conventions used throughout this manual.
- *Chapter 2* provides a detailed description of the registers.
- *Appendix A* provides several cross-referenced lists (sorted by Register Name and Address).

1.2 Nomenclature and Conventions

1.2.1 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].

1.2.2 Register Description

All registers in this document are described with the format of the sample table below. All offsets are in hexadecimal notation, while programmed bits are in either binomial or hexadecimal notation.

DST_HEIGHT_WIDTH_8 - W - 32 bits - [MMReg:0x158C]			
Field Name	**Bits**	**Default**	**Description**
DST_WIDTH *(mirror bits 0:7 of DST_WIDTH:DST_WIDTH)*	23:16	0x0	Destination Width Note: This is an initiator register. Y is incremented at end of blit. Write 15: 0 to E2_DST_X, Write 31: 16 to E2_DST_WIDTH, then signal blit_start. E2_DST_Y = E2_DEST_Y (+/-) E2_DST_HEIGHT as function of direction after blit is complete
DST_HEIGHT *(mirror bits 0:7 of DST_HEIGHT:DST_HEIGHT)*	31:24	0x0	Destination Height Write 15: 0 to E2_DST_Y, Write 31: 16 to E2_DST_HEIGHT

[W] (Reserved) 15: 0 DST_WIDTH 23: 16 Destination width: range 0 to 256 (ZERO extent)

Table 1-1 Register description table notation

Register Information	Example
Register name	DST_HEIGHT_WIDTH_8
Read / Write capability R = Readable W = Writable RW = Readable and Writable	W
Register size	32 bits
Register address(es)*	MMReg:0x158C
Field name	DST_WIDTH
Field position/size	23:16
Field default value	0x0
Field description	Destination....complete
Field mirror information	(mirror bits 0:7 of DST_WIDTH:DST_WIDTH)
Brief register description	[W] (Reserved) 15: 0 DST_WIDTH 23: 16 Destination width: range 0 to 256 (ZERO extent)
* Note: There may be more than one address; the convention used is as follows: **[aperName:offset]** - single mapping, to one aperture/decode and one offset **[aperName1, aperName2, …, aperNameN:offset]** - multiple mappings to different apertures/decodes but same offset **[aperName:startOffset-endOffset]** - mapped to an offset range in the same aperture/decode	

Chapter 2
Registers Description

To link to a topic of interest, use the following list of hypertext linked cross references:

"Memory Controller Registers" on page 2-2

"Bus Interface Registers" on page 2-51

"PCI-E Registers" on page 2-54

"Clock Generator Registers" on page 2-89

"VIP/I2C Registers" on page 2-97

"Video Graphics Array (VGA) Registers" on page 2-139

"Display Controller Registers" on page 2-170

"CRTC Control Registers " on page 2-255

"Display Output Registers" on page 2-288

2.1 Memory Controller Registers

MC_CONFIG - RW - 32 bits - [GpuF0MMReg:0x2000]			
Field Name	**Bits**	**Default**	**Description**
MCDW_WR_ENABLE	0	0x1	0=MCDW will ignore SRBM writes 1=MCDW will accept SRBM writes
MCDX_WR_ENABLE	1	0x1	0=MCDX will ignore SRBM writes 1=MCDX will accept SRBM writes
MCDY_WR_ENABLE	2	0x1	0=MCDY will ignore SRBM writes 1=MCDY will accept SRBM writes
MCDZ_WR_ENABLE	3	0x1	0=MCDZ will ignore SRBM writes 1=MCDZ will accept SRBM writes
MCB_WR_ENABLE	4	0x1	0=MCB will ignore SRBM writes 1=MCB will accept SRBM writes
MC_RD_ENABLE	7:5	0x0	0=SRBM reads will be directed at MCDW 1=SRBM reads will be directed at MCDX 2=SRBM reads will be directed at MCDY 3=SRBM reads will be directed at MCDZ 4=SRBM reads will be directed at MCB 5=Undefined 6=Undefined 7=All SRBM reads will be dropped by all MC tiles

MC_SEQ_CNTL - RW - 32 bits - [GpuF0MMReg:0x2600]			
Field Name	**Bits**	**Default**	**Description**
MEM_ADDR_MAP_COLS	1:0	0x0	0=2**8 columns 1=2**9 columns 2=2**10 columns 3=reserved
MEM_ADDR_MAP_BANK	2	0x0	0=4 banks 1=8 banks
SAFE_MODE	5:4	0x0	0=Disable safe mode 1=Ensure closing all pages before doing refresh 2=Ensure closing page before access a different page in the same bank 3=Reserved
CHANNEL_DISABLE	9:8	0x0	This field allows the user to disable the mclk branch for the specific unused channel. NOT FOR 600
PIPE_DELAY_OUT	12	0x0	This field specifies pipeline delay between mc & io. This field is NOT CONFIGURABLE for a specific ASIC for 600: 0 for 610: 0 for 630: 1 0=No pipeline delay between MC/IO for outgoing signals 1=pipeline delay
PIPE_DELAY_IN	13	0x0	This field specifies pipeline delay between mc & io. This field is NOT CONFIGURABLE for a specific ASIC for 600: 0 for 610: 0 for 630: 1 0=No pipeline delay between MC/IO for incoming signals 1=pipeline delay
MSKOFF_DAT_TL	16	0x0	for the byte which has data mask on, tie the corresponding dq to 0. ONLY 1 bit could be set to 1 among MSKOFF_DAT_TL, MSKOFF_DAT_TH, MSKOFF_DAT_AC 1=Tie low for the DQ whose corresponding DQM is on
MSKOFF_DAT_TH	17	0x0	for the byte which has data mask on, tie the corresponding dq to 1. ONLY 1 bit could be set to 1 among MSKOFF_DAT_TL, MSKOFF_DAT_TH, MSKOFF_DAT_AC 1=Tie high for the DQ whose corresponding DQM is on

MSKOFF_DAT_AC	18	0x0	for the byte which has data mask on, keep the previous dq value to avoid toggleing. ONLY 1 bit could be set to 1 among MSKOFF_DAT_TL, MSKOFF_DAT_TH, MSKOFF_DAT_AC 1=no toggling for the DQ whose corresponding DQM is on

This register specifies specific seq configuration

MC_SEQ_DRAM - RW - 32 bits - [GpuF0MMReg:0x2608]			
Field Name	**Bits**	**Default**	**Description**
ADR_2CK	0	0x0	Number of cycle(s) to send an address. One cycle for non-DDR4. Two cycles for DDR4. 0=One-cycle address 1=Two-cycle address
ADR_MUX	1	0x1	Address bus is shared between two channels or not. Not shared for DDR4. Shared for non-DDR4. 0=Address bus is not shared 1=Address bus is shared
ADR_DF1	2	0x1	Default value for address bus (during NOP). 0=Address default low 1=Address default high
AP8	3	0x0	Location of auto-precharge bit. 0=AP bit starts at MSB+1 1=AP bit is bit 8
DAT_DF1	4	0x1	Default value for data bus. 0=DAT default low 1=DAT default high
DQS_DF1	5	0x1	Default value for write strobes. 0=DQS default low 1=DQS default high
DQM_DF1	6	0x1	Default value for write mask. 0=DQM default low 1=DQM default high
DQM_ACT	7	0x1	Polarity of data mask. Active low for DDR4. Active high for non- DDR4. 0=DQM active low 1=DQM active high
STB_CNT	11:8	0xf	DRAM standby counter. Number of idle cycles before dynamic CKE is enabled. This prevents the CKE from turning off too easily.
CKE_DYN	12	0x0	Dynamic CKE. 0=Disable 1=Enable
CKE_ACT	13	0x1	Polarity of clock enable. Active low for DDR4. Active high for non- DDR4. 0=Active low 1=Active high
BO4	14	0x0	DRAM burst size. 0=DRAM is burst of 8 1=DRAM is burst of 4
DLL_CLR	15	0x0	Resets DLL lock timer. DRAM power up is completed once the DLL lock time is reached. If the DLL lock timer is reset, the DRAM power up flag is deasserted. 0=Not reset DLL timer 1=Reset DLL timer
DLL_CNT	23:16	0xf	DRAM DLL lock time in multiples of 256 mclk cycles.
DAT_INV	24	0x0	Enables/disables DDR write data inversion mode. 0=Disable write data inversion 1=Enable write data inversion

INV_ACM	25	0x1	Selects DDR write data inversion mode. 0=DC mode 1=AC mode
ODT_ENB	26	0x0	0=Disable ODT 1=Enable ODT
ODT_ACT	27	0x1	0=ODT active low 1=ODT active high
RST_CTL	28	0x1	Controls DRAM reset pin. Channel B only. 0=Drive reset low 1=Drive reset high
TRI_MIO_DYN	29	0x0	1=Tristate cmd/data/addr during dynamic cke

This register specifies the character of the DRAM interface.

MC_SEQ_RAS_TIMING_P - RW - 32 bits - [GpuF0MMReg:0x260C]			
Field Name	**Bits**	**Default**	**Description**
TRCDW	4:0	0xa	Number of cycles from active to write - 1.
TRCDWA	9:5	0xa	Number of cycles from active to write with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDW.
TRCDR	14:10	0xd	Number of cycles from active to read - 1.
TRCDRA	19:15	0xd	Number of cycles from active to read with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDR.
TRRD	23:20	0x5	Number of cycles from active bank a to active bank b - 1.
TRC	30:24	0x27	Number of cycles from active to active/auto refresh - 1.

RAS related parameters in hclk cycles for performance mode.

MC_SEQ_CAS_TIMING_P - RW - 32 bits - [GpuF0MMReg:0x2610]			
Field Name	**Bits**	**Default**	**Description**
TNOPW	1:0	0x0	Extra cycle(s) between successive write bursts. For debugging purpose only.
TNOPR	3:2	0x0	Extra cycle(s) between successive read bursts. For debugging purpose only.
TR2W	8:4	0x9	Read to write turn around time - 1.
TR2R	15:12	0x5	Read to read time - 1 (different rank).
TW2R	20:16	0x9	Write to read turn around time - 1.
TCL	28:24	0x6	CAS to data return latency - 2 (0 to 20).

CAS related paramters in hclk cycles for performance mode.

MC_SEQ_MISC_TIMING_P - RW - 32 bits - [GpuF0MMReg:0x2614]			
Field Name	**Bits**	**Default**	**Description**
TRP_WRA	5:0	0x15	From write with auto-prechrage to active - 1.
TCKE_HI	7:6	0x0	2 MSB of tCKE parameters, used to control exit power down time.
TRP_RDA	13:8	0x11	From read with auto-prechrage to active - 1.
TRP	19:16	0xb	Precharge command period - 1.
TRFC	26:20	0x2f	Auto-refresh command period - 1.
TCKE	31:28	0x4	4 LSB CKE power down exit timer.

Misc. DRAM parameters in hclk cycles for performance mode.

MC_SEQ_MISC_TIMING2_P - RW - 32 bits - [GpuF0MMReg:0x2618]			
Field Name	**Bits**	**Default**	**Description**
PA2RDATA	2:0	0x0	Read Preamble for DDR4.
PA2WDATA	6:4	0x0	Write Preamble for DDR4.
FAW	12:8	0x0	Four Active Window/2 - 5 in MCLK

TCKE_PULSE	19:16	0x0	minimum power down period/power up period

Misc. DRAM parameters in hclk cycles for performance mode.

MC_SEQ_RAS_TIMING_B - RW - 32 bits - [GpuF0MMReg:0x261C]			
Field Name	**Bits**	**Default**	**Description**
TRCDW	4:0	0xa	Number of cycles from active to write - 1.
TRCDWA	9:5	0xa	Number of cycles from active to write with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDW.
TRCDR	14:10	0xd	Number of cycles from active to read - 1.
TRCDRA	19:15	0xd	Number of cycles from active to read with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDR.
TRRD	23:20	0x5	Number of cycles from active bank a to active bank b - 1.
TRC	30:24	0x27	Number of cycles from active to active/auto refresh - 1.

RAS related parameters in hclk cycles for balanced mode

MC_SEQ_CAS_TIMING_B - RW - 32 bits - [GpuF0MMReg:0x2620]			
Field Name	**Bits**	**Default**	**Description**
TNOPW	1:0	0x0	Extra cycle(s) between successive write bursts. For debugging purpose only.
TNOPR	3:2	0x0	Extra cycle(s) between successive read bursts. For debugging purpose only.
TR2W	8:4	0x9	Read to write turn around time - 1.
TR2R	15:12	0x5	Read to read time - 1 (different rank).
TW2R	20:16	0x9	Write to read turn around time - 1.
TCL	28:24	0x6	CAS to data return latency - 2 (0 to 20).

CAS related paramters in hclk cycles for balanced mode.

MC_SEQ_MISC_TIMING_B - RW - 32 bits - [GpuF0MMReg:0x2624]			
Field Name	**Bits**	**Default**	**Description**
TRP_WRA	5:0	0x15	From write with auto-prechrage to active - 1.
TCKE_HI	7:6	0x0	2 MSB of tCKE parameters, used to control exit power down time.
TRP_RDA	13:8	0x11	From read with auto-prechrage to active - 1.
TRP	19:16	0xb	Precharge command period - 1.
TRFC	26:20	0x2f	Auto-refresh command period - 1.
TCKE	31:28	0x4	CKE power down exit timer.

Misc. DRAM parameters in hclk cycles for balanced mode.

MC_SEQ_MISC_TIMING2_B - RW - 32 bits - [GpuF0MMReg:0x2628]

Field Name	Bits	Default	Description
PA2RDATA	2:0	0x0	Read Preamble for DDR4.
PA2WDATA	6:4	0x0	Write Preamble for DDR4.
FAW	12:8	0x0	
TCKE_PULSE	19:16	0x0	minimum power down period/power up period

Misc. DRAM parameters in hclk cycles for balanced mode.

MC_SEQ_RAS_TIMING_S - RW - 32 bits - [GpuF0MMReg:0x262C]			
Field Name	**Bits**	**Default**	**Description**
TRCDW	4:0	0xa	Number of cycles from active to write - 1.
TRCDWA	9:5	0xa	Number of cycles from active to write with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDW.
TRCDR	14:10	0xd	Number of cycles from active to read - 1.
TRCDRA	19:15	0xd	Number of cycles from active to read with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDR.
TRRD	23:20	0x5	Number of cycles from active bank a to active bank b - 1.
TRC	30:24	0x27	Number of cycles from active to active/auto refresh - 1.

RAS related parameters in hclk cycles for battery mode.

MC_SEQ_CAS_TIMING_S - RW - 32 bits - [GpuF0MMReg:0x2630]			
Field Name	**Bits**	**Default**	**Description**
TNOPW	1:0	0x0	Extra cycle(s) between successive write bursts. For debugging purpose only.
TNOPR	3:2	0x0	Extra cycle(s) between successive read bursts. For debugging purpose only.
TR2W	8:4	0x9	Read to write turn around time - 1.
TR2R	15:12	0x5	Read to read time - 1 (different rank).
TW2R	20:16	0x9	Write to read turn around time - 1.
TCL	28:24	0x6	CAS to data return latency - 2 (0 to 20).

CAS related paramters in hclk cycles for battery mode.

MC_SEQ_MISC_TIMING_S - RW - 32 bits - [GpuF0MMReg:0x2634]			
Field Name	**Bits**	**Default**	**Description**
TRP_WRA	5:0	0x15	From write with auto-prechrage to active - 1.
TCKE_HI	7:6	0x0	2 MSB of tCKE parameters, used to control exit power down time.
TRP_RDA	13:8	0x11	From read with auto-prechrage to active - 1.
TRP	19:16	0xb	Precharge command period - 1.
TRFC	26:20	0x2f	Auto-refresh command period - 1.
TCKE	31:28	0x4	CKE power down exit timer.

Misc. DRAM parameters in hclk cycles for battery mode.

MC_SEQ_MISC_TIMING2_S - RW - 32 bits - [GpuF0MMReg:0x2638]			
Field Name	**Bits**	**Default**	**Description**
PA2RDATA	2:0	0x0	Read Preamble for DDR4.
PA2WDATA	6:4	0x0	Write Preamble for DDR4.
FAW	12:8	0x0	
TCKE_PULSE	19:16	0x0	minimum power down period/power up period

Misc. DRAM parameters in hclk cycles for battery mode.

MC_SEQ_RAS_TIMING_C - RW - 32 bits - [GpuF0MMReg:0x263C]			
Field Name	**Bits**	**Default**	**Description**
TRCDW	4:0	0xa	Number of cycles from active to write - 1.
TRCDWA	9:5	0xa	Number of cycles from active to write with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDW.
TRCDR	14:10	0xd	Number of cycles from active to read - 1.
TRCDRA	19:15	0xd	Number of cycles from active to read with auto-precharge - 1. A special case for DDR1. Otherwise the same as TRCDR.
TRRD	23:20	0x5	Number of cycles from active bank a to active bank b - 1.
TRC	30:24	0x27	Number of cycles from active to active/auto refresh - 1.

RAS related parameters in hclk cycles for context switch mode for context switch mode.

MC_SEQ_CAS_TIMING_C - RW - 32 bits - [GpuF0MMReg:0x2640]			
Field Name	**Bits**	**Default**	**Description**
TNOPW	1:0	0x0	Extra cycle(s) between successive write bursts. For debugging purpose only.
TNOPR	3:2	0x0	Extra cycle(s) between successive read bursts. For debugging purpose only.
TR2W	8:4	0x9	Read to write turn around time - 1.
TR2R	15:12	0x5	Read to read time - 1 (different rank).
TW2R	20:16	0x9	Write to read turn around time - 1.
TCL	28:24	0x6	CAS to data return latency - 2 (0 to 20).

CAS related paramters in hclk cycles for context switch mode.

MC_SEQ_MISC_TIMING_C - RW - 32 bits - [GpuF0MMReg:0x2644]			
Field Name	**Bits**	**Default**	**Description**
TRP_WRA	5:0	0x15	From write with auto-prechrage to active - 1.
TCKE_HI	7:6	0x0	2 MSB of tCKE parameters, used to control exit power down time.
TRP_RDA	13:8	0x11	From read with auto-prechrage to active - 1.

TRP	19:16	0xb	Precharge command period - 1.
TRFC	26:20	0x2f	Auto-refresh command period - 1.
TCKE	31:28	0x4	CKE power down exit timer.

Misc. DRAM parameters in hclk cycles for context switch mode.

MC_SEQ_MISC_TIMING2_C - RW - 32 bits - [GpuF0MMReg:0x2648]			
Field Name	**Bits**	**Default**	**Description**
PA2RDATA	2:0	0x0	Read Preamble for DDR4.
PA2WDATA	6:4	0x0	Write Preamble for DDR4.
FAW	12:8	0x0	
TCKE_PULSE	19:16	0x0	minimum power down period/power up period

Misc. DRAM parameters in hclk cycles for context switch mode.

MC_SEQ_CMD - RW - 32 bits - [GpuF0MMReg:0x26C4]			
Field Name	**Bits**	**Default**	**Description**
ADR	15:0	0x0	This field is mapped directly to the address bus.
MOP	18:16	0x0	DRAM command. 0=NOP 1=Load mode register 2=Precharge 3=Auto-refresh 4=Self-refresh
END	20	0x0	If set, the DLL lock timer starts counting. Once it reaches a pre- defined value, the DLL is stabilized and DRAM power up sequence is completed. See also DLL_CNT inside MC_SEQ_DRAM. 0=Not last operation 1=Last operation, wait for DLL to stabilize
CSB	22:21	0x0	Allows rank 0 and rank 1 to be selected independently. 0=Select both ranks 1=Select rank 1 2=Select rank 0 3=Select none
CHAN0	24	0x0	0=Select channel 0 1=Not select channel 0
CHAN1	25	0x0	0=Select channel 1 1=Not select channel 1

Command register for DRAM initialization.

MC_PMG_CMD - RW - 32 bits - [GpuF0MMReg:0x26CC]			
Field Name	**Bits**	**Default**	**Description**
ADR	15:0	0x0	The value of the mode register for resetting DRAM DLL.

MOP	18:16	0x0	Operation 0=NOP 1=Reset DLL 2=Precharge All 3=Auto-refresh 4=Self-refresh
END	20	0x0	This field is not used. 0=Not last operation 1=Last operation, wait for DLL to stabilize
CSB	22:21	0x0	This field is not used. 0=Select both ranks 1=Select rank 1 2=Select rank 0 3=Select none

Power manager command register. This register specifies the value used for resetting the DRAM DLL.

MC_PMG_CFG - RW - 32 bits - [GpuF0MMReg:0x26D0]			
Field Name	**Bits**	**Default**	**Description**
SYC_CLK	0	0x0	Controls mclk/yclk synchronization after on-chip DLL is reset. 0=Don't synchronize YCLK/MCLK after DLL is reset 1=Synchronize YCLK/MCLK after DLL is reset
RST_DLL	1	0x0	Controls DRAM DLL reset after waking up from self-refresh. 0=Don't reset DRAM DLL after self-refresh 1=Reset DRAM DLL after self-refresh
TRI_MIO	2	0x0	Controls memory IO tristate during power down. 0=Don't tri-state DRAM CMD and CLK signals dring self-refresh 1=tri-state DRAM CMD and CLK signals druing self-refresh
XSR_TMR	7:4	0x0	Multiple of 16 mclk cycles to wait before resetting DRAM DLL.
AUTO_SLF	8	0x0	Enable automatic selfrefresh mode 0=Disable 1=Enable
AUTO_SLF_IDLE_CNT	15:12	0x0	Number of idle cycles memory stays before put the memory into self refresh mode 1=256*2 2=256*3 3=256*4 4=256*5 5=256*6 6=256*7 7=256*8 8=256*9 9=256*10 10=256*11 11=256*12 12=256*13 13=256*14 14=256*15
SLF_IDLE_CNT	19:16	0x0	Number of SEQ idle cycles after SEQ receiving self-refresh command to the time SEQ issue the self-refresh command - 16

WRITE_DURING_DLOCK	20	0x0	0=no write during dll lock time 1=allow write transaction during dll lock time
EARLY_ACK_DYN	21	0x0	0=ack out-of-slf when DLL is locked 1=ack out-of-slf when tXSNR expires
EARLY_ACK_ACPI	22	0x0	0=ack out-of-slf when DLL is locked 1=ack out-of-slf when tXSNR expires
UNUSED_SEQ_SHUTDOWN	24	0x1	0=keep mclk branch running for unused SEQ pair 1=shut off unused SEQ pair

Power manager configuration register.

MC_IMP_CNTL - RW - 32 bits - [GpuF0MMReg:0x26D4]			
Field Name	**Bits**	**Default**	**Description**
MEM_IO_UPDATE_RATE	4:0	0x16	Update the impedance value to the PMTEST every 2^MEM_IO_UPDATE_DELAY cycles
MEM_IO_PMCOMP_STRD2	5	0x0	0=Disable 1=Enable
MEM_IO_SAMPLE_DELAY	12:8	0x6	Calibration Unit will sample every 2^MEM_IO_SAMPLE_DELAY cycles
MEM_IO_SAMPLE_CNT	15:13	0x7	Number of samples to be taken before update value to IO
MEM_IO_INC_THRESHOLD	20:16	0xe	Number of '1' get detected during 15 cycles before increase impedance value
MEM_IO_DEC_THRESHOLD	28:24	0x6	Number of '0' get detected during 15 cycles before decrease impedance value
CAL_WHEN_IDLE	29	0x1	0=Disable 1=Enable
CAL_WHEN_REFRESH	30	0x1	0=Disable 1=Enable
MEM_IMP_EN	31	0x0	0=Disable 1=Enable

Impedance Calibration Control

MC_IMP_DEBUG - RW - 32 bits - [GpuF0MMReg:0x2878]			
Field Name	**Bits**	**Default**	**Description**
MEM_IMP_DEBUG_N	3:0	0x0	
MEM_IMP_DEBUG_P	7:4	0x0	
MEM_IO_IMP_DEBUG_EN	8	0x0	0=Disable 1=Enable
MEM_STATUS_SEL	12	0x0	0=Vertical 1=Horizontal

MC_IMP_STATUS - RW - 32 bits - [GpuF0MMReg:0x2874]			
Field Name	**Bits**	**Default**	**Description**
IMP_N_MEM_DQ_SN_I0 (R)	3:0	0x0	
IMP_P_MEM_DQ_SP_I0 (R)	7:4	0x0	
IMP_N_MEM_DQ_SN_I1 (R)	11:8	0x0	
IMP_P_MEM_DQ_SP_I1 (R)	15:12	0x0	
IMP_N_VALUE_R_BACK (R)	19:16	0x0	
IMP_P_VALUE_R_BACK (R)	23:20	0x0	
IMP_CAL_COUNT (R)	27:24	0x0	

TEST_OUT_R_BACK (R)	28	0x0	
DUMMY_OUT_R_BACK (R)	29	0x0	

MC_IO_PAD_CNTL - RW - 32 bits - [GpuF0MMReg:0x2700]			
Field Name	**Bits**	**Default**	**Description**
VREFI_VCO_EN	0	0x0	
IMP_VREF_INTR	1	0x0	
IMP_VREF_INTN	3:2	0x0	
IMP_VREF_INTP	5:4	0x0	

MC_SEQ_RD_CTL_D0_P - RW - 32 bits - [GpuF0MMReg:0x264C]			
Field Name	**Bits**	**Default**	**Description**
RCV_DLY	2:0	0x1	Delay to turn on receive enable. 0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5
RCV_EXT	7:4	0x1	Extends receive enable signal to cover clock drift. 0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	NPL FIFO pointer reset mode. 0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh

RST_HLD	15:12	0x0	Disables NPL FIFO pointer reset after a read command for a certain period of time. This prevents the pointers (read and write) from resetting before the FIFO is read. 0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	Creates an extra strobe in the preamble of a burst. This is needed if DQS is default high and its falling edge is used as a trigger. 0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	Creates an extra strobe in the postamble of a burst. This is needed if DQS is default high and its rising edge is used as a trigger. 0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	Delay to read data out of a NPL FIFO. This is used to cover the NPL FIFO's write to read latency. 0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

Channel 0's read command parameters in hclk.

MC_SEQ_RD_CTL_D1_P - RW - 32 bits - [GpuF0MMReg:0x2650]			
Field Name	**Bits**	**Default**	**Description**

RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5
RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh
RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

Channel 1's read command parameters in hclk. See MC_SEQ_RD_CTL_I0.

MC_SEQ_WR_CTL_D0_P - RW - 32 bits - [GpuF0MMReg:0x2654]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	Write command to data output latency.
DQS_DLY	7:4	0x3	Write command to DQS latency.
DQS_XTR	8	0x0	Controls write preamble. 0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	Write command to output enable latency.
OEN_EXT	16	0x1	Extends output enable after data burst. 0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	Write command to on-die-termination enable latency.
ODT_EXT	28	0x0	Extends on-die-termination enable after data burst. 0=ODT not extended 1=ODT extended by one cycle

Channel 0's write command parameters in hclk.

MC_SEQ_WR_CTL_D1_P - RW - 32 bits - [GpuF0MMReg:0x2658]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	
DQS_XTR	8	0x0	0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

Channel 1's write command parameters in hclk. See MC_SEQ_WR_CTL_I1.

MC_SEQ_RD_CTL_D0_B - RW - 32 bits - [GpuF0MMReg:0x2694]			
Field Name	**Bits**	**Default**	**Description**
RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5

RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh
RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

MC_SEQ_RD_CTL_D1_B - RW - 32 bits - [GpuF0MMReg:0x2698]			
Field Name	**Bits**	**Default**	**Description**

RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5
RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh
RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

MC_SEQ_WR_CTL_D0_B - RW - 32 bits - [GpuF0MMReg:0x269C]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	
DQS_XTR	8	0x0	0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

MC_SEQ_WR_CTL_D1_B - RW - 32 bits - [GpuF0MMReg:0x26A0]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	
DQS_XTR	8	0x0	0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

MC_SEQ_RD_CTL_D0_S - RW - 32 bits - [GpuF0MMReg:0x26A4]			
Field Name	**Bits**	**Default**	**Description**
RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5
RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on

RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh
RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

MC_SEQ_RD_CTL_D1_S - RW - 32 bits - [GpuF0MMReg:0x26A8]			
Field Name	**Bits**	**Default**	**Description**
RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5

RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh
RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

MC_SEQ_WR_CTL_D0_S - RW - 32 bits - [GpuF0MMReg:0x26AC]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	

DQS_XTR	8	0x0	0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

MC_SEQ_WR_CTL_D1_S - RW - 32 bits - [GpuF0MMReg:0x26B0]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	
DQS_XTR	8	0x0	0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

MC_SEQ_RD_CTL_D0_C - RW - 32 bits - [GpuF0MMReg:0x26B4]			
Field Name	**Bits**	**Default**	**Description**
RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5
RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh

RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

MC_SEQ_RD_CTL_D1_C - RW - 32 bits - [GpuF0MMReg:0x26B8]			
Field Name	**Bits**	**Default**	**Description**
RCV_DLY	2:0	0x1	0=Turn on receive enable at CL-2 1=Turn on receive enable at CL-1 2=Turn on receive enable at CL 3=Turn on receive enable at CL+1 4=Turn on receive enable at CL+2 5=Turn on receive enable at CL+3 6=Turn on receive enable at CL+4 7=Turn on receive enable at CL+5

RCV_EXT	7:4	0x1	0=DQS receive enable not extended 1=DQS receive enable extended by 1 cycle 2=DQS receive enable extended by 2 cycles 3=DQS receive enable extended by 3 cycles 4=DQS receive enable extended by 4 cycles 5=DQS receive enable extended by 5 cycles 6=DQS receive enable extended by 6 cycles 7=DQS receive enable extended by 7 cycles 8=DQS receive enable always on
RST_SEL	9:8	0x2	0=Reset pointers off 1=Reset pointers on 2=Reset pointers before read 3=Reset pointers during refresh
RST_HLD	15:12	0x0	0=Disable reset by 11 cycles 1=Disable reset by 12 cycles 2=Disable reset by 13 cycles 3=Disable reset by 14 cycles 4=Disable reset by 15 cycles 5=Disable reset by 16 cycles 6=Disable reset by 17 cycles 7=Disable reset by 18 cycles 8=Disable reset by 19 cycles 9=Disable reset by 20 cycles 10=Disable reset by 21 cycles 11=Disable reset by 22 cycles 12=Disable reset by 23 cycles 13=Disable reset by 24 cycles 14=Disable reset by 25 cycles
STR_PRE	16	0x0	0=No read pre strobe 1=Extra read pre strobe
STR_PST	17	0x0	0=No read post strobe 1=Extra read post strobe
RBS_DLY	24:20	0x0	0=Assert RBS valid at CL+8 1=Assert RBS valid at CL+9 2=Assert RBS valid at CL+10 3=Assert RBS valid at CL+11 4=Assert RBS valid at CL+12 5=Assert RBS valid at CL+13 6=Assert RBS valid at CL+14 7=Assert RBS valid at CL+15 8=Assert RBS valid at CL+16 9=Assert RBS valid at CL+17 10=Assert RBS valid at CL+18 11=Assert RBS valid at CL+19 12=Assert RBS valid at CL+20 13=Assert RBS valid at CL+21 14=Assert RBS valid at CL+22 15=Assert RBS valid at CL+23 16=Assert RBS valid at CL+24 17=Assert RBS valid at CL+25

MC_SEQ_WR_CTL_D0_C - RW - 32 bits - [GpuF0MMReg:0x26BC]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	
DQS_XTR	8	0x0	0=No write preamble 1=Write preamble

OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

MC_SEQ_WR_CTL_D1_C - RW - 32 bits - [GpuF0MMReg:0x26C0]			
Field Name	**Bits**	**Default**	**Description**
DAT_DLY	3:0	0x3	
DQS_DLY	7:4	0x3	
DQS_XTR	8	0x0	0=No write preamble 1=Write preamble
OEN_DLY	15:12	0x3	
OEN_EXT	16	0x1	0=output enable not extended 1=output eanble extended by one cycle
OEN_SEL	21:20	0x3	
ODT_DLY	27:24	0x0	
ODT_EXT	28	0x0	0=ODT not extended 1=ODT extended by one cycle

MC_SEQ_IO_CTL_D0 - RW - 32 bits - [GpuF0MMReg:0x265C]			
Field Name	**Bits**	**Default**	**Description**
ADR_DLY	0	0x0	Delays address output by half a hclk.
CMD_DLY	1	0x0	Delays command output by half a hclk.
CKN_TRI	4	0x0	Turns off negative clock manually. 0=Normal 1=Tristate
CKP_TRI	5	0x0	Turns off positive clock manually. 0=Normal 1=Tristate
MIO_TRI	6	0x0	Turns off address and command manually. 0=Normal 1=Tristate
CKE_BIT	7	0x0	Bypass value for clock enable.
CKE_SEL	8	0x1	Selects clock enable bypass value. 0=Normal CKE 1=Set CKE bit
STRD2	9	0x0	0=Turn off reserved figures 1=Turn on reserved figures

Channel 0's misc. control parameters.

MC_SEQ_IO_CTL_D1 - RW - 32 bits - [GpuF0MMReg:0x2660]			
Field Name	**Bits**	**Default**	**Description**
ADR_DLY	0	0x0	
CMD_DLY	1	0x0	

CKN_TRI	4	0x0	0=Normal 1=Tristate
CKP_TRI	5	0x0	0=Normal 1=Tristate
MIO_TRI	6	0x0	0=Normal 1=Tristate
CKE_BIT	7	0x0	
CKE_SEL	8	0x1	0=Normal CKE 1=Set CKE bit
STRD2	9	0x0	0=Turn off reserved figures 1=Turn on reserved figures

Channel 1's misc. control parameters. See MC_SEQ_IO_CTL_I0.

MC_SEQ_IO_CTL_UNUSED - RW - 32 bits - [GpuF0MMReg:0x2898]			
Field Name	**Bits**	**Default**	**Description**
CKN_TRI	0	0x1	0=Normal 1=Tristate
CKP_TRI	1	0x1	0=Normal 1=Tristate
MIO_TRI	2	0x1	0=Normal 1=Tristate
DAT_TRI	3	0x1	0=Normal 1=Tristate
STRD2	4	0x0	0=Turn off reserved figures 1=Turn on reserved figures

Unused channel misc. control parameters. This is intended for the second 64-bit IO.

MC_SEQ_NPL_CTL_D0 - RW - 32 bits - [GpuF0MMReg:0x2664]			
Field Name	**Bits**	**Default**	**Description**
LD_INIT	1:0	0x0	NPL FIFO's pointer offset.
SYC_SEL	5:4	0x0	Selects mclk/yclk synchronization mode. 0=mclk/yclk sync off 1=mclk/yclk sync on 2=mclk/yclk sync during refresh 3=periodically turn on mclk/yclk sync
SYC_IDLE_CNT	31:8	0x0	number of cycles a mclk/yclk sync will be forced

Channel 0's NPL control parameters.

MC_SEQ_NPL_CTL_D1 - RW - 32 bits - [GpuF0MMReg:0x2668]			
Field Name	**Bits**	**Default**	**Description**
LD_INIT	1:0	0x0	
SYC_SEL	5:4	0x0	Selects mclk/yclk synchronization mode. The value should be same as MC_SEQ_NPL_CTL_D0 for 32bit mode 0=mclk/yclk sync off 1=mclk/yclk sync on 2=mclk/yclk sync during refresh 3=periodically turn on mclk/yclk sync

SYC_IDLE_CNT	31:8	0x0	number of cycles a mclk/yclk sync will be forced. The value should be same as MC_SEQ_NPL_CTL_D0 for 32bit mode

Channel 1's NPL control parameters. See MC_SEQ_NPL_CTL_I0.

MC_IO_PAD_CNTL_D0 - RW - 32 bits - [GpuF0MMReg:0x27F0]			
Field Name	**Bits**	**Default**	**Description**
DELAY_MASTER_SYNC	1:0	0x0	For 32bit mode, this value should be same as MC_IO_PAD_CNTL_D1
DIFF_STR	2	0x0	0=Strobe single ended 1=Strobe differential
UNI_STR	3	0x0	0=Bidirectional strobes 1=Unidirectional strobes

General Pad control

MC_IO_PAD_CNTL_D1 - RW - 32 bits - [GpuF0MMReg:0x27F4]			
Field Name	**Bits**	**Default**	**Description**
DELAY_MASTER_SYNC	1:0	0x0	
DIFF_STR	2	0x0	0=Strobe single ended 1=Strobe differential
UNI_STR	3	0x0	0=Bidirectional strobes 1=Unidirectional strobes

MC_SEQ_CK_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x266C]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_CK_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2670]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	

PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_CMD_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2674]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_CMD_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2678]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_DQ_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x267C]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_DQ_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2680]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	

USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_QS_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2684]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_QS_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2688]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_A_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x268C]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_A_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2690]

Field Name	Bits	Default	Description
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_IO_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2704]			
Field Name	**Bits**	**Default**	**Description**
DELAY_DATA_SYNC	0	0x0	0=Don't delay data sync 1=delay data sync by 1 yclk
DELAY_STR_SYNC	1	0x0	0=Don't delay strobe sync 1=delay strobe sync by 1 yclk
DELAY_CLK_SYNC	2	0x0	0=Don't delay clk sync 1=delay clk sync by 1 yclk
DELAY_CMD_SYNC	3	0x0	0=Don't delay cmd sync 1=delay cmd sync by 1 yclk
DELAY_ADR_SYNC	4	0x0	0=Don't delay adr sync 1=delay adr sync by 1 yclk
MEM_FALL_OUT_DATA	5	0x0	0=Data out on YCLK rise 1=Data out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_STR	6	0x0	0=Strobe out on YCLK rise 1=Strobe out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CLK	7	0x0	0=Clk out on YCLK rise 1=Clk out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CMD	8	0x0	0=Command out on YCLK rise 1=Command out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_ADR	9	0x0	0=Address out on YCLK rise 1=Address out on YCLK fall, 1/4 clock delay
FORCE_EN_RD_STR	10	0x0	0=Read strb enabled by MC 1=Always enable read strb
EN_RD_STR_DLY	11	0x0	0=count rising edge 1=count falling edge
DISABLE_CMD	12	0x0	0=Drive command 1=Disable command
DISABLE_ADR	13	0x0	0=Drive address 1=Disable address
VREFI_EN	14	0x0	0=VREFI disable 1=VREFI enable
VREFI_SEL	19:15	0x0	
CK_AUTO_EN	20	0x0	0=No CK duty cycle correction 1=Correct CK duty cycle
CK_DELAY_SEL	21	0x0	0=Use register value 1=Use auto cal value
CK_DELAY_N	23:22	0x0	
CK_DELAY_P	25:24	0x0	

MC_IO_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2708]			
Field Name	**Bits**	**Default**	**Description**

DELAY_DATA_SYNC	0	0x0	0=Don't delay data sync 1=delay data sync by 1 yclk
DELAY_STR_SYNC	1	0x0	0=Don't delay strobe sync 1=delay strobe sync by 1 yclk
DELAY_CLK_SYNC	2	0x0	0=Don't delay clk sync 1=delay clk sync by 1 yclk
DELAY_CMD_SYNC	3	0x0	0=Don't delay cmd sync 1=delay cmd sync by 1 yclk
DELAY_ADR_SYNC	4	0x0	0=Don't delay adr sync 1=delay adr sync by 1 yclk
MEM_FALL_OUT_DATA	5	0x0	0=Data out on YCLK rise 1=Data out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_STR	6	0x0	0=Strobe out on YCLK rise 1=Strobe out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CLK	7	0x0	0=Clk out on YCLK rise 1=Clk out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CMD	8	0x0	0=Command out on YCLK rise 1=Command out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_ADR	9	0x0	0=Address out on YCLK rise 1=Address out on YCLK fall, 1/4 clock delay
FORCE_EN_RD_STR	10	0x0	0=Read strb enabled by MC 1=Always enable read strb
EN_RD_STR_DLY	11	0x0	0=count rising edge 1=count falling edge
DISABLE_CMD	12	0x0	0=Drive command 1=Disable command
DISABLE_ADR	13	0x0	0=Drive address 1=Disable address
VREFI_EN	14	0x0	0=VREFI disable 1=VREFI enable
VREFI_SEL	19:15	0x0	
CK_AUTO_EN	20	0x0	0=No CK duty cycle correction 1=Correct CK duty cycle
CK_DELAY_SEL	21	0x0	0=Use register value 1=Use auto cal value
CK_DELAY_N	23:22	0x0	
CK_DELAY_P	25:24	0x0	

MC_IO_RD_DQ_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2710]			
Field Name	**Bits**	**Default**	**Description**
MADJ0	7:0	0x0	
MADJ1	15:8	0x0	
MADJ2	23:16	0x0	
MADJ3	31:24	0x0	

MC_IO_RD_DQ_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2714]			
Field Name	**Bits**	**Default**	**Description**
MADJ0	7:0	0x0	
MADJ1	15:8	0x0	
MADJ2	23:16	0x0	
MADJ3	31:24	0x0	

MC_IO_RD_QS_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2718]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_RD_QS_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x271C]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_RD_QS2_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2720]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_RD_QS2_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2724]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_WR_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2728]			
Field Name	**Bits**	**Default**	**Description**
CK_DLY	2:0	0x0	
CMD_DLY	5:3	0x0	
ADR_DLY	8:6	0x0	

MC_IO_WR_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x272C]			
Field Name	**Bits**	**Default**	**Description**
CK_DLY	2:0	0x0	
CMD_DLY	5:3	0x0	
ADR_DLY	8:6	0x0	

MC_IO_CK_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2730]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	
VREF_INTR	28	0x0	

MC_IO_CK_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2734]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	
VREF_INTR	28	0x0	

MC_IO_CMD_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2738]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	

DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_CMD_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x273C]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_DQ_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2740]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_DQ_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2744]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_QS_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2748]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_QS_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x274C]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_A_PAD_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2750]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_A_PAD_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2754]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_WR_DQ_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2758]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_WR_DQ_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x275C]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_WR_QS_CNTL_D0_I0 - RW - 32 bits - [GpuF0MMReg:0x2760]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_WR_QS_CNTL_D0_I1 - RW - 32 bits - [GpuF0MMReg:0x2764]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	

DLY3	11:9	0x0	

MC_IO_RD_STR_NCNTL_B0_D0 - RW - 32 bits - [GpuF0MMReg:0x26E8]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte0

MC_IO_RD_STR_NCNTL_B1_D0 - RW - 32 bits - [GpuF0MMReg:0x280C]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte1

MC_IO_RD_STR_NCNTL_B2_D0 - RW - 32 bits - [GpuF0MMReg:0x26F8]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte2

MC_IO_RD_STR_NCNTL_B3_D0 - RW - 32 bits - [GpuF0MMReg:0x27F8]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select

SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte3

MC_IO_RD_STR_NCNTL_B4_D0 - RW - 32 bits - [GpuF0MMReg:0x2800]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte4

MC_IO_RD_STR_NCNTL_B5_D0 - RW - 32 bits - [GpuF0MMReg:0x2808]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte5

MC_IO_RD_STR_NCNTL_B6_D0 - RW - 32 bits - [GpuF0MMReg:0x2810]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte6

MC_IO_RD_STR_NCNTL_B7_D0 - RW - 32 bits - [GpuF0MMReg:0x2818]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	Bit0 select
SEL1	5:3	0x0	Bit1 select
SEL2	8:6	0x0	Bit2 select
SEL3	11:9	0x0	Bit3 select
SEL4	14:12	0x0	Bit4 select
SEL5	17:15	0x0	Bit5 select
SEL6	20:18	0x0	Bit6 select
SEL7	23:21	0x0	Bit7 select
SELM	26:24	0x0	

Falling Edge Strobe Select For Read Data Byte7

MC_SEQ_CK_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2768]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_CK_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x276C]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_CMD_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2770]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_CMD_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x2774]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_DQ_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2778]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_DQ_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x277C]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_QS_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2780]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str

LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_QS_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x2784]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_A_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2788]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_SEQ_A_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x278C]			
Field Name	**Bits**	**Default**	**Description**
NMOS_PD	1:0	0x0	
PSTR_OFF_H	7:4	0x0	
NSTR_OFF_H	11:8	0x0	
USE_CAL_STR	12	0x0	0=Ignore cal ctl str 1=Use cal ctl str
LOAD_STR	13	0x0	
PSTR_OFF_V	19:16	0x0	
NSTR_OFF_V	23:20	0x0	

MC_IO_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2790]			
Field Name	**Bits**	**Default**	**Description**

DELAY_DATA_SYNC	0	0x0	0=Don't delay data sync 1=delay data sync by 1 yclk
DELAY_STR_SYNC	1	0x0	0=Don't delay strobe sync 1=delay strobe sync by 1 yclk
DELAY_CLK_SYNC	2	0x0	0=Don't delay clk sync 1=delay clk sync by 1 yclk
DELAY_CMD_SYNC	3	0x0	0=Don't delay cmd sync 1=delay cmd sync by 1 yclk
DELAY_ADR_SYNC	4	0x0	0=Don't delay adr sync 1=delay adr sync by 1 yclk
MEM_FALL_OUT_DATA	5	0x0	0=Data out on YCLK rise 1=Data out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_STR	6	0x0	0=Strobe out on YCLK rise 1=Strobe out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CLK	7	0x0	0=Clk out on YCLK rise 1=Clk out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CMD	8	0x0	0=Command out on YCLK rise 1=Command out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_ADR	9	0x0	0=Address out on YCLK rise 1=Address out on YCLK fall, 1/4 clock delay
FORCE_EN_RD_STR	10	0x0	0=Read strb enabled by MC 1=Always enable read strb
EN_RD_STR_DLY	11	0x0	0=count rising edge 1=count falling edge
DISABLE_CMD	12	0x0	0=Drive command 1=Disable command
DISABLE_ADR	13	0x0	0=Drive address 1=Disable address
VREFI_EN	14	0x0	0=VREFI disable 1=VREFI enable
VREFI_SEL	19:15	0x0	
CK_AUTO_EN	20	0x0	0=No CK duty cycle correction 1=Correct CK duty cycle
CK_DELAY_SEL	21	0x0	0=Use register value 1=Use auto cal value
CK_DELAY_N	23:22	0x0	
CK_DELAY_P	25:24	0x0	

MC_IO_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x2794]			
Field Name	**Bits**	**Default**	**Description**
DELAY_DATA_SYNC	0	0x0	0=Don't delay data sync 1=delay data sync by 1 yclk
DELAY_STR_SYNC	1	0x0	0=Don't delay strobe sync 1=delay strobe sync by 1 yclk
DELAY_CLK_SYNC	2	0x0	0=Don't delay clk sync 1=delay clk sync by 1 yclk
DELAY_CMD_SYNC	3	0x0	0=Don't delay cmd sync 1=delay cmd sync by 1 yclk
DELAY_ADR_SYNC	4	0x0	0=Don't delay adr sync 1=delay adr sync by 1 yclk
MEM_FALL_OUT_DATA	5	0x0	0=Data out on YCLK rise 1=Data out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_STR	6	0x0	0=Strobe out on YCLK rise 1=Strobe out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CLK	7	0x0	0=Clk out on YCLK rise 1=Clk out on YCLK fall, 1/4 clock delay
MEM_FALL_OUT_CMD	8	0x0	0=Command out on YCLK rise 1=Command out on YCLK fall, 1/4 clock delay

MEM_FALL_OUT_ADR	9	0x0	0=Address out on YCLK rise 1=Address out on YCLK fall, 1/4 clock delay
FORCE_EN_RD_STR	10	0x0	0=Read strb enabled by MC 1=Always enable read strb
EN_RD_STR_DLY	11	0x0	0=count rising edge 1=count falling edge
DISABLE_CMD	12	0x0	0=Drive command 1=Disable command
DISABLE_ADR	13	0x0	0=Drive address 1=Disable address
VREFI_EN	14	0x0	0=VREFI disable 1=VREFI enable
VREFI_SEL	19:15	0x0	
CK_AUTO_EN	20	0x0	0=No CK duty cycle correction 1=Correct CK duty cycle
CK_DELAY_SEL	21	0x0	0=Use register value 1=Use auto cal value
CK_DELAY_N	23:22	0x0	
CK_DELAY_P	25:24	0x0	

MC_IO_RD_DQ_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x2798]			
Field Name	**Bits**	**Default**	**Description**
MADJ0	7:0	0x0	
MADJ1	15:8	0x0	
MADJ2	23:16	0x0	
MADJ3	31:24	0x0	

MC_IO_RD_DQ_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x279C]			
Field Name	**Bits**	**Default**	**Description**
MADJ0	7:0	0x0	
MADJ1	15:8	0x0	
MADJ2	23:16	0x0	
MADJ3	31:24	0x0	

MC_IO_RD_QS_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27A0]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_RD_QS_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27A4]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_RD_QS2_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27A8]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_RD_QS2_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27AC]			
Field Name	**Bits**	**Default**	**Description**
DLY0	7:0	0x0	
DLY1	15:8	0x0	
DLY2	23:16	0x0	
DLY3	31:24	0x0	

MC_IO_WR_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27B0]			
Field Name	**Bits**	**Default**	**Description**
CK_DLY	2:0	0x0	
CMD_DLY	5:3	0x0	
ADR_DLY	8:6	0x0	

MC_IO_WR_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27B4]			
Field Name	**Bits**	**Default**	**Description**
CK_DLY	2:0	0x0	
CMD_DLY	5:3	0x0	
ADR_DLY	8:6	0x0	

MC_IO_CK_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27B8]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	
VREF_INTR	28	0x0	

MC_IO_CK_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27BC]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	
VREF_INTR	28	0x0	

MC_IO_CMD_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27C0]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_CMD_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27C4]

Field Name	Bits	Default	Description
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_DQ_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27C8]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_DQ_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27CC]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_QS_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27D0]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	

DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_QS_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27D4]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_A_PAD_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27D8]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	
VREF_INT	27:26	0x0	

MC_IO_A_PAD_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27DC]			
Field Name	**Bits**	**Default**	**Description**
PTERM	3:0	0x0	
NTERM	7:4	0x0	
PDRV	11:8	0x0	
NDRV	15:12	0x0	
RECV_DUTY	17:16	0x0	
DRV_DUTY	19:18	0x0	
PREAMP	21:20	0x0	
SELFTIME	22	0x0	
SLEW	24:23	0x0	
VMODE	25	0x0	

VREF_INT	27:26	0x0	

MC_IO_WR_DQ_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27E0]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_WR_DQ_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27E4]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_WR_QS_CNTL_D1_I0 - RW - 32 bits - [GpuF0MMReg:0x27E8]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_WR_QS_CNTL_D1_I1 - RW - 32 bits - [GpuF0MMReg:0x27EC]			
Field Name	**Bits**	**Default**	**Description**
DLY0	2:0	0x0	
DLY1	5:3	0x0	
DLY2	8:6	0x0	
DLY3	11:9	0x0	

MC_IO_RD_STR_NCNTL_B0_D1 - RW - 32 bits - [GpuF0MMReg:0x2820]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B1_D1 - RW - 32 bits - [GpuF0MMReg:0x2828]			
Field Name	**Bits**	**Default**	**Description**

SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B2_D1 - RW - 32 bits - [GpuF0MMReg:0x2830]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B3_D1 - RW - 32 bits - [GpuF0MMReg:0x2838]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B4_D1 - RW - 32 bits - [GpuF0MMReg:0x2840]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B5_D1 - RW - 32 bits - [GpuF0MMReg:0x2848]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	

SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B6_D1 - RW - 32 bits - [GpuF0MMReg:0x2850]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_IO_RD_STR_NCNTL_B7_D1 - RW - 32 bits - [GpuF0MMReg:0x2858]			
Field Name	**Bits**	**Default**	**Description**
SEL0	2:0	0x0	
SEL1	5:3	0x0	
SEL2	8:6	0x0	
SEL3	11:9	0x0	
SEL4	14:12	0x0	
SEL5	17:15	0x0	
SEL6	20:18	0x0	
SEL7	23:21	0x0	
SELM	26:24	0x0	

MC_SEQ_GENERAL_CONFIG - RW - 32 bits - [GpuF0MMReg:0x26D8]			
Field Name	**Bits**	**Default**	**Description**
MODE_32BIT	0	0x1	0=64-bit channel mode 1=32-bit channel mode
DUAL_IO	1	0x0	0=Single IO configuration 1=Dual IO configuration
MODE_16BIT	4	0x0	1=16-bit channel mode

General SEQ configuration

MC_SEQ_RS_CNTL - RW - 32 bits - [GpuF0MMReg:0x26DC]			
Field Name	**Bits**	**Default**	**Description**
RRDREQ_LCL_CREDIT	3:0	0x4	
XBF_HWM	9:4	0x12	High water mark for mclk to sclk async FIFO, for 64bit BO4, the water mark should be increased
DAT_INV	12	0x0	0=Disable read data inversion 1=Enable read data inversion
MSK_DFI	13	0x1	0=Inverse mask active low 1=Inverse mask active high

RRDREQ_RETURN_PEND	17:16	0x0	0=Return the read data to RS whenever the data is ready 1=Return the read data to RS after all the data for that burst has been received 2=Return the read data to RS when the data is ready and the last read for that burst has been sent out to the memory 3=Reserved
RRDREQ_RS_CREDIT	23:20	0x8	

SEQ to RS control register

MC_SEQ_STATUS_M - RW - 32 bits - [GpuF0MMReg:0x26C8]			
Field Name	**Bits**	**Default**	**Description**
PWRUP_COMPL_D0 (R)	0	0x0	0=CHAN_D0 SDRAM init in progress 1=CHAN_D0 SDRAM ready
PWRUP_COMPL_D1 (R)	1	0x0	0=CHAN_D1 SDRAM init in progress 1=CHAN_D1 SDRAM ready
CMD_RDY_D0 (R)	2	0x0	0=CHAN_D0 Command register busy 1=CHAN_D0 Command register ready
CMD_RDY_D1 (R)	3	0x0	0=CHAN_D1 Command register busy 1=CHAN_D1 Command register ready
SLF_D0 (R)	4	0x0	0=CHAN_D0 Not in Self Refresh mode 1=CHAN_D0 In Self Refresh mode
SLF_D1 (R)	5	0x0	0=CHAN_D1 Not in Self Refresh mode 1=CHAN_D1 In Self Refresh mode
SEQ00_ARB_CMD_FIFO_EMPTY (R)	8	0x0	0=SEQ00 arb interface cmd fifo not empty 1=SEQ00 arb interface cmd fifo empty
SEQ01_ARB_CMD_FIFO_EMPTY (R)	9	0x0	0=SEQ01 arb interface cmd fifo not empty 1=SEQ01 arb interface cmd fifo empty
SEQ10_ARB_CMD_FIFO_EMPTY (R)	10	0x0	0=SEQ10 arb interface cmd fifo not empty 1=SEQ10 arb interface cmd fifo empty
SEQ11_ARB_CMD_FIFO_EMPTY (R)	11	0x0	0=SEQ11 arb interface cmd fifo not empty 1=SEQ11 arb interface cmd fifo empty
SEQ00_RS_DATA_FIFO_FULL (R)	12	0x0	0=SEQ00 rs interface data fifo not full 1=SEQ00 rs interface data fifo full
SEQ01_RS_DATA_FIFO_FULL (R)	13	0x0	0=SEQ01 rs interface data fifo not full 1=SEQ01 rs interface data fifo full
SEQ10_RS_DATA_FIFO_FULL (R)	14	0x0	0=SEQ10 rs interface data fifo not full 1=SEQ10 rs interface data fifo full
SEQ11_RS_DATA_FIFO_FULL (R)	15	0x0	0=SEQ11 rs interface data fifo not full 1=SEQ11 rs interface data fifo full

MC_SEQ_STATUS_S - RW - 32 bits - [GpuF0MMReg:0x288C]			
Field Name	**Bits**	**Default**	**Description**
SEQ00_ARB_DATA_FIFO_FULL (R)	0	0x0	0=SEQ00 arb interface data fifo not full 1=SEQ00 arb interface data fifo full
SEQ01_ARB_DATA_FIFO_FULL (R)	1	0x0	0=SEQ01 arb interface data fifo not full 1=SEQ01 arb interface data fifo full
SEQ10_ARB_DATA_FIFO_FULL (R)	2	0x0	0=SEQ10 arb interface data fifo not full 1=SEQ10 arb interface data fifo full
SEQ11_ARB_DATA_FIFO_FULL (R)	3	0x0	0=SEQ11 arb interface data fifo not full 1=SEQ11 arb interface data fifo full
SEQ00_ARB_CMD_FIFO_FULL (R)	4	0x0	0=SEQ00 arb interface cmd fifo not full 1=SEQ00 arb interface cmd fifo full
SEQ01_ARB_CMD_FIFO_FULL (R)	5	0x0	0=SEQ01 arb interface cmd fifo not full 1=SEQ01 arb interface cmd fifo full
SEQ10_ARB_CMD_FIFO_FULL (R)	6	0x0	0=SEQ10 arb interface cmd fifo not full 1=SEQ10 arb interface cmd fifo full

SEQ11_ARB_CMD_FIFO_FULL (R)	7	0x0	0=SEQ11 arb interface cmd fifo not full 1=SEQ11 arb interface cmd fifo full
SEQ00_RS_DATA_FIFO_EMPTY (R)	8	0x0	0=SEQ00 rs interface data fifo not EMPTY 1=SEQ00 rs interface data fifo EMPTY
SEQ01_RS_DATA_FIFO_EMPTY (R)	9	0x0	0=SEQ01 rs interface data fifo not EMPTY 1=SEQ01 rs interface data fifo EMPTY
SEQ10_RS_DATA_FIFO_EMPTY (R)	10	0x0	0=SEQ10 rs interface data fifo not EMPTY 1=SEQ10 rs interface data fifo EMPTY
SEQ11_RS_DATA_FIFO_EMPTY (R)	11	0x0	0=SEQ11 rs interface data fifo not EMPTY 1=SEQ11 rs interface data fifo EMPTY

MC_NPL_STATUS - RW - 32 bits - [GpuF0MMReg:0x2888]			
Field Name	**Bits**	**Default**	**Description**
D0_I0_PDELAY (R)	1:0	0x0	
D0_I0_NDELAY (R)	3:2	0x0	
D0_I0_PEARLY (R)	4	0x0	
D0_I0_NEARLY (R)	5	0x0	
D0_I1_PDELAY (R)	7:6	0x0	
D0_I1_NDELAY (R)	9:8	0x0	
D0_I1_PEARLY (R)	10	0x0	
D0_I1_NEARLY (R)	11	0x0	
D1_I0_PDELAY (R)	13:12	0x0	
D1_I0_NDELAY (R)	15:14	0x0	
D1_I0_PEARLY (R)	16	0x0	
D1_I0_NEARLY (R)	17	0x0	
D1_I1_PDELAY (R)	19:18	0x0	
D1_I1_NDELAY (R)	21:20	0x0	
D1_I1_PEARLY (R)	22	0x0	
D1_I1_NEARLY (R)	23	0x0	

2.2 Bus Interface Registers

MM_INDEX - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0x0]			
Field Name	**Bits**	**Default**	**Description**
MM_OFFSET	30:0	0x0	This field specifies the offset (in MM space) of the register or the offset in FB memory to be accessed. All accesses must be dword aligned, therefore, bits 1:0 are tied to zero. NOTE: Bits 0:1 of this field are hardwired to ZERO.
MM_APER	31	0x0	This bit specifies whether the address offset is for Register aperture or FB aperture (Linear Aperture). 0=Register Aperture 1=Linear Aperture 0

General Memory Access. The MM_INDEX and MM_DATA pair of registers are used to indirectly accessed all other memory mapped registers in the lower 64KB space and the Frame buffer.

MM_DATA - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0x4]			
Field Name	**Bits**	**Default**	**Description**
MM_DATA	31:0	0x0	This field contains the data to be written to or the data read from the address specified in MM_INDEX.

General Memory Access. The MM_INDEX and MM_DATA pair of registers are used to indirectly access all other BIF memory mapped registers and the frame buffer.

BUS_CNTL - RW - 32 bits - [GpuF0MMReg:0x5420]			
Field Name	**Bits**	**Default**	**Description**
BIOS_ROM_WRT_EN	0	0x0	Unused 0=Disable 1=Enable
BIOS_ROM_DIS	1	0x0	Unused 0=Enable 1=Disable
PMI_IO_DIS	2	0x0	The PMI_STATUS_CNTL.POWER_STATE is used to program the power state. If the power state is D1-D3, then IO access is disabled. If this bit is set to 1, it will enable IO access. 0=Normal 1=Disable
PMI_MEM_DIS	3	0x0	The PMI_STATUS_CNTL.POWER_STATE is used to program the power state. If the power state is D1-D3, then MEM access is disabled. If this bit is set to 1, it will enable MEM access. 0=Normal 1=Disable

PMI_BM_DIS	4	0x0	The PMI_STATUS_CNTL.POWER_STATE is used to program the power state. If the power state is D1-D3, then bus mastering is disabled. If this bit is set to 1, it will enable bus mastering. 0=Normal 1=Disable
PMI_INT_DIS	5	0x0	The PMI_STATUS_CNTL.POWER_STATE is used to program the power state. If the power state is D1-D3, then INTx messages are disabled. If this bit is set to 1, it will enable sending INTx messages. 0=Normal 1=Disable
VGA_REG_COHERENCY_DIS	6	0x0	Disable VGA register coherency. 0=Enable 1=Disable
VGA_MEM_COHERENCY_DIS	7	0x0	Disable VGA memory coherency. 0=Enable 1=Disable
BIF_ERR_RTR_BKPRESSURE_EN	8	0x0	Enable Wrapper backpressure RTR to Gijoe3 when a previous error is pending. When Gijoe3 signals error is done, Wrapper will assert RTR to accept the next request 0=Disable 1=Enable
VGA_COHE_SPEC_TIMER_DIS	9	0x0	0=Enable 1=Disable
ALLOW_TC_TO_PCIE	10	0x0	Allow the traffic class bit from clients to propagate to PCIE core. If not, it will be tied to 0 0=Disable 1=Enable

PCI Express Bus Control Register

CONFIG_CNTL - RW - 32 bits - [GpuF0MMReg:0x5424]			
Field Name	**Bits**	**Default**	**Description**
CFG_VGA_RAM_EN (R)	0	0x0	VGA RAM enable 0=Disable 1=Enable
VGA_DIS	1	0x0	VGA Disable. Unused.
GENMO_MONO_ADDRESS_B (R)	2	0x0	Monochrome emulation or Colour emulation 0=Monochrome emulation, regs at 0x3Bx 1=Color/Graphic emulation, regs at 0x3Dx
GRPH_ADRSEL (R)	4:3	0x0	Graphics address and aperture size select 0=A0000-128K 1=A0000-64K 2=B0000-32K 3=B8000-32K

Configuration Control Register

CONFIG_MEMSIZE - RW - 32 bits - [GpuF0MMReg:0x5428]			
Field Name	**Bits**	**Default**	**Description**

CONFIG_MEMSIZE	31:0	0x0	Configuration memory size NOTE: Bits 0:19 of this field are hardwired to ZERO.

Scratch regsiter for BIOS to inform driver memory size

CONFIG_F0_BASE - R - 32 bits - [GpuF0MMReg:0x542C]			
Field Name	**Bits**	**Default**	**Description**
F0_BASE	31:0	0x0	F0 Base Address NOTE: Bits 0:24 of this field are hardwired to ZERO.

Configuration F0 Base Register

CONFIG_APER_SIZE - R - 32 bits - [GpuF0MMReg:0x5430]			
Field Name	**Bits**	**Default**	**Description**
APER_SIZE	31:0	0x0	Strap-loadable register based on strap MEM_AP_SIZE NOTE: Bits 0:23 of this field are hardwired to ZERO.

Function 0 Configuration Memory Aperture Size

CONFIG_REG_APER_SIZE - R - 32 bits - [GpuF0MMReg:0x5434]			
Field Name	**Bits**	**Default**	**Description**
REG_APER_SIZE	19:0	0x0	Strap-loadable register based on strap REG_AP_SIZE

Function 0 Configuration Register Aperture Size

2.3 PCI-E Registers

PCIE_INDEX - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0x30]			
Field Name	**Bits**	**Default**	**Description**
PCIE_INDEX	7:0	0x0	index of bifdec
Index register for the PCI Express common indirect registers			

PCIE_DATA - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0x34]			
Field Name	**Bits**	**Default**	**Description**
PCIE_DATA	31:0	0x0	data of bifdec
Data register for the PCI Express common indirect registers			

PCIE_RX_NUM_NACK - R - 32 bits - PCIEIND:0xE			
Field Name	**Bits**	**Default**	**Description**
RX_NUM_NACK	31:0	0x0	Total number of nacks received
Num nacks received			

PCIE_RX_NUM_NACK_GENERATED - R - 32 bits - PCIEIND:0xF			
Field Name	**Bits**	**Default**	**Description**
RX_NUM_NACK_GENERATED	31:0	0x0	Total number of nacks generated
Num nacks generated			

PCIE_CI_CNTL - RW - 32 bits - PCIEIND:0x20			
Field Name	**Bits**	**Default**	**Description**
CI_BE_SPLIT_MODE	1:0	0x0	0=Normal byte splitting rules for PCI-Express 1.0A 1=Force a split on QW boundary with maximum packet length = 2 2=Bypass mode that forces full byte enables
CI_SLAVE_SPLIT_MODE	2	0x0	Completions split on Channels 0=RC - Full completions from Channel A or B 1=RC - Completions split on Channel A and B evenly
CI_SLAVE_GEN_USR_DIS	3	0x0	Sends USR for invalid addresses 0=Sends USR for invalid addresses 1=Disables slave from sending USR, and instead sends a successful CMPLT_D with dummy data.
CI_MST_CMPL_DUMMY_DATA	4	0x1	0xDEADBEEF or 0xFFFFFFFF 0=0xDEADBEEF 1=0xFFFFFFFF
CI_MST_TAG_MODE	5	0x0	incremental tag or first available tag 0=incremental tag 1=first available tag

CI_SLV_RC_RD_REQ_SIZE	7:6	0x1	Slave read requests supported size to client. 0=32/64 byte requests supported 1=64 byte requests only 2=16/32/64
CI_SLV_ORDERING_DIS	8	0x0	Disable slave ordering logic 0=Enable slave ordering logic 1=Disable slave ordering logic
CI_RC_ORDERING_DIS	9	0x0	Disable RC ordering logic 0=Enable RC ordering logic 1=Disable RC ordering logic
CI_SLV_CPL_ALLOC_DIS	10	0x0	Slave CPL buffer is sub-divided or not 0=Slave CPL buffer is sub-divided between ports based on number of lanes active 1=Slave CPL buffer is not sub-divided
CI_SLV_CPL_ALLOC_MODE (R)	11	0x0	Slave Cpl buffer method for sub-division. 0 - dynamic, 1 - register limits CI_SLV_CPL_STATIC_ALLOC_LIMIT_(N)S

chip interface control register

PCIE_LC_STATE6 - R - 32 bits - PCIEIND:0x22			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE24	5:0	0x0	24th previous state
LC_PREV_STATE25	13:8	0x0	25th previous state
LC_PREV_STATE26	21:16	0x0	26th previous state
LC_PREV_STATE27	29:24	0x0	27th previous state

Link Control State Registers

PCIE_LC_STATE7 - R - 32 bits - PCIEIND:0x23			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE28	5:0	0x0	28th previous state
LC_PREV_STATE29	13:8	0x0	29th previous state
LC_PREV_STATE30	21:16	0x0	30th previous state
LC_PREV_STATE31	29:24	0x0	31st previous state

Link Control State Registers

PCIE_LC_STATE8 - R - 32 bits - PCIEIND:0x24			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE32	5:0	0x0	32nd previous state
LC_PREV_STATE33	13:8	0x0	33rd previous state
LC_PREV_STATE34	21:16	0x0	34th previous state
LC_PREV_STATE35	29:24	0x0	35th previous state

Link Control State Registers

PCIE_LC_STATE9 - R - 32 bits - PCIEIND:0x25			
Field Name	**Bits**	**Default**	**Description**

LC_PREV_STATE36	5:0	0x0	36th previous state
LC_PREV_STATE37	13:8	0x0	37th previous state
LC_PREV_STATE38	21:16	0x0	38th previous state
LC_PREV_STATE39	29:24	0x0	39th previous state

Link Control State Registers

PCIE_LC_STATE10 - R - 32 bits - PCIEIND:0x26			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE40	5:0	0x0	40th previous state
LC_PREV_STATE41	13:8	0x0	41st previous state
LC_PREV_STATE42	21:16	0x0	42nd previous state
LC_PREV_STATE43	29:24	0x0	43rd previous state

Link Control State Registers

PCIE_LC_STATE11 - R - 32 bits - PCIEIND:0x27			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE44	5:0	0x0	44th previous state
LC_PREV_STATE45	13:8	0x0	45th previous state
LC_PREV_STATE46	21:16	0x0	46th previous state
LC_PREV_STATE47	29:24	0x0	47th previous state

Link Control State Registers

PCIE_P_CNTL - RW - 32 bits - PCIEIND:0x40			
Field Name	**Bits**	**Default**	**Description**
P_PWRDN_EN	0	0x0	Enable powering down transmitter and receiver pads along with PLL macros
P_SYMALIGN_MODE	1	0x0	Data Valid generation bit - iMODE = 0 (Relax Mode): update its symbol right away when detect any bit shift, i.e. data_valid will always assert. iMODE = 1 (Aggressive Mode): need confirmation before muxing out the data
P_PLL_PWRDN_IN_L1L23	3	0x0	Enable PLL powerdown in L1 or L23 Ready states - only if all the associated LC's are in Sates L1 / L23 corresponding to 4 / 2 lanes based on mpConfig and architecture
P_PLL_BUF_PDNB	4	0x1	Disable 10X clock pad on a per PLL basis - should be 1'b0 in order to activate this powersafe feature. 0=Enable PLL Buffer to power down during L1 1=Always keep PLL Buffer running
P_TXCLK_SND_PWRDN	5	0x0	Enable powering down TXCLK clock pads on the transmit side. Each clock pad corresponds to logic associated with 4 lanes.
P_TXCLK_RCV_PWRDN	6	0x0	Enable powering down TXCLK clock pads on the receive side. Each clock pad corresponds to logic associated with 4 lanes.

PI_SYMALIGN_DIS_ELIDLE	7	0x0	Symbol Alignment Statemachine control signal: iDIS_ELIDLE = 0, ElectIdle assertion will be effective in state machine re-initialization. iDIS_ELIDLE = 1, ElectIdle will be ineffective in state machine re-initialization
P_MASK_RCVR_EIDLE_EN	8	0x0	Enable EIDLE mask for powered down receivers. 0=dont intercept ELEC_IDLE in power down 1=intercept ELEC_IDLE in RX power down
P_PLL_PDNB	9	0x1	Enable PLL only (not the buffer) to power down in L1 or L23ready states. 0=Enable PLL to power down during L1 1=Always keep PLL running
P_EBUF_SYNC_MODE	10	0x0	0=double flops 1=single flop
P_LDSK_MASK_RCVR_ELEC_IDLE	11	0x0	0=GEN1:not mask-off GEN2: mask-off 1=mask-off for GEN1 and GEN2
P_ALLOW_PRX_FRONTEND_SHUTOFF	12	0x0	Enable PHY's RX FRONTEND to shut off during L1 when PLL power down is enabled. 0=RX Frontend is always power on 1=RX Frontend is shutoff during L1 when PLL power down is enabled
P_ALWAYS_USE_FAST_TXCLK	13	0x0	Bypass TXCLK_SWITCH and use 500MHz TXCLK from PLL for both GEN1 and GEN2 speed. 0=TXCLK will be either 250MHz or 500MHz depends on port speeds 1=Bypass TXCLK_SWITCH and always use 500MHz TXCLK
P_ELEC_IDLE_MODE	15:14	0x0	Electrical Idle Mode for PI (Physical Layer). 0=GEN1 - entry:PHY, exit:PHY ; GEN2 - entry:infer, exit:PHY 1=GEN1 - entry:infer, exit:PHY; GEN2 - entry:infer, exit PHY 2=GEN1 - entry:PHY, exit:PHY ; GEN2 - entry:PHY, exit:PHY 3=Reserved
RXP_XBAR_MUX0	17:16	0x0	Data routing cross bar mux - default 1'b0
RXP_XBAR_MUX1	19:18	0x1	Data routing cross bar mux - default 1'b1
RXP_XBAR_MUX2	21:20	0x2	Data routing cross bar mux - default 1'b2
RXP_XBAR_MUX3	23:22	0x3	Data routing cross bar mux - default 1'b3
PI_RXEN_GATER	27:24	0x2	
RXP_REALIGN_ON_EACH_TSX_OR_SKP	28	0x1	0=LDSK only taking deskew on deskewing error detect 1=taking deskew on every TSX and SKP OS
LC_RXP_DONT_ALIGN_ON_TSx	29	0x1	Control Lane Deskew TS detection in L1 and L23 0=Don't mask out TS ordered sets during L1 and L23. 1=Mask out lane deskew TSx detection during L1 and L23.

PHY Control Register

PCIE_P_BUF_STATUS - RW - 32 bits - PCIEIND:0x41			
Field Name	**Bits**	**Default**	**Description**
P_ELASTIC_BUF_OVERFLOW_0	0	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 0
P_ELASTIC_BUF_OVERFLOW_1	1	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 1
P_ELASTIC_BUF_OVERFLOW_2	2	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 2

P_ELASTIC_BUF_OVERFLOW_3	3	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 3
P_ELASTIC_BUF_OVERFLOW_4	4	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 4
P_ELASTIC_BUF_OVERFLOW_5	5	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 5
P_ELASTIC_BUF_OVERFLOW_6	6	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 6
P_ELASTIC_BUF_OVERFLOW_7	7	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 7
P_ELASTIC_BUF_OVERFLOW_8	8	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 8
P_ELASTIC_BUF_OVERFLOW_9	9	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 9
P_ELASTIC_BUF_OVERFLOW_10	10	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 10
P_ELASTIC_BUF_OVERFLOW_11	11	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 11
P_ELASTIC_BUF_OVERFLOW_12	12	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 12
P_ELASTIC_BUF_OVERFLOW_13	13	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 13
P_ELASTIC_BUF_OVERFLOW_14	14	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 14
P_ELASTIC_BUF_OVERFLOW_15	15	0x0	Rx to Tx time domain hand-off buffer under/over flow: lane 15
P_DESKEW_BUF_OVERFLOW_0	16	0x0	Symbol skew buffer over/underflow: lane 0
P_DESKEW_BUF_OVERFLOW_1	17	0x0	Symbol skew buffer over/underflow: lane 1
P_DESKEW_BUF_OVERFLOW_2	18	0x0	Symbol skew buffer over/underflow: lane 2
P_DESKEW_BUF_OVERFLOW_3	19	0x0	Symbol skew buffer over/underflow: lane 3
P_DESKEW_BUF_OVERFLOW_4	20	0x0	Symbol skew buffer over/underflow: lane 4
P_DESKEW_BUF_OVERFLOW_5	21	0x0	Symbol skew buffer over/underflow: lane 5
P_DESKEW_BUF_OVERFLOW_6	22	0x0	Symbol skew buffer over/underflow: lane 6
P_DESKEW_BUF_OVERFLOW_7	23	0x0	Symbol skew buffer over/underflow: lane 7
P_DESKEW_BUF_OVERFLOW_8	24	0x0	Symbol skew buffer over/underflow: lane 8
P_DESKEW_BUF_OVERFLOW_9	25	0x0	Symbol skew buffer over/underflow: lane 9
P_DESKEW_BUF_OVERFLOW_10	26	0x0	Symbol skew buffer over/underflow: lane 10
P_DESKEW_BUF_OVERFLOW_11	27	0x0	Symbol skew buffer over/underflow: lane 11
P_DESKEW_BUF_OVERFLOW_12	28	0x0	Symbol skew buffer over/underflow: lane 12
P_DESKEW_BUF_OVERFLOW_13	29	0x0	Symbol skew buffer over/underflow: lane 13
P_DESKEW_BUF_OVERFLOW_14	30	0x0	Symbol skew buffer over/underflow: lane 14
P_DESKEW_BUF_OVERFLOW_15	31	0x0	Symbol skew buffer over/underflow: lane 15

PHY BUFFER STATUS REGISTER

PCIE_P_DECODER_STATUS - RW - 32 bits - PCIEIND:0x42			
Field Name	**Bits**	**Default**	**Description**
P_DECODE_ERR_0	0	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_1	1	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_2	2	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_3	3	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc

P_DECODE_ERR_4	4	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_5	5	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_6	6	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_7	7	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_8	8	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_9	9	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_10	10	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_11	11	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_12	12	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_13	13	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_14	14	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DECODE_ERR_15	15	0x0	Indicates which lane has the decoding error, i.e. Can't decode the incoming data. bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_0	16	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_1	17	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_2	18	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_3	19	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_4	20	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_5	21	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_6	22	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_7	23	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_8	24	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_9	25	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_10	26	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_11	27	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_12	28	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc

P_DISPARITY_ERR_13	29	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_14	30	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc
P_DISPARITY_ERR_15	31	0x0	Indicates which lane has the link error: bit15 => Lane 15 (0 = OK, 1 = error), etc

PHY DECODER STATUS REGISTER

PCIE_P_MISC_DEBUG_STATUS - RW - 32 bits - PCIEIND:0x43			
Field Name	**Bits**	**Default**	**Description**
P_LANE_REVERSAL (R)	2	0x0	Lane Reversal 0=All lane order is normal 1=All lane order is reversed
P_HW_DEBUG	15:4	0x0	
P_INSERT_ERROR_0	16	0x0	Transmit invalid symbol 10'b0001111001 on lane 0 0=Normal Operation 1=Inserting error on Transmitting Lane0 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_1	17	0x0	Transmit invalid symbol 10'b0001111001 on lane 1 0=Normal Operation 1=Inserting error on Transmitting Lane1 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_2	18	0x0	Transmit invalid symbol 10'b0001111001 on lane 2 0=Normal Operation 1=Inserting error on Transmitting Lane2 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_3	19	0x0	Transmit invalid symbol 10'b0001111001 on lane 3 0=Normal Operation 1=Inserting error on Transmitting Lane3 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_4	20	0x0	Transmit invalid symbol 10'b0001111001 on lane 4 0=Normal Operation 1=Inserting error on Transmitting Lane4 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_5	21	0x0	Transmit invalid symbol 10'b0001111001 on lane 5 0=Normal Operation 1=Inserting error on Transmitting Lane5 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_6	22	0x0	Transmit invalid symbol 10'b0001111001 on lane 6 0=Normal Operation 1=Inserting error on Transmitting Lane6 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_7	23	0x0	Transmit invalid symbol 10'b0001111001 on lane 7 0=Normal Operation 1=Inserting error on Transmitting Lane7 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_8	24	0x0	Transmit invalid symbol 10'b0001111001 on lane 8 0=Normal Operation 1=Inserting error on Transmitting Lane8 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_9	25	0x0	Transmit invalid symbol 10'b0001111001 on lane 9 0=Normal Operation 1=Inserting error on Transmitting Lane9 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_10	26	0x0	Transmit invalid symbol 10'b0001111001 on lane 10 0=Normal Operation 1=Inserting error on Transmitting Lane10 by replacing one symbol with an invalid symbol

P_INSERT_ERROR_11	27	0x0	Transmit invalid symbol 10'b0001111001 on lane 11 0=Normal Operation 1=Inserting error on Transmitting Lane11 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_12	28	0x0	Transmit invalid symbol 10'b0001111001 on lane 12 0=Normal Operation 1=Inserting error on Transmitting Lane12 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_13	29	0x0	Transmit invalid symbol 10'b0001111001 on lane 13 0=Normal Operation 1=Inserting error on Transmitting Lane13 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_14	30	0x0	Transmit invalid symbol 10'b0001111001 on lane 14 0=Normal Operation 1=Inserting error on Transmitting Lane14 by replacing one symbol with an invalid symbol
P_INSERT_ERROR_15	31	0x0	Transmit invalid symbol 10'b0001111001 on lane 15 0=Normal Operation 1=Inserting error on Transmitting Lane15 by replacing one symbol with an invalid symbol

PHY MISCELLANEOUS DEBUG STATUS REGISTER

PCIE_P_SYMSYNC_CTL - RW - 32 bits - PCIEIND:0x46			
Field Name	**Bits**	**Default**	**Description**
P_SYMSYNC_ELECT_IDLE_DET_EN	0	0x1	Use Electrical Idle Detect to filter out garbage data
P_SYMSYNC_SYNC_MODE	1	0x0	SYMSYNC synchronous mode - 1 look for iMGood consecutive good COMMAs, 0 look for iMGood consecutive good symbols
P_SYMSYNC_M_GOOD	9:2	0x7	M parameter of Good symbols or Commas (should be greater than two)
P_SYMSYNC_N_BAD	17:10	0x1	N parameter of Bad symbols (can be 1 or more)
P_SYMSYNC_PAD_MODE	19:18	0x3	Mode select of Good known symbols for replacement of the Bad symbols
P_SYMSYNC_BYPASS_MODE	20	0x1	Bypass mode - 1 just let data and DValid flow through 0=Bypass Symsync and Disable Symsync 1=Enable Symsync
P_SYMSYNC_ENABLE_IN_GEN1	21	0x0	Enable Symsync for GEN1 0=SYMSYNC is enabled for GEN2 only 1=Enable Symsync for GEN1 as well

SYMSYNC Control Registers

PCIE_P_IMP_CNTL_STRENGTH - RW - 32 bits - PCIEIND:0x60			
Field Name	**Bits**	**Default**	**Description**
P_TX_STR_CNTL_READ_BACK (R)	3:0	0x0	Store the readback value of current controller
P_TX_IMP_CNTL_READ_BACK (R)	7:4	0x0	Store the readback value of TX impedance controller
P_RX_IMP_CNTL_READ_BACK (R)	11:8	0x0	Store the readback value of RX impedance controller
P_TX_STR_CNTL	19:16	0x7	Set the initial default current strength to 4'b0111
P_TX_IMP_CNTL	23:20	0x6	Default TX impedance control value
P_RX_IMP_CNTL	27:24	0x6	Default RX impedance control value
PI_HALT_IMP_CAL	28	0x0	
P_PAD_MANUAL_OVERRIDE	31	0x0	Enable Current and Impedance control values to override 0=Allow normal impedance compensation operation 1=Default to manual settings

PHY IMPEDANCE CONTROL STRENGTH REGISTER

PCIE_P_IMP_CNTL_UPDATE - RW - 32 bits - PCIEIND:0x61			
Field Name	**Bits**	**Default**	**Description**
P_IMP_PAD_UPDATE_RATE	4:0	0xe	PAD's update interval 0=PHY130 default 0xf 1=PHY90 default 0xe
P_IMP_PAD_SAMPLE_DELAY	12:8	0x1	Sampling window
P_IMP_PAD_INC_THRESHOLD	20:16	0x18	Incremental resolution
P_IMP_PAD_DEC_THRESHOLD	28:24	0x8	Decremental resolution

Impedance PAD defaults

PCIE_P_STR_CNTL_UPDATE - RW - 32 bits - PCIEIND:0x62			
Field Name	**Bits**	**Default**	**Description**
P_STR_PAD_UPDATE_RATE	4:0	0xf	PAD's update interval 0=PHY130 default 0xf 1=PHY90 default 0xe
P_STR_PAD_SAMPLE_DELAY	12:8	0x1	Sampling window
P_STR_PAD_INC_THRESHOLD	20:16	0x18	Incremental resolution
P_STR_PAD_DEC_THRESHOLD	28:24	0x8	Decremental resolution

Current PAD defaults

PCIE_P_PAD_MISC_CNTL - RW - 32 bits - PCIEIND:0x63			
Field Name	**Bits**	**Default**	**Description**
P_PAD_I_DUMMYOUT (R)	0	0x0	Input from analog - 0 if PMOS cur is stronger
P_PAD_IMP_DUMMYOUT (R)	1	0x0	Input from analog - 0 if PMOS imp is stronger
P_PAD_IMP_TESTOUT (R)	2	0x0	Input from analog - 1 if NMOS imp is stronger
P_LINK_RETRAIN_ON_ERR_EN	3	0x0	Disable error counts in LaneDeskew if Symbol unlocking, Code Errors or Deskew Errors are detected
P_PLLCAL_INC_LOWER_PHASE	6:4	0x1	0=0us 1=1us 2=2us 3=4us 4=8us 5=12us 6=16us 7=24us

Pad Miscellaneous Control Registers

PCIE_P_DECODE_ERR_CNTL - RW - 32 bits - PCIEIND:0xEF			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_RESET	15:0	0x0	
DISPARITY_ERR_CNT_RESET	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_0 - R - 32 bits - PCIEIND:0xF0			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_0	15:0	0x0	
DISPARITY_ERR_CNT_0	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_1 - R - 32 bits - PCIEIND:0xF1			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_1	15:0	0x0	
DISPARITY_ERR_CNT_1	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_2 - R - 32 bits - PCIEIND:0xF2			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_2	15:0	0x0	
DISPARITY_ERR_CNT_2	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_3 - R - 32 bits - PCIEIND:0xF3			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_3	15:0	0x0	
DISPARITY_ERR_CNT_3	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_4 - R - 32 bits - PCIEIND:0xF4			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_4	15:0	0x0	
DISPARITY_ERR_CNT_4	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_5 - R - 32 bits - PCIEIND:0xF5			
Field Name	**Bits**	**Default**	**Description**

CODE_ERR_CNT_5	15:0	0x0	
DISPARITY_ERR_CNT_5	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_6 - R - 32 bits - PCIEIND:0xF6			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_6	15:0	0x0	
DISPARITY_ERR_CNT_6	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_7 - R - 32 bits - PCIEIND:0xF7			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_7	15:0	0x0	
DISPARITY_ERR_CNT_7	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_8 - R - 32 bits - PCIEIND:0xF8			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_8	15:0	0x0	
DISPARITY_ERR_CNT_8	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_9 - R - 32 bits - PCIEIND:0xF9			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_9	15:0	0x0	
DISPARITY_ERR_CNT_9	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_10 - R - 32 bits - PCIEIND:0xFA			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_10	15:0	0x0	
DISPARITY_ERR_CNT_10	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_11 - R - 32 bits - PCIEIND:0xFB

Field Name	Bits	Default	Description
CODE_ERR_CNT_11	15:0	0x0	
DISPARITY_ERR_CNT_11	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_12 - R - 32 bits - PCIEIND:0xFC			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_12	15:0	0x0	
DISPARITY_ERR_CNT_12	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_13 - R - 32 bits - PCIEIND:0xFD			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_13	15:0	0x0	
DISPARITY_ERR_CNT_13	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_14 - R - 32 bits - PCIEIND:0xFE			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_14	15:0	0x0	
DISPARITY_ERR_CNT_14	31:16	0x0	

PCIE_P_DECODE_ERR_CNT_15 - R - 32 bits - PCIEIND:0xFF			
Field Name	**Bits**	**Default**	**Description**
CODE_ERR_CNT_15	15:0	0x0	
DISPARITY_ERR_CNT_15	31:16	0x0	

PCIE_TX_CNTL - RW - 32 bits - PCIEIND_P:0x20			
Field Name	**Bits**	**Default**	**Description**
TX_REPLAY_NUM_COUNT (R)	9:0	0x0	TX Replay Number Counter - counter to keep track of the number of replays that have occured
TX_SNR_OVERRIDE	11:10	0x0	Snoop Not Required Override - control of the Snoop bit for master requests 0=Generate bit as normal 1=Override equation, and always set bit 2=Override equation, and always clear bit 3=Invalid

TX_RO_OVERRIDE	13:12	0x0	Relaxed Ordering Override - control relaxed ordering bit for master requests 0=Generate bit as normal 1=Override equation, and always set bit 2=Override equation, and always clear bit 3=Invalid
TX_PACK_PACKET_DIS	14	0x0	Packet Packing Disable - back-to-back packing of TLP and DLLP 0=Place packets as close as allowable 1=Place STP/SDP in lane 0 only
TX_GENERATE_CRC_ERR	15	0x0	Generate CRC errors from TX by zeroing CRC field. 0=Generate proper CRC 1=Generate bad CRC
TX_GAP_BTW_PKTS	18:16	0x0	Number of idle cycles between DLLP and TLP
TX_FLUSH_TLP_DIS	19	0x1	Disable flushing TLPs when Data Link is down 0=Normal 1=Disable
TX_CPL_PASS_P	20	0x0	Ordering rule: Let Completion Pass Posted 0=no pass 1=CPL pass
TX_NP_PASS_P	21	0x0	Ordering rule: Let Non-Posted Pass Posted 0=no pass 1=NP pass
TX_FC_UPDATE_TIMEOUT_SEL	25:24	0x2	To adjust the length of the timeout interval before sending out flow control update 0=Disable flow control 1=4x clock cycle 2=1024x clock cycle 3=4096x clock cycle
TX_FC_UPDATE_TIMEOUT	31:26	0x7	Interval length to send flow control update

TX Control Register

PCIE_TX_SEQ - R - 32 bits - PCIEIND_P:0x24			
Field Name	**Bits**	**Default**	**Description**
TX_NEXT_TRANSMIT_SEQ	11:0	0x0	Next Transmit Sequence Number to send out
TX_ACKD_SEQ	27:16	0x0	Last Acknowledged Sequence Number

TX Sequence Register

PCIE_TX_REPLAY - RW - 32 bits - PCIEIND_P:0x25			
Field Name	**Bits**	**Default**	**Description**
TX_REPLAY_NUM	9:0	0x3	Register to control Replay Number before Link goes to Retrain
TX_REPLAY_TIMER_OVERWRITE	15	0x0	Trigger for Replay Timer
TX_REPLAY_TIMER	31:16	0x90	Replay Timer - when expired do Replay

TX Replay Register

PCIE_ERR_CNTL - RW - 32 bits - PCIEIND_P:0x6A			
Field Name	**Bits**	**Default**	**Description**
ERR_REPORTING_DIS	0	0x0	Disable PCI Express Advanced Error Reporting

ERR_GEN_INTERRUPT	1	0x0	Enable Interrupt Generation for errors
SYM_UNLOCKED_EN	2	0x0	Enable Reporting of Symbol Unlocked Errors 0=disable reporting unlocked symbol errors 1=report unlocked symbol errors

Error Control Registers

PCIE_RX_CNTL - RW - 32 bits - PCIEIND_P:0x70			
Field Name	**Bits**	**Default**	**Description**
RX_IGNORE_IO_ERR	0	0x0	Ignore Malformed I/O TLP Errors
RX_IGNORE_BE_ERR	1	0x0	Ignore Malformed Byte Enable TLP Errors
RX_IGNORE_MSG_ERR	2	0x0	Ignore Malformed Message Error
RX_IGNORE_CRC_ERR (R)	3	0x0	Ignore CRC Errors
RX_IGNORE_CFG_ERR	4	0x0	Ignore Malformed Configuration Errors
RX_IGNORE_CPL_ERR	5	0x0	Ignore Malformed Completion Errors
RX_IGNORE_EP_ERR	6	0x0	Ignore Malformed EP Errors
RX_IGNORE_LEN_MISMATCH_ERR	7	0x0	Ignore Malformed Length Mismatch Errors
RX_IGNORE_MAX_PAYLOAD_ERR	8	0x0	Ignore Malformed Maximum Payload Errors
RX_IGNORE_TC_ERR	9	0x0	Ignore Malformed Traffic Class Errors
RX_IGNORE_CFG_UR	10	0x0	RESERVED
RX_IGNORE_IO_UR	11	0x0	RESERVED
RX_IGNORE_VEND0_UR	12	0x0	Ignore Vendor Type 0 Messages
RX_NAK_IF_FIFO_FULL	13	0x0	Send NAK if RX internal FIFO is full
RX_GEN_ONE_NAK	14	0x1	Generate NAK only for the first bad packet until replayed
RX_FC_INIT_FROM_REG	15	0x0	Flow Control Initialization from registers 0=Init FC from FIFO sizes 1=Init FC from registers
RX_RCB_CPL_TIMEOUT	18:16	0x0	RCB cpl timeout 0=Disable 1=50us 2=2.5ms 3=6.25ms 4=12.5ms 5=25ms 6=125ms 7=0.25ms
RX_RCB_CPL_TIMEOUT_MODE	19	0x0	RCB cpl timeout on link down
RX_PCIE_CPL_TIMEOUT_DIS	20	0x0	

RX Control Register

PCIE_RX_CREDITS_ALLOCATED_P - R - 32 bits - PCIEIND_P:0x80			
Field Name	**Bits**	**Default**	**Description**
RX_CREDITS_ALLOCATED_PD	11:0	0x0	For posted TLP data, the number of FC units granted to transmitter since initialization, modulo 4096
RX_CREDITS_ALLOCATED_PH	23:16	0x0	For posted TLP header, the number of FC units granted to transmitter since initialization, modulo 256

RX Credits Allocated Register (Posted)

PCIE_RX_CREDITS_ALLOCATED_NP - R - 32 bits - PCIEIND_P:0x81			
Field Name	**Bits**	**Default**	**Description**

RX_CREDITS_ALLOCATED_NPD	11:0	0x0	For non-posted TLP data, the number of FC units granted to transmitter since initialization, modulo 4096
RX_CREDITS_ALLOCATED_NPH	23:16	0x0	For non-posted TLP header, the number of FC units granted to transmitter since initialization, modulo 256

RX Credits Allocated Register (Non-Posted)

PCIE_RX_CREDITS_ALLOCATED_CPL - R - 32 bits - PCIEIND_P:0x82			
Field Name	**Bits**	**Default**	**Description**
RX_CREDITS_ALLOCATED_CPLD	11:0	0x0	For completion TLP data, the number of FC units granted to transmitter since initialization, modulo 4096
RX_CREDITS_ALLOCATED_CPLH	23:16	0x0	For completion TLP header, the number of FC units granted to transmitter since initialization, modulo 256

RX Credits Allocated Register (Completion)

PCIE_RX_CREDITS_RECEIVED_P - R - 32 bits - PCIEIND_P:0x83			
Field Name	**Bits**	**Default**	**Description**
RX_CREDITS_RECEIVED_PD	11:0	0x0	For posted TLP data, the number of FC units consumed by valid TLP received since initialization, modulo 4096
RX_CREDITS_RECEIVED_PH	23:16	0x0	For posted TLP header, the number of FC units consumed by valid TLP received since initialization, modulo 256

RX Credits Received Register (Posted)

PCIE_RX_CREDITS_RECEIVED_NP - R - 32 bits - PCIEIND_P:0x84			
Field Name	**Bits**	**Default**	**Description**
RX_CREDITS_RECEIVED_NPD	11:0	0x0	For non-posted TLP data, the number of FC units consumed by valid TLP received since initialization, modulo 4096
RX_CREDITS_RECEIVED_NPH	23:16	0x0	For non-posted TLP header, the number of FC units consumed by valid TLP received since initialization, modulo 256

RX Credits Received Register (Non-Posted)

PCIE_RX_CREDITS_RECEIVED_CPL - R - 32 bits - PCIEIND_P:0x85			
Field Name	**Bits**	**Default**	**Description**
RX_CREDITS_RECEIVED_CPLD	11:0	0x0	For completion TLP data, the number of FC units consumed by valid TLP received since initialization, module 4096
RX_CREDITS_RECEIVED_CPLH	23:16	0x0	For completion TLP header, the number of FC units consumed by valid TLP received since initialization, module 256

RX Credits Received Register (Completion)

PCIE_LC_CNTL - RW - 32 bits - PCIEIND_P:0xA0			
Field Name	**Bits**	**Default**	**Description**
LC_CM_HI_ENABLE_COUNT	0	0x0	Enable count for CM_HIGH - when transmitter is to be turned on stop when the counter reaches CM_HI_COUNT_LIMIT_ON. If number of lanes = 1 or 2: CM_HI_COUNT_LIMIT_ON = 12 or 10. If number of lanes = 3 or 4: CM_HI_COUNT_LIMIT_ON = 10 or 12. If number of lanes > 4: CM_HI_COUNT_LIMIT_ON = 10 or 15.
LC_DONT_ENTER_L23_IN_D0	1	0x0	Do not enter L23 in D0 state.
LC_RESET_L_IDLE_COUNT_EN	2	0x0	Enable reset of electrical idle counter.
LC_RESET_LINK	3	0x0	Reset an individual link without resetting the other ports.
LC_16X_CLEAR_TX_PIPE	7:4	0x5	Adjust the time that the LC waits for the pipe to be idle. Setting this field to 0 results in the maximum time. Otherwise, the delay increases as this field is incremented.
LC_L0S_INACTIVITY	11:8	0x0	L0s inactivity timer setting 0=L0s is disabled 1=40ns 2=80ns 3=120ns 4=200ns 5=400ns 6=1us 7=2us 8=4us 9=10us 10=40us 11=100us 12=400us 13=1ms 14=4ms
LC_L1_INACTIVITY	15:12	0x0	L1 inactivity timer setting 0=L1 is disabled 1=1us 2=2us 3=4us 4=10us 5=20us 6=40us 7=100us 8=400us 9=1ms 10=4ms 11=10ms 12=40ms 13=100ms 14=400ms
LC_PMI_TO_L1_DIS	16	0x0	Disable the transition to L1 caused by programming PMI_STATE to non-D0
LC_INC_N_FTS_EN	17	0x0	Enable incrementing N_FTS for each transition to recovery
LC_LOOK_FOR_IDLE_IN_L1L23	19:18	0x0	Controls the number of clocks to wait for Electrical Idle set in L1, L23 0=250 1=100 2=10000 3=3000000
LC_FACTOR_IN_EXT_SYNC	20	0x0	Factor in the extended sync bit in the calculation for the replay timer adjustment

LC_WAIT_FOR_PM_ACK_DIS	21	0x0	Disables waiting for PM_ACK in L23 ready entry handshake
LC_WAKE_FROM_L23	22	0x0	For upstream component, wake the link from L23 ready
LC_L1_IMMEDIATE_ACK	23	0x0	Always ACK an ASPM L1 entry DLLP (ie. never generate PM_NAK)
LC_ASPM_TO_L1_DIS	24	0x0	Disable ASPM L1
LC_DELAY_COUNT	26:25	0x0	Controls minimum amount of time to stay in L0s or L1 0=255/ 4095 (Power-down) 1=1250 / 16383 (Power-down) 2=5000/ 65535 (Power-down) 3=25000 / 262143 (Power-down)
LC_DELAY_L0S_EXIT	27	0x0	Enable staying in L0s for a minimum time
LC_DELAY_L1_EXIT	28	0x0	Enable staying in L1 for a minimum time
LC_EXTEND_WAIT_FOR_EL_IDLE	29	0x1	Wait for Electrical idle in L1/L23 ready value
LC_ESCAPE_L1L23_EN	30	0x1	Enable L1/L23 entry escape arcs
LC_GATE_RCVR_IDLE	31	0x0	Ignore PHY Electrical idle detector 0=LC will look for PE_LC_IdleDetected 1=To gate off PE_LC_IdleDetected to LC, so that LC never sees receivers enter EIDLE

Link Control Register

PCIE_LC_CNTL2 - RW - 32 bits - PCIEIND_P:0xB1			
Field Name	**Bits**	**Default**	**Description**
LC_TIMED_OUT_STATE (R)	5:0	0x0	State that the LC was in when the deadman timer expired.
LC_STATE_TIMED_OUT	6	0x0	Deadman timer expired.
LC_LOOK_FOR_BW_REDUCTION	7	0x1	Enable check for bandwidth change when reporting Link Bandwidth Notification Status. 0=Do not check if bandwidth was reduced. 1=Check if bandwidth was reduced.
LC_MORE_TS2_EN	8	0x0	Send out 128 sets instead of 16.
LC_X12_NEGOTIATION_DIS	9	0x1	Disable x12 negotiation.
LC_LINK_UP_REVERSAL_EN	10	0x0	Allow reversal for a wider width in link up.
LC_ILLEGAL_STATE	11	0x0	The LC is in an illegal state.
LC_ILLEGAL_STATE_RESTART_EN	12	0x0	Enable the LC to be restarted when it is in an illegal state.
LC_WAIT_FOR_OTHER_LANES_MODE	13	0x0	Eliminate delay introduced by waiting for other lanes. 0=Identical Training Set based . 1=Timer based.
LC_ELEC_IDLE_MODE	15:14	0x0	Electrical Idle Mode for LC. 0=GEN1 - entry:PHY, exit:PHY ; GEN2 - entry:infer, exit:PHY 1=GEN1 - entry:infer, exit:PHY; GEN2 - entry:infer, exit PHY 2=GEN1 - entry:PHY, exit:PHY ; GEN2 - entry:PHY, exit:PHY 3=Reserved
LC_DISABLE_INFERRED_ELEC_IDLE_DET	16	0x0	Disable Inferred Electrical Idle detection. 0=Inferred Electrical Idle Detection is enabled 1=Inferred Electrical Idle Detection is disabled
LC_ALLOW_PDWN_IN_L1	17	0x0	Set the BIF_CHIP_CLK_PDWN output to 1 when the LC is in the L1 state.
LC_ALLOW_PDWN_IN_L23	18	0x0	Set the BIF_CHIP_CLK_PDWN output to 1 when the LC is in the L23_Ready state.
LC_DEASSERT_RX_EN_IN_L0S	19	0x0	Turn off transmitters when the link is in L0s.
LC_BLOCK_EL_IDLE_IN_L0	20	0x0	Prevent the Electrical Idle from causing a transition from Rcv_L0 to Rcv_L0s.
LC_RCV_L0_TO_RCV_L0S_DIS	21	0x0	Disable transition from Rcv_L0 to Rcv_L0s
LC_ASSERT_INACTIVE_DURING_HOLD	22	0x0	Assert the INACTIVE_LANES signals when CHIP_BIF_hold_training is high.
LC_WAIT_FOR_LANES_IN_LW_NEG	24:23	0x0	
LC_PWR_DOWN_NEG_OFF_LANES	25	0x1	
LC_DISABLE_LOST_SYM_LOCK_ARCS	26	0x1	

LC_LINK_BW_NOTIFICATION_DIS (R)	27	0x0	
LC_ENABLE_RX_CR_EN_DEASSERTION	28	0x0	To enable deassertion of PG2RX_CR_EN to lock clock recovery parameter when lane is in electrical idle 0=CR_EN is always asserted 1=CR_EN is deasserted when RX_EN is deasserted during L0s/L1 and inactive lanes
LC_TEST_TIMER_SEL	30:29	0x0	State timeout select 0=LTSSM uses spec compliant timeout values. 1=LTSSM uses simulation timeout values. 2=LTSSM uses decreased timeout values for lab testing. 3=Reserved
LC_ENABLE_INFERRED_ELEC_IDLE_FOR_PI	31	0x1	Enable Inferred Electrical Idle Detection for PI (Physical Layer blocks) 0=Inferred Electrical Idle Detection is disabled for PI (Physical Layer block) 1=Inferred Electrical Idle Detection is enabled for PI (Physical Layer block)

Link Control Register 2

PCIE_LC_LINK_WIDTH_CNTL - RW - 32 bits - PCIEIND_P:0xA2			
Field Name	**Bits**	**Default**	**Description**
LC_LINK_WIDTH	2:0	0x6	RESERVED
LC_LINK_WIDTH_RD (R)	6:4	0x0	Read back link width
LC_RECONFIG_ARC_MISSING_ESCAPE	7	0x0	RESERVED
LC_RECONFIG_NOW	8	0x0	RESERVED
LC_RENEGOTIATION_SUPPORT (R)	9	0x0	RESERVED 0=Other end does not support link width renegotiation. 1=Other end does support link width renegotiation.
LC_RENEGOTIATE_EN	10	0x0	Enable re-negotiation
LC_SHORT_RECONFIG_EN	11	0x0	RESERVED
LC_UPCONFIGURE_SUPPORT	12	0x0	
LC_UPCONFIGURE_DIS	13	0x0	
LC_UPCFG_WAIT_FOR_RCVR_DIS	14	0x0	0=Enable 1=Disable
LC_UPCFG_TIMER_SEL	15	0x0	0=1 msec 1=use LC_WAIT_FOR_LANES_IN_LW_NEG values
LC_DEASSERT_TX_PDNB	16	0x0	TX_PDNB Control for unused lanes 0=Keep TX_PDNB asserts for unused lanes. 1=Deassert TX_PDNB for unused lanes

Link Width Control

PCIE_LC_N_FTS_CNTL - RW - 32 bits - PCIEIND_P:0xA3			
Field Name	**Bits**	**Default**	**Description**
LC_XMIT_N_FTS	7:0	0xc	Number of FTS to override the strap value
LC_XMIT_N_FTS_OVERRIDE_EN	8	0x0	Enable the previous field to override the strap value.
LC_XMIT_FTS_BEFORE_RECOVERY	9	0x0	Transmit FTS before Recovery.
LC_XMIT_N_FTS_LIMIT	23:16	0xff	Limit that the number of FTS can increment to when incrementing is enabled.
LC_N_FTS (R)	31:24	0x0	Number of FTS captured from the other end of the link.

LC Number of FTS Control

PCIE_LC_STATE0 - R - 32 bits - PCIEIND_P:0xA5			
Field Name	**Bits**	**Default**	**Description**
LC_CURRENT_STATE	5:0	0x0	Current LC State
LC_PREV_STATE1	13:8	0x0	1st Previous LC State
LC_PREV_STATE2	21:16	0x0	2nd Previous LC State
LC_PREV_STATE3	29:24	0x0	3rd Previous LC State

Link Control State Register

PCIE_LC_STATE1 - R - 32 bits - PCIEIND_P:0xA6			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE4	5:0	0x0	4th Previous LC State
LC_PREV_STATE5	13:8	0x0	5th Previous LC State
LC_PREV_STATE6	21:16	0x0	6th Previous LC State
LC_PREV_STATE7	29:24	0x0	7th Previous LC State

Link Control State Register

PCIE_LC_STATE2 - R - 32 bits - PCIEIND_P:0xA7			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE8	5:0	0x0	8th Previous LC State
LC_PREV_STATE9	13:8	0x0	9th Previous LC State
LC_PREV_STATE10	21:16	0x0	10th Previous LC State
LC_PREV_STATE11	29:24	0x0	11th Previous LC State

Link Control State Register

PCIE_LC_STATE3 - R - 32 bits - PCIEIND_P:0xA8			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE12	5:0	0x0	12th Previous LC State
LC_PREV_STATE13	13:8	0x0	13th Previous LC State
LC_PREV_STATE14	21:16	0x0	14th Previous LC State
LC_PREV_STATE15	29:24	0x0	15th Previous LC State

Link Control State Register

PCIE_LC_STATE4 - R - 32 bits - PCIEIND_P:0xA9			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE16	5:0	0x0	16th Previous LC State
LC_PREV_STATE17	13:8	0x0	17th Previous LC State
LC_PREV_STATE18	21:16	0x0	18th Previous LC State
LC_PREV_STATE19	29:24	0x0	19th Previous LC State

Link Control State Register

PCIE_LC_STATE5 - R - 32 bits - PCIEIND_P:0xAA			
Field Name	**Bits**	**Default**	**Description**
LC_PREV_STATE20	5:0	0x0	20th Previous LC State
LC_PREV_STATE21	13:8	0x0	21st Previous LC State
LC_PREV_STATE22	21:16	0x0	22nd Previous LC State
LC_PREV_STATE23	29:24	0x0	23rd Previous LC State

Link Control State Register

VENDOR_ID - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x0]			
Field Name	**Bits**	**Default**	**Description**
VENDOR_ID (R)	15:0	0x1002	This field identifies the manufacturer of the device. 0FFFFh is an invalid value for Vendor ID.

Vendor Identification

DEVICE_ID - R - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x2]			
Field Name	**Bits**	**Default**	**Description**
DEVICE_ID	15:0	0x0	This field identifies the particular device. This identifier is allocated by the vendor.

Device Identification

COMMAND - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x4]			
Field Name	**Bits**	**Default**	**Description**
IO_ACCESS_EN	0	0x0	Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses.State after RST# is 0. 0=Disable 1=Enable
MEM_ACCESS_EN	1	0x0	Controls a device's response to Memory Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to Memory Space accesses. State after RST# is 0. 0=Disable 1=Enable
BUS_MASTER_EN	2	0x0	Controls the ability of a PCI Express Endpoint to issue Memory and I/O Read/Write Requests, and the ability of a Root or Switch Port to forward Memory and I/O Read/Write Requests in the upstream direction. 0=Disable 1=Enable

SPECIAL_CYCLE_EN (R)	3	0x0	Does not apply to PCI Express. Hardwired to 0. 0=Disable 1=Enable
MEM_WRITE_INVALIDATE_EN (R)	4	0x0	Does not apply to PCI Express. Hardwired to 0. 0=Disable 1=Enable
PAL_SNOOP_EN (R)	5	0x0	Does not apply to PCI Express. Hardwired to 0. 0=Disable 1=Enable
PARITY_ERROR_RESPONSE	6	0x0	Parity Error Response. Default value of this field is 0. 0=Disable 1=Enable
AD_STEPPING (R)	7	0x0	Address and Data Stepping. Does not apply to PCI Express. Hardwired to 0. 0=Disable 1=Enable
SERR_EN	8	0x0	When set, enables reporting of Non-fatal and Fatal errors detected by the device to the Root Complex. 0=Disable 1=Enable
FAST_B2B_EN (R)	9	0x0	Does not apply to PCI Express. Hardwired to 0. 0=Disable 1=Enable
INT_DIS	10	0x0	Controls the ability of a PCI Express device to generate INTx interrupt Messages. When set, devices are prevented from generating INTx interrupt Messages. Default value 0 0=Disable 1=Enable

The Command register provides coarse control over a device's ability to generate and respond to PCI cycles.

STATUS - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x6]			
Field Name	**Bits**	**Default**	**Description**
INT_STATUS (R)	3	0x0	Indicates that an INTx interrupt Message is pending internally to the device.
CAP_LIST (R)	4	0x1	Indicates the presence of an extended capability list item. Since all PCI Express devices are required to implement the PCI Express capability structure, this bit must be set to 1.
PCI_66_EN (R)	5	0x0	Does not apply to PCI Express. Hardwired to 0.
UDF_EN (R)	6	0x0	User Defined Status Enable 0=Disable 1=Enable
FAST_BACK_CAPABLE (R)	7	0x0	Does not apply to PCI Express. Hardwired to 0.
MASTER_DATA_PARITY_ERROR	8	0x0	This bit is set by Requestor if its Parity Error Enable bit is set and either of the following two conditions occurs: 1) Requestor receives a Completion marked poisoned 2) Requestor poisons a write Request 0=Inactive 1=Active
DEVSEL_TIMING (R)	10:9	0x0	Does not apply to PCI Express. Hardwired to 0.
SIGNAL_TARGET_ABORT (R)	11	0x0	This bit is set when a device completes a Request using Completer Abort Completion Status. 0=No Abort 1=Target Abort

RECEIVED_TARGET_ABORT	12	0x0	This bit is set when a Requestor receives a Completion with Unsupported Request Completion Status. 0=Inactive 1=Active
RECEIVED_MASTER_ABORT	13	0x0	This bit is set when a Requestor receives a Completion with Unsupported Request Completion Status. 0=Inactive 1=Active
SIGNALED_SYSTEM_ERROR	14	0x0	This bit must be set whenever the device asserts SERR#. 0=No Error 1=SERR assert
PARITY_ERROR_DETECTED	15	0x0	This bit is set when a device sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR Enable bit in the Command register is 1.

The Status register is used to record status information for PCI bus related events.

REVISION_ID - R - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x8]			
Field Name	**Bits**	**Default**	**Description**
MINOR_REV_ID	3:0	0x0	Major revision ID. Set by the vendor.
MAJOR_REV_ID	7:4	0x0	Minor revision ID. Set by the vendor.

Specifies a device specific revision identifier. The value is chosen by the vendor.

PROG_INTERFACE - R - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x9]			
Field Name	**Bits**	**Default**	**Description**
PROG_INTERFACE	7:0	0x0	Unused, only in test environment

Register-Level Programming Interface Register

SUB_CLASS - R - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xA]			
Field Name	**Bits**	**Default**	**Description**
SUB_CLASS	7:0	0x0	The Class Code register is read-only and is used with the Base Class Code to identify the specific type of device.

Sub Class Code Register

BASE_CLASS - R - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xB]			
Field Name	**Bits**	**Default**	**Description**
BASE_CLASS	7:0	0x0	The Class Code register is read-only and is used to identify the generic function of the device.

Base Class Code Register

CACHE_LINE - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xC]			
Field Name	**Bits**	**Default**	**Description**
CACHE_LINE_SIZE	7:0	0x0	This read/write register specifies the system cacheline size in units of DWORDs.

Cache Line Size Register

LATENCY - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xD]			
Field Name	**Bits**	**Default**	**Description**
LATENCY_TIMER (R)	7:0	0x0	Primary/Master latency timer does not apply to PCI Express. Register is hardwired to 0.

Master Latency Timer Register

HEADER - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xE]			
Field Name	**Bits**	**Default**	**Description**
HEADER_TYPE (R)	6:0	0x0	Type 0 or Type 1 Configuration Space
DEVICE_TYPE (R)	7	0x0	Single function or multi function device 0=Single-Function Device 1=Multi-Function Device

Configuration Space Header

BIST - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xF]			
Field Name	**Bits**	**Default**	**Description**
BIST_COMP (R)	3:0	0x0	A value of 0 means the device has passed its test. Non-zero values mean the device failed. Device-specific failure codes can be encoded in the non-zero value.
BIST_STRT (R)	6	0x0	Write a 1 to invoke BIST. Device resets the bit when BIST is complete. Software should fail the device if BIST is not complete after 2 seconds.
BIST_CAP (R)	7	0x0	This bit is read-only and returns 1 the bridge supports BIST, otherwise 0 is returned

Built In Self Test Register used for control and status of built-in self tests

CAP_PTR - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x34]			
Field Name	**Bits**	**Default**	**Description**
CAP_PTR (R)	7:0	0x50	Pointer to a linked list of additional capabilities implemented by this device. 50=Point to PM Capability

Capability Pointer

INTERRUPT_LINE - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x3C]			
Field Name	**Bits**	**Default**	**Description**
INTERRUPT_LINE Interrupt Line Register	7:0	0xff	Interrupt Line register communicates interrupt line routing information.

INTERRUPT_PIN - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x3D]			
Field Name	**Bits**	**Default**	**Description**
INTERRUPT_PIN (R) Interrupt Pin Register	7:0	0x0	The Interrupt Pin is a read-only register that identifies the legacy interrupt Message(s) the device (or device function) uses NOTE: Bits 3:7 of this field are hardwired to ZERO.

ADAPTER_ID - R - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x2C]			
Field Name	**Bits**	**Default**	**Description**
SUBSYSTEM_VENDOR_ID *(mirror of ADAPTER_ID_W:SUBSYSTEM_VENDOR_ID)*	15:0	0x0	Subsystem Vendor ID. Specified by the vendor.
SUBSYSTEM_ID *(mirror of ADAPTER_ID_W:SUBSYSTEM_ID)* Subsystem Vendor and Subsystem ID Register	31:16	0x0	Subsystem ID. Specified by the vendor.

MIN_GRANT - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x3E]			
Field Name	**Bits**	**Default**	**Description**
MIN_GNT (R)	7:0	0x0	Registers do not apply to PCI Express. Hardwired to 0.

MAX_LATENCY - RW - 8 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x3F]			
Field Name	**Bits**	**Default**	**Description**
MAX_LAT (R)	7:0	0x0	Registers do not apply to PCI Express. Hardwired to 0.

ADAPTER_ID_W - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x4C]

Field Name	Bits	Default	Description
SUBSYSTEM_VENDOR_ID	15:0	0x0	Subsystem Vendor ID. Specified by the vendor.
SUBSYSTEM_ID	31:16	0x0	Subsystem Vendor ID. Specified by the vendor.

Adapter ID

PMI_CAP_LIST - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x50]			
Field Name	**Bits**	**Default**	**Description**
CAP_ID (R)	7:0	0x1	Capability ID Must be set to 01h 1=PCIE Power Management Registers
NEXT_PTR (R)	15:8	0x58	Next Capability Pointer

Power Management Capbility List

PMI_CAP - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x52]			
Field Name	**Bits**	**Default**	**Description**
VERSION (R)	2:0	0x3	Version 3=PMI Spec 1.2
PME_CLOCK (R)	3	0x0	Does not apply to PCI Express. Hardwired to 0.
DEV_SPECIFIC_INIT (R)	5	0x0	Device Specific Initialization
AUX_CURRENT (R)	8:6	0x0	AUX Current
D1_SUPPORT (R)	9	0x0	D1 Support 1=Support D1 PM State.
D2_SUPPORT (R)	10	0x0	D2 Support 1=Support D2 PM State.
PME_SUPPORT (R)	15:11	0x0	For a device, this indicates the power states in which the device may generate a PME.

Power Management Capabilities Register

PMI_STATUS_CNTL - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x54]			
Field Name	**Bits**	**Default**	**Description**
POWER_STATE	1:0	0x0	Power State
NO_SOFT_RESET (R)	3	0x0	
PME_EN (R)	8	0x0	PME Enable
DATA_SELECT (R)	12:9	0x0	Data Select
DATA_SCALE (R)	14:13	0x0	Data Scale
PME_STATUS (R)	15	0x0	PME Status
B2_B3_SUPPORT (R)	22	0x0	B2/B3 Support Does not apply to PCI Express. Hardwired to 0.
BUS_PWR_EN (R)	23	0x0	Bus Power/Clock Control Enable Does not apply to PCI Express. Hardwired to 0.
PMI_DATA (R)	31:24	0x0	Data

Power Management Status/Control Register

PCIE_CAP_LIST - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x58]			
Field Name	**Bits**	**Default**	**Description**

CAP_ID (R)	7:0	0x10	Indicates the PCI Express Capability structure. This field must return a Capability ID of 10h indicating that this is a PCI Express Capability structure. 10=PCI Express capable
NEXT_PTR (R)	15:8	0xa0	Next Capability Pointer -- The offset to the next PCI capability structure or 00h if no other items exist in the linked list of capabilities.

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 2.3 configuration space capability list.

PCIE_CAP - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x5A]			
Field Name	**Bits**	**Default**	**Description**
VERSION (R)	3:0	0x2	Indicates PCI-SIG defined PCI Express capability structure version number. 0=PCI Express Cap Version
DEVICE_TYPE (R)	7:4	0x0	Indicates the type of PCI Express logical device. 0=PCI Express Endpoint 1=Legacy PCI Express Endpoint 4=PCI Express Root Complex
SLOT_IMPLEMENTED (R)	8	0x0	This bit when set indicates that the PCI Express Link associated with this Port is connected to a slot
INT_MESSAGE_NUM (R)	13:9	0x0	Interrupt Message Number.
TCS_ROUTING_SUPPORTED (R)	14	0x0	Trusted Configuration Routing supported.

The PCI Express Capabilities register identifies PCI Express device type and associated capabilities.

DEVICE_CAP - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x5C]			
Field Name	**Bits**	**Default**	**Description**
MAX_PAYLOAD_SUPPORT (R)	2:0	0x0	This field indicates the maximum payload size that the device can support for TLPs. 0=128B size
PHANTOM_FUNC (R)	4:3	0x0	This field indicates the support for use of unclaimed function numbers to extend the number of outstanding transactions allowed by logically combining unclaimed function numbers with the Tag identifier. 0=No Phantom Functions
EXTENDED_TAG (R)	5	0x1	This field indicates the maximum supported size of the Tag field as a Requester. 0=8 Bit Tag Supported
L0S_ACCEPTABLE_LATENCY (R)	8:6	0x0	This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state.
L1_ACCEPTABLE_LATENCY (R)	11:9	0x0	This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state.
ROLE_BASED_ERR_REPORTING (R)	15	0x0	0=Role-Based Error Reporting Disabled 1=Role-Based Error Reporting Enabled
CAPTURED_SLOT_POWER_LIMIT (R)	25:18	0x0	(Upstream Ports only) In combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by slot.
CAPTURED_SLOT_POWER_SCALE (R)	27:26	0x0	Specifies the scale used for the Slot Power Limit Value.
FLR_CAPABLE (R)	28	0x0	This field indicates that a device is capable of initiating Function Level Resets.

The Device Capabilities register identifies PCI Express device specific capabilities.

DEVICE_CNTL - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x60]			
Field Name	**Bits**	**Default**	**Description**
CORR_ERR_EN	0	0x0	This bit controls reporting of correctable errors. Default value of this field is 0. 0=Disable 1=Enable
NON_FATAL_ERR_EN	1	0x0	This bit controls reporting of Non-fatal errors. Default value of this field is 0. 0=Disable 1=Enable
FATAL_ERR_EN	2	0x0	This bit controls reporting of Fatal errors. Default value of this field is 0. 0=Disable 1=Enable
USR_REPORT_EN	3	0x0	This bit enables reporting of Unsupported Requests when set. Default value of this field is 0. 0=Disable 1=Enable
RELAXED_ORD_EN	4	0x1	If this bit is set, the device is permitted to set the Relaxed Ordering bit in the Attributes field of transactions it initiates that do not require strong write ordering. Default value of this bit is 1. 0=Disable 1=Enable
MAX_PAYLOAD_SIZE (R)	7:5	0x0	This field sets maximum TLP payload size for the device. Default value of this field is 000b. 0=128B size
EXTENDED_TAG_EN	8	0x0	When set, this bit enables a device to use an 8-bit Tag field as a requester. If the bit is cleared, the device is restricted to a 5-bit Tag field. Default value of this field is 0. 0=Disable 1=Enable
PHANTOM_FUNC_EN (R)	9	0x0	When set, this bit enables a device to use unclaimed functions as Phantom Functions to extend the number of outstanding transaction identifiers. If the bit is cleared, the device is not allowed to use Phantom Functions. 0=Disable 1=Enable
AUX_POWER_PM_EN (R)	10	0x0	This bit when set enables a device to draw AUX power independent of PME AUX power. 0=Disable 1=Enable
NO_SNOOP_EN	11	0x1	If this bit is set to 1, the device is permitted to set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency. Default value of this bit is 1. 0=Disable 1=Enable
MAX_REQUEST_SIZE (R)	14:12	0x0	This field sets the maximum Read Request size for the Device as a Requester. Default value of this field is 010b. 0=128B size
BRIDGE_CFG_RETRY_EN (R)	15	0x0	0=Disable 1=Enable

The Device Control register controls PCI Express device specific parameters.

DEVICE_STATUS - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x62]			
Field Name	**Bits**	**Default**	**Description**
CORR_ERR	0	0x0	This bit indicates status of correctable errors detected.
NON_FATAL_ERR	1	0x0	This bit indicates status of Nonfatal errors detected.
FATAL_ERR	2	0x0	This bit indicates status of Fatal errors detected.
USR_DETECTED	3	0x0	This bit indicates that the device received an Unsupported Request.
AUX_PWR (R)	4	0x0	Devices that require AUX power report this bit as set if AUX power is detected by the device.
TRANSACTIONS_PEND (R)	5	0x0	Endpoints: This bit when set indicates that the device has issued Non-Posted Requests which have not been completed. Root and Switch Ports: This bit when set indicates that a Port has issued Non-Posted Requests on its own behalf (using the Port's own Requester ID) which have not been completed.

The Device Status register provides information about PCI Express device specific parameters.

LINK_CAP - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x64]			
Field Name	**Bits**	**Default**	**Description**
LINK_SPEED (R)	3:0	0x1	This field indicates the maximum Link speed of the given PCI Express Link. 1=2.5 Gb/s 2=5.0 Gb/s
LINK_WIDTH (R)	9:4	0x0	This field indicates the maximum width of the given PCI Express Link. 1=x1 2=x2 4=x4 8=x8 12=x12 16=x16 32=x32
PM_SUPPORT (R)	11:10	0x3	This field indicates the level of ASPM supported on the given PCI Express Link.
L0S_EXIT_LATENCY (R)	14:12	0x1	This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0.
L1_EXIT_LATENCY (R)	17:15	0x2	This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0.
CLOCK_POWER_MANAGEMENT (R)	18	0x0	
SURPRISE_DOWN_ERR_REPORTING (R)	19	0x0	
DL_ACTIVE_REPORTING_CAPABLE (R)	20	0x0	
LINK_BW_NOTIFICATION_CAP (R)	21	0x0	
PORT_NUMBER (R)	31:24	0x0	This field indicates the PCI Express Port number for the given PCI Express Link.

The Link Capabilities register identifies PCI Express Link specific capabilities.

LINK_CNTL - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x68]			
Field Name	**Bits**	**Default**	**Description**
PM_CONTROL	1:0	0x0	This field controls the level of ASPM supported on the given PCI Express Link. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled
READ_CPL_BOUNDARY (R)	3	0x0	Read Completion Boundary. Indicates the RCB value for the Root Port 0=64 Byte 1=128 Byte
LINK_DIS (R)	4	0x0	This bit disables the Link when set to 1b. Default value of this field is 0b.
RETRAIN_LINK (R)	5	0x0	A write of 1b to this bit initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. Reads of this bit always return 0b.
COMMON_CLOCK_CFG	6	0x0	This bit when set indicates that this component and the component at the opposite end of this Link are operating with a distributed common reference clock. Default value of this field is 0b.
EXTENDED_SYNC	7	0x0	This bit when set forces the transmission of 4096 FTS ordered sets in the L0s state followed by a single SKP ordered set
CLOCK_POWER_MANAGEMENT_EN	8	0x0	This bit determines if device is permitted to use CLKREQ# signal to power manage link clock.
HW_AUTONOMOUS_WIDTH_DISABLE	9	0x0	When set to 1, this bit disables hardware from changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width.
LINK_BW_MANAGEMENT_INT_EN (R)	10	0x0	
LINK_AUTONOMOUS_BW_INT_EN (R)	11	0x0	

The Link Control register controls PCI Express Link specific parameters.

LINK_STATUS - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x6A]			
Field Name	**Bits**	**Default**	**Description**
CURRENT_LINK_SPEED (R)	3:0	0x1	Indicates the negotiated Link speed of the given PCI Express Link 1=2.5 Gb/s 2=5.0 Gb/s
NEGOTIATED_LINK_WIDTH (R)	9:4	0x0	This field indicates the negotiated width of the given PCI Express Link. Defined encodings are: 000001b X1 000010b X2 000100b X4 001000b X8 001100b X12 010000b X16 100000b X32 All other encodings are reserved. 1=x1 2=x2 4=x4 8=x8 12=x12 16=x16 32=x32

LINK_TRAINING (R)	11	0x0	This read-only bit indicates that Link training is in progress (Physical Layer LTSSM in Configuration or Recovery state) or that 1b was written to the Retrain Link bit but Link training has not yet begun. Hardware clears this bit once Link training is complete.
SLOT_CLOCK_CFG (R)	12	0x1	This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. If the device uses an independent clock irrespective of the presence of a reference on the connector, this bit must be clear. 0=Diff Clock 1=Same Clock
DL_ACTIVE (R)	13	0x0	
LINK_BW_MANAGEMENT_STATUS (R)	14	0x0	
LINK_AUTONOMOUS_BW_STATUS (R)	15	0x0	

The Link Status register provides information about PCI Express Link specific parameters.

DEVICE_CAP2 - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x7C]			
Field Name	**Bits**	**Default**	**Description**
CPL_TIMEOUT_RANGE_SUP (R)	3:0	0x0	PCIE completion timeout range supported
CPL_TIMEOUT_DIS_SUP (R)	4	0x0	PCIE completion timeout disabled supported

The Device Capabilities 2 register identifies PCI Express device specific capabilities.

DEVICE_CNTL2 - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x80]			
Field Name	**Bits**	**Default**	**Description**
CPL_TIMEOUT_VALUE	3:0	0x0	PCIE completion timeout value
CPL_TIMEOUT_DIS	4	0x0	Disable PCIE completion timeout

The Device Control 2 register controls PCI Express device specific parameters.

DEVICE_STATUS2 - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x82]			
Field Name	**Bits**	**Default**	**Description**
RESERVED (R)	15:0	0x0	

The Device Status 2 register provides information about PCI Express device specific parameters.

LINK_CAP2 - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x84]			
Field Name	**Bits**	**Default**	**Description**
RESERVED (R)	31:0	0x0	

The Link Capabilities 2 register identifies PCI Express Link specific capabilities.

LINK_CNTL2 - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x88]			
Field Name	**Bits**	**Default**	**Description**
TARGET_LINK_SPEED	3:0	0x1	The upper limit on the operational speed. This field restricts the data rate values advertised by an upstream component.
ENTER_COMPLIANCE	4	0x0	This bit forces a port's transmitter to enter Compliance.
HW_AUTONOMOUS_SPEED_DISABLE	5	0x0	Controls the component's ability to autonomously direct changes in link speed.
DE_EMPHASIS_SEL	6	0x0	Selectable de-emphasis (in GEN 2 data rate) 0 : -6dB, 1 : -3.6dB
DE_EMPHASIS_ENFORCE (R)	7	0x0	For RC, when this bit is set, CHIP should use de-emphasis value in bit 6 and ignore what was sent in TS1 ordereed sets in Recover.RcvrLock
XMIT_MARGIN	10:8	0x0	These bits control the value of the non-deemphasized voltage level at the transmitter pins
ENTER_MOD_COMPLIANCE	11	0x0	LTSSM transmits modified compliance pattern in Polling.Compliance if this bit is set to 1.

The Link Control 2 register controls PCI Express Link specific parameters.

LINK_STATUS2 - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x8A]			
Field Name	**Bits**	**Default**	**Description**
RESERVED (R)	15:0	0x0	

The Link Status 2 register provides information about PCI Express Link specific parameters.

MSI_CAP_LIST - R - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xA0]			
Field Name	**Bits**	**Default**	**Description**
CAP_ID	7:0	0x5	Register identifies if a device function is MSI capable
NEXT_PTR	15:8	0x0	Pointer to the next item on the capabilities list

Message Signaled Interrupt Capability Registers

MSI_MSG_CNTL - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xA2]			
Field Name	**Bits**	**Default**	**Description**
MSI_EN	0	0x0	Enable MSI messaging 0=Disable 1=Enable
MSI_MULTI_CAP (R)	3:1	0x0	Multiple Message Capable register is read to determine the number of requested messages. 0=1 message allocated 1=2 messages allocated 2=4 messages allocated 3=8 messages allocated 4=16 messages allocated 5=32 messages allocated 6=Reserved 7=Reserved

MSI_MULTI_EN	6:4	0x0	Multiple Message Enable register is written to indicate the number of allocated messages. 0=1 message allocated 1=2 messages allocated 2=4 messages allocated 3=8 messages allocated 4=16 messages allocated 5=32 messages allocated 6=Reserved 7=Reserved
MSI_64BIT (R)	7	0x0	Signifies if a device function is capable of generating a 64-bit message address 0=Not capable of generating 1 64-bit message address 1=Capable of generating 1 64-bit message address

Message Signaled Interrupts Control Register

MSI_MSG_ADDR_LO - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xA4]			
Field Name	**Bits**	**Default**	**Description**
MSI_MSG_ADDR_LO	31:2	0x0	Message Lower Address - use lower 32-bits of address

Message Lower Address

MSI_MSG_ADDR_HI - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xA8]			
Field Name	**Bits**	**Default**	**Description**
MSI_MSG_ADDR_HI	31:0	0x0	Message Upper Address - use upper 32-bit of address

Message Upper Address

MSI_MSG_DATA_64 - RW - 16 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xAC]			
Field Name	**Bits**	**Default**	**Description**
MSI_DATA_64	15:0	0x0	Message Data. System specified.

64-bit MSI Message Data

MSI_MSG_DATA - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0xA8]			
Field Name	**Bits**	**Default**	**Description**
MSI_DATA	15:0	0x0	Message Data. System specified.

MSI Message Data

PCIE_ADV_ERR_RPT_ENH_CAP_LIST - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x150]			
Field Name	**Bits**	**Default**	**Description**

CAP_ID (R)	15:0	0x1	This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability.
CAP_VER (R)	19:16	0x1	This field is a PCI-SIG defined version number that indicates the version of the capability structure present.
NEXT_PTR (R)	31:20	0x190	This field contains the offset to the next PCI Express capability structure or 000h if no other items exist in the linked list of capabilities.

Advanced Error Reporting Enhanced Capability header

PCIE_UNCORR_ERR_STATUS - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x154]			
Field Name	**Bits**	**Default**	**Description**
DLP_ERR_STATUS	4	0x0	Data Link Protocol Error Status
SURPDN_ERR_STATUS (R)	5	0x0	
PSN_ERR_STATUS	12	0x0	Poisoned TLP Status
FC_ERR_STATUS (R)	13	0x0	Flow Control Protocol Error Status
CPL_TIMEOUT_STATUS	14	0x0	Completion Timeout Status
CPL_ABORT_ERR_STATUS (R)	15	0x0	Completer Abort Status
UNEXP_CPL_STATUS	16	0x0	Unexpected Completion Status
RCV_OVFL_STATUS (R)	17	0x0	Receiver Overflow Status
MAL_TLP_STATUS	18	0x0	Malformed TLP Status
ECRC_ERR_STATUS (R)	19	0x0	ECRC Error Status
UNSUPP_REQ_ERR_STATUS	20	0x0	Unsupported Request Error Status

The Uncorrectable Error Status register reports error status of individual error sources on a PCI Express device.

PCIE_UNCORR_ERR_MASK - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x158]			
Field Name	**Bits**	**Default**	**Description**
DLP_ERR_MASK	4	0x0	Data Link Protocol Error Mask
SURPDN_ERR_MASK (R)	5	0x0	
PSN_ERR_MASK	12	0x0	Poisoned TLP Mask
FC_ERR_MASK (R)	13	0x0	Flow Control Protocol Error Mask
CPL_TIMEOUT_MASK	14	0x0	Completion Timeout Mask
CPL_ABORT_ERR_MASK (R)	15	0x0	Completer Abort Mask
UNEXP_CPL_MASK	16	0x0	Unexpected Completion Mask
RCV_OVFL_MASK (R)	17	0x0	Receiver Overflow Mask
MAL_TLP_MASK	18	0x0	Malformed TLP Mask
ECRC_ERR_MASK (R)	19	0x0	ECRC Error Mask
UNSUPP_REQ_ERR_MASK	20	0x0	Unsupported Request Error Mask

The Uncorrectable Error Mask register controls reporting of individual errors by the device to the PCI Express Root Complex via a PCI Express error Message.

PCIE_UNCORR_ERR_SEVERITY - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x15C]			
Field Name	**Bits**	**Default**	**Description**
DLP_ERR_SEVERITY	4	0x1	Data Link Protocol Error Severity
SURPDN_ERR_SEVERITY (R)	5	0x1	
PSN_ERR_SEVERITY	12	0x0	Poisoned TLP Severity
FC_ERR_SEVERITY (R)	13	0x1	Flow Control Protocol Error Severity
CPL_TIMEOUT_SEVERITY	14	0x0	Completion Timeout Error Severity
CPL_ABORT_ERR_SEVERITY (R)	15	0x0	Completer Abort Error Severity
UNEXP_CPL_SEVERITY	16	0x0	Unexpected Completion Error Severity
RCV_OVFL_SEVERITY (R)	17	0x1	Receiver Overflow Error Severity
MAL_TLP_SEVERITY	18	0x1	Malformed TLP Severity

ECRC_ERR_SEVERITY (R)	19	0x0	ECRC Error Severity
UNSUPP_REQ_ERR_SEVERITY	20	0x0	Unsupported Request Error Severity

The Uncorrectable Error Severity register controls whether an individual error is reported as a Nonfatal or Fatal error.

PCIE_CORR_ERR_STATUS - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x160]			
Field Name	**Bits**	**Default**	**Description**
RCV_ERR_STATUS	0	0x0	Receiver Error Status (
BAD_TLP_STATUS	6	0x0	Bad TLP Status
BAD_DLLP_STATUS	7	0x0	Bad DLLP Status
REPLAY_NUM_ROLLOVER_STATUS	8	0x0	REPLAY_NUM Rollover Status
REPLAY_TIMER_TIMEOUT_STATUS	12	0x0	Replay Timer Timeout Status
ADVISORY_NONFATAL_ERR_STATUS	13	0x0	

The Correctable Error Status register reports error status of individual correctable error sources on a PCI Express device.

PCIE_CORR_ERR_MASK - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x164]			
Field Name	**Bits**	**Default**	**Description**
RCV_ERR_MASK	0	0x0	Receiver Error Mask
BAD_TLP_MASK	6	0x0	Bad TLP Mask
BAD_DLLP_MASK	7	0x0	Bad DLLP Mask
REPLAY_NUM_ROLLOVER_MASK	8	0x0	REPLAY_NUM Rollover Mask
REPLAY_TIMER_TIMEOUT_MASK	12	0x0	Replay Timer Timeout Mask
ADVISORY_NONFATAL_ERR_MASK	13	0x1	

The Correctable Error Mask register controls reporting of individual correctable errors by device to the PCI Express Root Complex via a PCI Express error Message.

PCIE_ADV_ERR_CAP_CNTL - RW - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x168]			
Field Name	**Bits**	**Default**	**Description**
FIRST_ERR_PTR (R)	4:0	0x0	The First Error Pointer is a read-only register that identifies the bit position of the first error reported in the Uncorrectable Error Status register.
ECRC_GEN_CAP (R)	5	0x0	This bit indicates that the device is capable of generating ECRC
ECRC_GEN_EN	6	0x0	This bit when set enables ECRC generation. Default value of this field is 0.
ECRC_CHECK_CAP (R)	7	0x0	This bit indicates that the device is capable of checking ECRC
ECRC_CHECK_EN	8	0x0	This bit when set enables ECRC checking. Default value of this field is 0.

Advanced Error Capabilities and Control Register

PCIE_HDR_LOG0 - R - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x16C]			
Field Name	**Bits**	**Default**	**Description**
TLP_HDR	31:0	0x0	TLP Header 1st DW

Header Log Register captures the Header for the TLP corresponding to a detected error;

PCIE_HDR_LOG1 - R - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x170]			
Field Name	**Bits**	**Default**	**Description**
TLP_HDR	31:0	0x0	TLP Header 2nd DW
Header Log Register			

PCIE_HDR_LOG2 - R - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x174]			
Field Name	**Bits**	**Default**	**Description**
TLP_HDR	31:0	0x0	TLP Header 3rd DW
Header Log Register			

PCIE_HDR_LOG3 - R - 32 bits - [AudioPcie GpuF0Pcie,GpuF1Pcie:0x178]			
Field Name	**Bits**	**Default**	**Description**
TLP_HDR	31:0	0x0	TLP Header 4th DW
Header Log Register			

2.4 Clock Generator Registers

PLL_BYPASSCLK_SEL - RW - 32 bits - [GpuF0MMReg:0x608]			
Field Name	**Bits**	**Default**	**Description**
SPLL_CLKOUT_SEL	7:0	0x2	1=VCLK-UPLL 2=BCLK 4=XTALIN 8=PCLK 16=MCLK Channel B/ TEST_MCLK 32=MCLK CHANNEL C/TEST_MCLK 64=TEST_SCLK 128=SCAN_SCLK
MPLL_CLKOUT_SEL	15:8	0x2	1=VCLK-UPLL 2=BCLK 4=XTALIN 8=PCLK 16=TEST_MCLK 32=SPLLOUT 64=TEST_SCLK 128=SCAN_MCLK

SPLL_CNTL_MODE - RW - 32 bits - [GpuF0MMReg:0x60C]			
Field Name	**Bits**	**Default**	**Description**
SPLL_SW_DIR_CONTROL	0	0x1	1=SW controls the PLL directly. SW will make sure the way they program SPLL_FUNC_CNTL register follows the PLL's requested protocol
SPLL_REFCLK_SRC_SEL	1	0x1	0=Ref clock from GPIO 1=Ref clock from XTALIN
SPLL_TEST	2	0x0	1=Enable SPLL test mode
SPLL_FASTEN	3	0x1	1=Enable SPLL fast lock
SPLL_ENSAT	4	0x1	1=Enable saturation behavior
SPLL_DIV_SYNC	5	0x0	1=Enable sync of the FB and RF dividers to the FBEN and RFEN clock from SPLL. This is needed if baby stepping option is used.

Software or HW control.

MPLL_FUNC_CNTL - RW - 32 bits - [GpuF0MMReg:0x610]			
Field Name	**Bits**	**Default**	**Description**
MPLL_RESET	0	0x1	0=Run 1=Reset
MPLL_SLEEP	1	0x0	0=Power Up 1=Power Down
MPLL_REF_DIV	4:2	0x1	Reference Divider DEF=0x1
MPLL_FB_DIV	12:5	0x6f	Feedback Divider DEF=111
MPLL_PULSEEN	13	0x0	0=Don't pulse clock 1=Send the numbre of pulses indicated by PULSENUM
MPLL_PULSENUM	15:14	0x0	Number of pulses to be sent.
MPLL_SW_HILEN	19:16	0x0	Post divider setting: No of high periods of CLKOP
MPLL_SW_LOLEN	23:20	0x0	Post divider setting: No of low periods of CLKOP

MPLL_DIVEN	24	0x0	1=Enable PLL CLKOUT divider
MPLL_BYPASS_EN	25	0x1	1=Enable Bypass mode
MPLL_MCLK_SEL	26	0x0	1=Use MPLL output as mclk
MPLL_CHG_STATUS (R)	29	0x0	1=Previous write/change to MPLL_FUNC_CNTL register has been completed. SW should not issue another write to this register until this bit is asserted
MPLL_CTLREQ	30	0x0	1=For debug purpose: when SW_DIR_CONTROL is set, assert this bit will trigger an update of the PLL clock output mux control. Before write to this bit, HILEN/LOLEN/PULSEEN/PULSENUM should already contain the new set of value
MPLL_CTLACK (R)	31	0x0	1=For debug purpose: when SW_DIR_CONTROL is set, this value replicates the value of the CTLREQ once the command has been received and it is safe to send another request

MPLL function control register

MPLL_CNTL_MODE - RW - 32 bits - [GpuF0MMReg:0x614]			
Field Name	**Bits**	**Default**	**Description**
MPLL_SW_DIR_CONTROL	0	0x1	1=SW controls the PLL directly. SW will make sure the way they program MPLL_FUNC_CNTL register follows the PLL's requested protocol
MPLL_REFCLK_SRC_SEL	1	0x1	0=Ref clock from GPIO 1=Ref clock from XTALIN
MPLL_TEST	2	0x0	1=Enable MPLL test mode
MPLL_FASTEN	3	0x1	1=Enable MPLL fast lock
MPLL_ENSAT	4	0x1	1=Enable saturation behavior

Software of HW control.

GENERAL_PWRMGT - RW - 32 bits - [GpuF0MMReg:0x618]			
Field Name	**Bits**	**Default**	**Description**
GLOBAL_PWRMGT_EN	0	0x0	0=dynamic power managerment off 1=dynamic power managerment on
STATIC_PM_EN	1	0x0	0=Disable 1=Enable
MOBILE_SU	2	0x0	0=Regular 1=Optimize power consumption in Suspend mode for mobile. D2 acts as if in D3 power state.
THERMAL_PROTECTION_DIS	3	0x1	0=thermal protection Enabled 1=thermal protection Disabled
THERMAL_PROTECTION_TYPE	4	0x0	0=Normal protection - do not turn off gfx clock 1= Catastrophic thermal protection - turn off gfx clock
ENABLE_GEN2PCIE	5	0x0	0=Disabled 1=Enabled
SW_GPIO_INDEX	7:6	0x0	
LOW_VOLT_D2_ACPI	8	0x0	0=Enable low voltage during D2 ACPI state
LOW_VOLT_D3_ACPI	9	0x1	0=Enable low voltage during D3 ACPI state
VOLT_PWRMGT_EN	10	0x1	0=Off 1=Volt power management on
OPEN_DRAIN_PADS	11	0x1	0= Resistor divider type 1=Voltage control GPIO PADS are open drain type
AVP_SCLK_EN	12	0x1	0=Turn off sclk to AVP 1=Turn ON sclk to AVP
IDCT_SCLK_EN	13	0x1	0=Turn off sclk to IDCT 1=Turn ON sclk to IDCT

GPU_COUNTER_ACPI	14	0x1	0=Enable counter in all states 1=Stop gpu counter in D1, D2, D3-cold states
GPU_COUNTER_CLK	15	0x0	0=Use 27Mhz crystal clock 1=Use 27/2 = 13.5Mhz clock
BACKBIAS_PAD_EN	16	0x0	1=Pad enable for back bias
BACKBIAS_VALUE	17	0x0	0=Back bias disabled in software control mode 1=Back bias enabled in software control mode
BACKBIAS_DPM_CNTL	18	0x0	0=Back bias software control 1=Back bias DPM controlled
SPREAD_SPECTRUM_INDEX	20:19	0x0	
DYN_SPREAD_SPECTRUM_EN	21	0x0	1=Enable dynamic spread spectrum ctrl during DPM mode

SCLK_PWRMGT_CNTL - RW - 32 bits - [GpuF0MMReg:0x620]			
Field Name	**Bits**	**Default**	**Description**
SCLK_PWRMGT_OFF	0	0x0	0=SCLK power managerment on 1=SCLK power managerment off
SCLK_TURNOFF	1	0x0	0=NOT USED. sclk is always on
SPLL_TURNOFF	2	0x0	1=Enable SPLL Power down during D3 stage, override HW pwrmgt control.
SU_SCLK_USE_BCLK	3	0x0	0=Use slower SCLK under suspend mode 1=Use BCLK as SCLK under suspend mode
DYNAMIC_GFX_ISLAND_PWR_DOWN	4	0x0	0=Disable Power Down 1=Enable Power Down
DYNAMIC_GFX_ISLAND_LP	5	0x0	0=Disable Low Power, value retention mode 1=Enable Low Power
CLK_TURN_ON_STAGGER	6	0x1	0=Disable clock stagger while turning ON clocks 1=Enable
CLK_TURN_OFF_STAGGER	7	0x1	0=Disable clock stagger while turning OFF clocks 1=Enable
FIR_FORCE_TREND_SEL	8	0x0	1=Force Trend select
FIR_TREND_MODE	9	0x0	0=Select UpTrend 1=Select DownTrend
DYN_GFX_CLK_OFF_EN	10	0x0	1=Enable gfx clock to go be turned OFF during dynamic pwr mgmnt
VDDC3D_TURNOFF_D1	11	0x1	1=Enable GFX sclk to be turned off during D1 state
VDDC3D_TURNOFF_D2	12	0x1	1=Enable GFX sclk to be turned off during D2 state
VDDC3D_TURNOFF_D3	13	0x1	1=Enable GFX sclk to be turned off during D3 state
SPLL_TURNOFF_D2	14	0x0	1=Enable SPLL Power down during D2 stage
SCLK_LOW_D1	15	0x0	1=Enable SCLK to low state during D1
DYN_GFX_CLK_OFF_MC_EN	16	0x1	1=Enable gfx clock to be turned OFF for mc tiles during dynamic pwr mgt

SCLK domain static power managerment

MCLK_PWRMGT_CNTL - RW - 32 bits - [GpuF0MMReg:0x624]			
Field Name	**Bits**	**Default**	**Description**
MPLL_PWRMGT_OFF	0	0x0	0=MCLK power managerment on during ACPI 1=MCLK power managerment off during ACPI
YCLK_TURNOFF	1	0x0	0=Turn off YCLK during D2/D3 state
MPLL_TURNOFF	2	0x0	0=Enable M domain PLL to be turned off at power state D3
SU_MCLK_USE_BCLK	3	0x0	0=Shut down MCLK during suspend mode 1=Use BCLK as MCLK under suspend mode

DLL_READY	4	0x0	0=DLL is not ready. Status from Software 1=DLL is ready
MC_BUSY (R)	5	0x0	0=MC is idle 1=MC is not idle
SPARE	6	0x0	0=SPARE
MC_INT_CNTL	7	0x1	0=SW overwrite - Control the DLL lines through software, if HW control doesnt work 1=HW control
MRDCKA_SLEEP	8	0x0	0=Enable Channel A DLL 1=PowerDown Channel A DLL
MRDCKB_SLEEP	9	0x0	0=Enable Channel B DLL 1=PowerDown Channel B DLL
MRDCKC_SLEEP	10	0x0	0=Enable Channel C DLL 1=PowerDown Channel C DLL
MRDCKD_SLEEP	11	0x0	0=Enable Channel D DLL 1=PowerDown Channel D DLL
MRDCKE_SLEEP	12	0x0	0=Enable Channel E DLL 1=PowerDown Channel E DLL
MRDCKF_SLEEP	13	0x0	0=Enable Channel F DLL 1=PowerDown Channel F DLL
MRDCKG_SLEEP	14	0x0	0=Enable Channel G DLL 1=PowerDown Channel G DLL
MRDCKH_SLEEP	15	0x0	0=Enable Channel H DLL 1=PowerDown Channel H DLL
MRDCKA_RESET	16	0x1	0=Enable Channel A DLL 1=Reset Channel A DLL
MRDCKB_RESET	17	0x1	0=Enable Channel B DLL 1=Reset Channel B DLL
MRDCKC_RESET	18	0x1	0=Enable Channel C DLL 1=Reset Channel C DLL
MRDCKD_RESET	19	0x1	0=Enable Channel D DLL 1=Reset Channel D DLL
MRDCKE_RESET	20	0x1	0=Enable Channel E DLL 1=Reset Channel E DLL
MRDCKF_RESET	21	0x1	0=Enable Channel F DLL 1=Reset Channel F DLL
MRDCKG_RESET	22	0x1	0=Enable Channel G DLL 1=Reset Channel G DLL
MRDCKH_RESET	23	0x1	0=Enable Channel H DLL 1=Reset Channel H DLL
DLL_READY_READ (R)	24	0x0	0=DLL is not ready - status from CG 1=DLL is ready
USE_DISPLAY_GAP	25	0x1	0=Not followed 1=Display Gap interface followed
USE_DISP_URGENT_NORMAL	26	0x1	1=Use CG_MC_display_urgent during normal mclk switching
USE_DISPLAY_GAP_CTXSW	27	0x1	1=During context switch use display gap
MPLL_TURNOFF_D2	28	0x0	0=Enable M domain PLL to be turned off at power state D2
USE_DISP_URGENT_CTXSW	29	0x1	1=Use CG_MC_display_urgent during ctxsw mclk switching

DLL_CNTL - RW - 32 bits - [GpuF0MMReg:0x62C]			
Field Name	**Bits**	**Default**	**Description**
DLL_RESET_TIME	9:0	0x1f4	DEF=500
DLL_LOCK_TIME	21:12	0xfa	DEF=250
MRDCKA_BYPASS	24	0x0	0=Enable Bypass Channel A DLL 1=Disable Bypass Channel A DLL

MRDCKB_BYPASS	25	0x0	0=Enable Bypass Channel B DLL 1=Disable Bypass Channel B DLL
MRDCKC_BYPASS	26	0x0	0=Enable Bypass Channel C DLL 1=Disable Bypass Channel C DLL
MRDCKD_BYPASS	27	0x0	0=Enable Bypass Channel D DLL 1=Disable Bypass Channel D DLL
MRDCKE_BYPASS	28	0x0	0=Enable Bypass Channel E DLL 1=Disable Bypass Channel E DLL
MRDCKF_BYPASS	29	0x0	0=Enable Bypass Channel F DLL 1=Disable Bypass Channel F DLL
MRDCKG_BYPASS	30	0x0	0=Enable Bypass Channel G DLL 1=Disable Bypass Channel G DLL
MRDCKH_BYPASS	31	0x0	0=Enable Bypass Channel H DLL 1=Disable Bypass Channel H DLL

DLL control register

SPLL_TIME - RW - 32 bits - [GpuF0MMReg:0x630]			
Field Name	**Bits**	**Default**	**Description**
SPLL_LOCK_TIME	15:0	0x2000	DEF=0x2000
SPLL_RESET_TIME	31:16	0x1f4	DEF=500

SPLL related timing counter

MPLL_TIME - RW - 32 bits - [GpuF0MMReg:0x634]			
Field Name	**Bits**	**Default**	**Description**
MPLL_LOCK_TIME	15:0	0x2000	
MPLL_RESET_TIME	31:16	0x1f4	

MPLL related timing counter

ERROR_STATUS - R - 32 bits - [GpuF0MMReg:0x640]			
Field Name	**Bits**	**Default**	**Description**
OVERCLOCK_DETECTION_SCLK	0	0x0	0=No overclock for SCLK 1=SCLK overclock
OVERCLOCK_DETECTION_YCLK	1	0x0	0=No overclock for YCLK 1=YCLK overclock
SPLL_UNLOCK	2	0x0	
YPLL_UNLOCK	3	0x0	
YPLL2_UNLOCK	4	0x0	
UPLL_UNLOCK	5	0x0	
ACPI_STATE	8:6	0x0	
MCHG_STATE	11:9	0x0	
FCHANGE_STATE	15:12	0x0	
DPM_STATE	19:16	0x0	
SCHANGE_STATE	22:20	0x0	
SPLL_DIVEN_STATE	24:23	0x0	
VCHG_STAGE	26:25	0x0	
SPLL_SW_FSM_STATE	29:27	0x0	

CG related error status, any reads to this register will clean the status.

CG_CLKPIN_CNTL - RW - 32 bits - [GpuF0MMReg:0x644]			
Field Name	**Bits**	**Default**	**Description**
OSC_EN	0	0x1	0=Disable Oscillation 1=Enable Oscillation
XTL_LOW_GAIN	1	0x1	0=High Gain 1=Low Gain
CG_CLK_TO_OUTPIN	2	0x0	0=Disabled 1=Send out selected clock for jitter test
OSC_USE_CORE	3	0x0	0=Pad routing OSC 1=Core routing OSC
TEST_MCLK_RE	4	0x0	0=Receiver Enable for TEST_MCLK pad
TEST_YCLK_RE	5	0x0	0=Receiver Enable for TEST_YCLK pad
GENERICA_OE	6	0x0	0=Enable selected clock to be observed through GENERICA pad
MUX_TCLK_TO_XCLK	7	0x0	1=Mux free running tclk into xclk

PLL_TEST_CNTL - RW - 32 bits - [GpuF0MMReg:0x79C]			
Field Name	**Bits**	**Default**	**Description**
TST_SRC_SEL	3:0	0x0	
TST_REF_SEL	7:4	0x0	
REF_TEST_COUNT	14:8	0x0	
TST_RESET	15	0x0	
TEST_COUNT (R)	31:17	0x0	FIX=0

PLL frequency measurement control

CG_TC_JTAG_0 - RW - 32 bits - [GpuF0MMReg:0x7A0]			
Field Name	**Bits**	**Default**	**Description**
CG_TC_TMS	7:0	0x0	DEF = 0x0 8 consecutive values for TMS. Bit 0 is sent first.
CG_TC_TDI	15:8	0x0	DEF = 0x0 8 consecutive values for TDI. Bit 0 is sent first.
CG_TC_MODE	16	0x0	0=Disabled 1=CG JTAG mode Enabled Indicates what clock should be used for TCK in the JTAG transactions.
CG_TC_TDO_MASK	31:24	0x0	DEF = 0x0 A mask indicating whether the TDO value should be read back for a given JTAG cycle. Bit 0 corresponds to the first TDO sample. This mask can be used to prevent the readback of unknown values across the bus interface during simulation. This field can be set to all 1's on real hardware.

CG Interface to the Test Controller (TC) using IEEE JTAG protocol. This register can be written with 8 consecutive values for the inputs to the TC's JTAG port. These 8 inputs are sent at consecutive TCK clock edges. The final value is held for indefinitely many TCK clock edges until the next write to this register. The register can be used to walk through several states of the JTAG state machine and typically the state machine would be left in a 'paused' state. The TDO values sampled at the 8 edges for which input was provided is available for readback from the TC_CG_TDO field of the CG_TC_JTAG_1 register.

CG_TC_JTAG_1 - R - 32 bits - [GpuF0MMReg:0x7A4]			
Field Name	**Bits**	**Default**	**Description**
TC_CG_TDO	7:0	0x0	8 consecutive sampled values of TDO. Bit 0 corresponds to the cycle that the first bit of CG_TC_JTAG_0.CG_TC_TMS and CG_TC_JTAG_0.CG_TC_TDI were sampled by the Test Controller.
TC_CG_DONE	31	0x0	0=We have completed less than 8 JTAG cycles since the last write to CG_TC_JTAG_0 1=All 8 JTAG cycles have been completed since the last write to CG_TC_JTAG_0 Indicates whether the JTAG sequence has completed.

TDO readback and status bits for the CG JTAG interface described in more detail in the CG_TC_JTAG_0 register description.

CG_MISC_REG - RW - 32 bits - [GpuF0MMReg:0x7C8]			
Field Name	**Bits**	**Default**	**Description**
SYNCHRONIZER_COUNTER	31:28	0x0	delay for restarting the clock while switching from one clock to another

Miscellaneous Register

CG_SPLL_SPREAD_SPECTRUM_LOW - RW - 32 bits - [GpuF0MMReg:0x820]			
Field Name	**Bits**	**Default**	**Description**
SSEN	0	0x0	1=Spread Spectrum enable
DITHEN	1	0x0	1=Enable Fractional Accumulation
BWADJ_EQ_CLKF	2	0x1	1=Loop BW control same as CLKF
CLKS	10:3	0x0	
CLKV	21:11	0x0	
BWADJ	31:22	0x0	

CG_SPLL_SPREAD_SPECTRUM_MED - RW - 32 bits - [GpuF0MMReg:0x824]			
Field Name	**Bits**	**Default**	**Description**
SSEN	0	0x0	1=Spread Spectrum enable
DITHEN	1	0x0	1=Enable Fractional Accumulation
BWADJ_EQ_CLKF	2	0x1	1=Loop BW control same as CLKF
CLKS	10:3	0x0	
CLKV	21:11	0x0	
BWADJ	31:22	0x0	

CG_SPLL_SPREAD_SPECTRUM_HIGH - RW - 32 bits - [GpuF0MMReg:0x828]			
Field Name	**Bits**	**Default**	**Description**
SSEN	0	0x0	1=Spread Spectrum enable
DITHEN	1	0x0	1=Enable Fractional Accumulation

BWADJ_EQ_CLKF	2	0x1	1=Loop BW control same as CLKF
CLKS	10:3	0x0	
CLKV	21:11	0x0	
BWADJ	31:22	0x0	

CG_SPLL_SPREAD_SPECTRUM_CTXSW - RW - 32 bits - [GpuF0MMReg:0x82C]			
Field Name	**Bits**	**Default**	**Description**
SSEN	0	0x0	1=Spread Spectrum enable
DITHEN	1	0x0	1=Enable Fractional Accumulation
BWADJ_EQ_CLKF	2	0x1	1=Loop BW control same as CLKF
CLKS	10:3	0x0	
CLKV	21:11	0x0	
BWADJ	31:22	0x0	

CG_MPLL_SPREAD_SPECTRUM - RW - 32 bits - [GpuF0MMReg:0x830]			
Field Name	**Bits**	**Default**	**Description**
SSEN	0	0x0	1=Spread Spectrum enable
DITHEN	1	0x0	1=Enable Fractional Accumulation
BWADJ_EQ_CLKF	2	0x1	1=Loop BW control same as CLKF
CLKS	10:3	0x0	
CLKV	21:11	0x0	
BWADJ	31:22	0x0	

CG_UPLL_SPREAD_SPECTRUM - RW - 32 bits - [GpuF0MMReg:0x834]			
Field Name	**Bits**	**Default**	**Description**
SSEN	0	0x0	1=Spread Spectrum enable
DITHEN	1	0x0	1=Enable Fractional Accumulation
BWADJ_EQ_CLKF	2	0x1	1=Loop BW control same as CLKF
CLKS	10:3	0x0	
CLKV	21:11	0x0	
BWADJ	31:22	0x0	

2.5 VIP/I2C Registers

2.5.1 I2C Registers

I2C_CNTL_0 - RW - 32 bits - [GpuF0MMReg:0xBC0]			
Field Name	**Bits**	**Default**	**Description**
I2C_DONE	0	0x0	Read only. Indicate whether current I2C request is finished or not 0=I2c is busy 1=transfer is complete
I2C_NACK	1	0x0	Read only. Status bit indicate whether I2C slave did not acknowledge. 1=Slave did not issue acknowledge
I2C_HALT	2	0x0	Read only. Status bit indicate where I2C bus transfer is time out. 1=Time-out condition, transfer is halted
I2C_SOFT_RST	5	0x0	Software reset I2C interface block 0=Normal 1=Resets i2c controller
I2C_DRIVE_EN	6	0x0	Enable I2C pad driving pull-up action 0=Pullup by external resistor 1=I2C pads drive SDA
I2C_DRIVE_SEL	7	0x0	If DRIVE_EN is HIGH, select drive time 0=Drive for 10MCLKs 1=20MCLKS
I2C_START	8	0x0	Indicate whether use the start condition in I2C protocol. 0=No start 1=Start
I2C_STOP	9	0x0	Indicate whether use the stop condition in I2C protocol. 0=No stop 1=Stop
I2C_RECEIVE	10	0x0	Master receive/transmit mode selection 0=Send 1=Receive
I2C_ABORT	11	0x0	If 1, abort the current I2C operation by sending STOP bit. 0=No abort 1=Abort
I2C_GO	12	0x0	Write this bit initiate I2C operation. Read this bit indicate the I2C operation is finished or not.
I2C_PRESCALE	31:16	0x0	I2C clock divider to generate I2C SCL output. It also indirectly control the sampling rate.

I2C control registers

I2C_CNTL_1 - RW - 32 bits - [GpuF0MMReg:0xBC4]			
Field Name	**Bits**	**Default**	**Description**
I2C_DATA_COUNT	3:0	0x0	Byte count for data to be transferred through I2C interface. The data should be in the 16 bytes I2C buffer
I2C_ADDR_COUNT	6:4	0x0	Byte count for I2C addresses. Maximum 3 bytes of address can be transferred.
I2C_INTRA_BYTE_DELAY	15:8	0x0	
I2C_SEL	16	0x0	Not used in Rage5 0=Pullup by external resistor 1=I2C pads drive SCL
I2C_EN	17	0x0	Enable I2C

I2C_TIME_LIMIT	31:24	0x0	Time out limit. Total wait time = TIME_LIMIT * 4 * PRESCLAE(15:8) cycles for SCL to be LOW

I2C control registers

I2C_DATA - RW - 32 bits - [GpuF0MMReg:0xBC8]			
Field Name	**Bits**	**Default**	**Description**
I2C_DATA	7:0	0x0	I2C data interface. Programmers use this 8bits interface to write and read I2C bus data.

I2C data registers. Programmers use this 8bits interface to write and read I2C bus data.

DC_I2C_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D30]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_GO (W)	0	0x0	Write 1 to start I2C transfer.
DC_I2C_SOFT_RESET	1	0x0	Write 1 to reset I2C controller
DC_I2C_SEND_RESET	2	0x0	Set to 1 to send reset sequence (9 clocks with no data) at start of transfer. This sequence is sent after DC_I2C_GO is written to 1, before the first transaction only.
DC_I2C_SW_STATUS_RESET	3	0x0	Write 1 to reset DC_I2C_SW_STATUS flags, will reset SW_DONE, ABORTED, TIMEOUT, SW_INTERRUPTED, BUFFER_OVERFLOW, STOPPED_ON_NACK, NACK0, NACK1, NACK2, NACK3
DC_I2C_SDVO_EN	4	0x0	Set to 1 to send two transactions to configure SDVO bus for DDC before main transaction. The SDVO transaction is as follows: S-AAw-a-07-a-02-a-P-S-AAw-a-08-a-7A-a-P where AA is the address and is selected by DC_I2C_SDVO_ADDR_SEL. The SDVO transactions take place after the RESET transaction (if enabled) and before the remaining transactions. 0=Disable 1=Enable
DC_I2C_SDVO_ADDR_SEL	6	0x0	Use to select address for SDVO I2C bus configuration 0=0x70 1=0x72
DC_I2C_DDC_SELECT	10:8	0x0	Select DDC pins set, dddc1, ddc2, ddc3 0=0 = DDC1 1=1 = DDC2 2=2 = DDC3 3=3-7 = Reserved
DC_I2C_TRANSACTION_COUNT	21:20	0x0	Number of transactions to be done in current transfer. 0=transaction0 only 1=transaction0, transaction1 2=transaction0, transaction1, transaction2 3=transaction0, transaction1, transaction2, transaction3 (DC_I2C_REPEAT=0 only)

DC_I2C_ARBITRATION - RW - 32 bits - [GpuF0MMReg:0x7D34]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_SW_PRIORITY	1:0	0x1	Sets priority for software I2C requests. This setting applies only when HDCP is using I2C bus and software also wants to use the same I2C bus 0=Normal - If DC_I2C_NO_QUEUED_SW_GO = 0, software I2C transaction will be queued after HW I2C. If DC_I2C_NO_QUEUED_SW_GO = 1, software I2C transaction is not queued, in this case, software have to poll for DC_I2C_DDCx_HW_DONE doing any I2C transaction 1=High - Software always interrupts HW I2C if HDCP is using the same I2C bus, HW I2C will automatically resume once software I2C is completed 2=Reserved 3=Reserved
DC_I2C_NO_QUEUED_SW_GO	4	0x0	Set to 1 to disable queuing of software I2C GO. If this bit is set, then if software writes DC_I2C_GO while I2C is in use by hardware, the GO request will be ignored and the DC_I2C_SW_INTERRUPTED bit set.
DC_I2C_NO_RESTART_SW_GO	5	0x0	Set to 1 to disable restart of software I2C transaction that was interrupted by hardware. Typically this bit should be 0, unless there is a problem with the I2C restart mechanism. When this bit is set to 0, the DC_I2C_SW_DONE bit will not be set if hardware interrupts the software transfer.
DC_I2C_ABORT_HW_XFER (W)	8	0x0	Write 1 to abort current HW transfer (send stop if transfer has started)
DC_I2C_ABORT_SW_XFER (W)	12	0x0	Write 1 to abort current SW transfer (send stop if transfer has started)

Configure arbitration between hardware and software use of the DC_I2C engine

DC_I2C_INTERRUPT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D38]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_SW_DONE_INT (R)	0	0x0	SW_DONE interrupt status
DC_I2C_SW_DONE_ACK (W)	1	0x0	Acknowledge bit for DC_I2C_SW_DONE_INT. Write 1 to clear interrupt.
DC_I2C_SW_DONE_MASK	2	0x0	Mask bit for DC_I2C_SW_DONE_INT. Set to 1 to enable interrupt.
DC_I2C_DDC1_HW_DONE_INT (R)	4	0x0	DDC1 HW_DONE interrupt status
DC_I2C_DDC1_HW_DONE_ACK (W)	5	0x0	DDC1 Acknowledge bit for DC_I2C_HW_DDC1_DONE_INT. Write 1 to clear interrupt.
DC_I2C_DDC1_HW_DONE_MASK	6	0x0	DDC1 Mask bit for DC_I2C_HW_DDC1_DONE_INT. Set to 1 to enable interrupt.
DC_I2C_DDC2_HW_DONE_INT (R)	8	0x0	DDC2 HW_DONE interrupt status
DC_I2C_DDC2_HW_DONE_ACK (W)	9	0x0	DDC2 Acknowledge bit for DC_I2C_HW_DDC2_DONE_INT. Write 1 to clear interrupt.
DC_I2C_DDC2_HW_DONE_MASK	10	0x0	DDC2 Mask bit for DC_I2C_HW_DDC2_DONE_INT. Set to 1 to enable interrupt.
DC_I2C_DDC3_HW_DONE_INT (R)	12	0x0	DDC3 HW_DONE interrupt status
DC_I2C_DDC3_HW_DONE_ACK (W)	13	0x0	DDC3 Acknowledge bit for DC_I2C_HW_DDC3_DONE_INT. Write 1 to clear interrupt.
DC_I2C_DDC3_HW_DONE_MASK	14	0x0	DDC3 Mask bit for DC_I2C_HW_DDC3_DONE_INT. Set to 1 to enable interrupt.
DC_I2C_DDC4_HW_DONE_INT (R)	16	0x0	
DC_I2C_DDC4_HW_DONE_ACK (W)	17	0x0	
DC_I2C_DDC4_HW_DONE_MASK	18	0x0	

DC_I2C_SW_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D3C]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_SW_STATUS (R)	1:0	0x0	Current SW status of DC_I2C 0=Idle 1=In use by SW 2=In use by HW 3=Reserved
DC_I2C_SW_DONE (R)	2	0x0	Set on completion of SW transfer. Cleared by writing DC_I2C_SW_DONE_ACK to 1
DC_I2C_SW_ABORTED (R)	4	0x0	Indicates that abort request ccurred during SW transfer, stopping transfer. Cleared on GO.
DC_I2C_SW_TIMEOUT (R)	5	0x0	Indicates that timeout condition occurred during SW transfer, stopping transfer. Cleared on GO.
DC_I2C_SW_INTERRUPTED (R)	6	0x0	Indicates that SW transfer was interrupted by hardware request. Cleared on GO.
DC_I2C_SW_BUFFER_OVERFLOW (R)	7	0x0	Indicates that buffer overflow occurred during SW transfer, stopping transfer. Cleared on GO.
DC_I2C_SW_STOPPED_ON_NACK (R)	8	0x0	Indicates that SW transfer was interrpted due to NACK when STOP_ON_NACK=1. Cleared on GO.
DC_I2C_SW_SDVO_NACK (R)	10	0x0	
DC_I2C_SW_NACK0 (R)	12	0x0	Indicates that I2C slave did not issue an acknowledge during the first SW transaction. Cleared on GO.
DC_I2C_SW_NACK1 (R)	13	0x0	Indicates that I2C slave did not issue an acknowledge during the second SW transaction. Cleared on GO.
DC_I2C_SW_NACK2 (R)	14	0x0	Indicates that I2C slave did not issue an acknowledge during the third SW transaction. Cleared on GO.
DC_I2C_SW_NACK3 (R)	15	0x0	Indicates that I2C slave did not issue an acknowledge during the fourth SW transaction. Cleared on GO.
DC_I2C_SW_REQ (R)	18	0x0	Software requests use of DC_I2C interface (indicates that request is pending - i.e. queued). Cleared when request becomes active or by DC_I2C_ABORT_SW_XFER.

Status fields for DC_I2C engine

DC_I2C_DDC1_HW_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D40]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC1_HW_STATUS (R)	1:0	0x0	Current HW status of DC_I2C 0=Idle 1=In use by SW 2=In use by HW 3=Reserved
DC_I2C_DDC1_HW_DONE (R)	3	0x0	Set on completion of HW transfer. Cleared by writing DC_I2C_HW_DONE_ACK to 1
DC_I2C_DDC1_HW_REQ (R)	16	0x0	Hardware requests use of DC_I2C interface (indicates that request is pending - i.e. queued). Cleared when request becomes active or by DC_I2C_ABORT_HW_XFER.
DC_I2C_DDC1_HW_URG (R)	17	0x0	Indicates that hardware I2C request is urgent (used by arbitration logic).

Status fields for DC_I2C engine

DC_I2C_DDC2_HW_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D44]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC2_HW_STATUS (R)	1:0	0x0	Current HW status of DC_I2C 0=Idle 1=In use by SW 2=In use by HW 3=Reserved
DC_I2C_DDC2_HW_DONE (R)	3	0x0	Set on completion of HW transfer. Cleared by writing DC_I2C_HW_DONE_ACK to 1
DC_I2C_DDC2_HW_REQ (R)	16	0x0	Hardware requests use of DC_I2C interface (indicates that request is pending - i.e. queued). Cleared when request becomes active or by DC_I2C_ABORT_HW_XFER.
DC_I2C_DDC2_HW_URG (R)	17	0x0	Indicates that hardware I2C request is urgent (used by arbitration logic).

Status fields for DC_I2C engine

DC_I2C_DDC3_HW_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D48]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC3_HW_STATUS (R)	1:0	0x0	Current HW status of DC_I2C 0=Idle 1=In use by SW 2=In use by HW 3=Reserved
DC_I2C_DDC3_HW_DONE (R)	3	0x0	Set on completion of HW transfer. Cleared by writing DC_I2C_HW_DONE_ACK to 1
DC_I2C_DDC3_HW_REQ (R)	16	0x0	Hardware requests use of DC_I2C interface (indicates that request is pending - i.e. queued). Cleared when request becomes active or by DC_I2C_ABORT_HW_XFER.
DC_I2C_DDC3_HW_URG (R)	17	0x0	Indicates that hardware I2C request is urgent (used by arbitration logic).

Status fields for DC_I2C engine

DC_I2C_DDC1_SPEED - RW - 32 bits - [GpuF0MMReg:0x7D4C]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC1_THRESHOLD	1:0	0x2	Select threshold to use to determine whether value sampled on SDA is a 1 or 0. Specified in terms of the ratio between the number of sampled ones and the total number of times SDA is sampled. 0=>0 1=1/4 of total samples 2=1/2 of total samples 3=3/4 of total samples
DC_I2C_DDC1_PRESCALE	31:16	0x0	prescale = (m * xtal_frequency) / (desired_i2c_speed), where m is multiply factor, default: m = 1

DDC1 speed setting

DC_I2C_DDC1_SETUP - RW - 32 bits - [GpuF0MMReg:0x7D50]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC1_DATA_DRIVE_EN	0	0x0	Select whether SDA pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SDA high 0=Pullup by external resistor 1=I2C pads drive SDA
DC_I2C_DDC1_DATA_DRIVE_SEL	1	0x0	Select number of clocks to drive SDA high 0:Drive for 10 SCLKs 1:Drive for 20 SCLKs 0=Drive for 10MCLKs 1=20MCLKS
DC_I2C_DDC1_CLK_DRIVE_EN	7	0x0	Select whether SCL pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SCL high 0=Pullup by external resistor 1=I2C pads drive SCL
DC_I2C_DDC1_INTRA_BYTE_DELAY	15:8	0x0	Use to specify delay between bytes in units of I2C reference.
DC_I2C_DDC1_INTRA_TRANSACTION_DELAY	23:16	0x0	Use to specify delay between transactions in units of I2C reference.
DC_I2C_DDC1_TIME_LIMIT	31:24	0x0	Time limit, in units of 256 I2C fast reference (TOCLK) pulses, to wait before timeout when clock is stalled by external device.

DDC1 SETUP

DC_I2C_DDC2_SPEED - RW - 32 bits - [GpuF0MMReg:0x7D54]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC2_THRESHOLD	1:0	0x2	Select threshold to use to determine whether value sampled on SDA is a 1 or 0. Specified in terms of the ratio between the number of sampled ones and the total number of times SDA is sampled. 0=>0 1=1/4 of total samples 2=1/2 of total samples 3=3/4 of total samples
DC_I2C_DDC2_PRESCALE	31:16	0x0	prescale = (m * xtal_frequency) / (desired_i2c_speed), where m is multiply factor, default: m = 1

DDC2 speed setting

DC_I2C_DDC2_SETUP - RW - 32 bits - [GpuF0MMReg:0x7D58]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC2_DATA_DRIVE_EN	0	0x0	Select whether SDA pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SDA high 0=Pullup by external resistor 1=I2C pads drive SDA
DC_I2C_DDC2_DATA_DRIVE_SEL	1	0x0	Select number of clocks to drive SDA high 0:Drive for 10 SCLKs 1:Drive for 20 SCLKs 0=Drive for 10MCLKs 1=20MCLKS

DC_I2C_DDC2_CLK_DRIVE_EN	7	0x0	Select whether SCL pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SCL high 0=Pullup by external resistor 1=I2C pads drive SCL
DC_I2C_DDC2_INTRA_BYTE_DELAY	15:8	0x0	Use to specify delay between bytes in units of I2C reference.
DC_I2C_DDC2_INTRA_TRANSACTION_DELAY	23:16	0x0	Use to specify delay between transactions in units of I2C reference.
DC_I2C_DDC2_TIME_LIMIT	31:24	0x0	Time limit, in units of 256 I2C fast reference (TOCLK) pulses, to wait before timeout when clock is stalled by external device.

DDC2 SETUP

DC_I2C_DDC3_SPEED - RW - 32 bits - [GpuF0MMReg:0x7D5C]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC3_THRESHOLD	1:0	0x2	Select threshold to use to determine whether value sampled on SDA is a 1 or 0. Specified in terms of the ratio between the number of sampled ones and the total number of times SDA is sampled. 0=>0 1=1/4 of total samples 2=1/2 of total samples 3=3/4 of total samples
DC_I2C_DDC3_PRESCALE	31:16	0x0	prescale = (m * xtal_frequency) / (desired_i2c_speed), where m is multiply factor, default: m = 1

DDC2 speed setting

DC_I2C_DDC3_SETUP - RW - 32 bits - [GpuF0MMReg:0x7D60]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC3_DATA_DRIVE_EN	0	0x0	Select whether SDA pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SDA high 0=Pullup by external resistor 1=I2C pads drive SDA
DC_I2C_DDC3_DATA_DRIVE_SEL	1	0x0	Select number of clocks to drive SDA high 0:Drive for 10 SCLKs 1:Drive for 20 SCLKs 0=Drive for 10MCLKs 1=20MCLKS
DC_I2C_DDC3_CLK_DRIVE_EN	7	0x0	Select whether SCL pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SCL high 0=Pullup by external resistor 1=I2C pads drive SCL
DC_I2C_DDC3_INTRA_BYTE_DELAY	15:8	0x0	Use to specify delay between bytes in units of I2C reference.
DC_I2C_DDC3_INTRA_TRANSACTION_DELAY	23:16	0x0	Use to specify delay between transactions in units of I2C reference.
DC_I2C_DDC3_TIME_LIMIT	31:24	0x0	Time limit, in units of 256 I2C fast reference (TOCLK) pulses, to wait before timeout when clock is stalled by external device.

DDC3 SETUP

DC_I2C_TRANSACTION0 - RW - 32 bits - [GpuF0MMReg:0x7D64]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_RW0	0	0x0	Read/write indicator for first transaction - set to 0 for write, 1 for read. This bit only controls DC_I2C behaviour - the R/W bit in the transaction is programmed into the I2C buffer as the LSB of the address byte. 0=WRITE 1=READ
DC_I2C_STOP_ON_NACK0	8	0x0	Determines whether the current transfer will stop if a NACK is received during the first transaction (current transaction always stops). 0=STOP CURRENT TRANSACTION, GO TO NEXT TRANSACTION 1=STOP ALL TRANSACTIONS, SEND STOP BIT
DC_I2C_ACK_ON_READ0	9	0x0	Determines whether hardware will send an ACK after the last byte on a read in the first transaction. 0=Send NACK 1=Send ACK
DC_I2C_START0	12	0x0	Determines whether a start bit will be sent before the first transaction 0=NO START 1=START
DC_I2C_STOP0	13	0x0	Determines whether a stop bit will be sent after the first transaction 0=NO STOP 1=STOP
DC_I2C_COUNT0	23:16	0x0	Byte count for first transaction (excluding the first byte, which is usually the address).

Configuration for first transaction

DC_I2C_TRANSACTION1 - RW - 32 bits - [GpuF0MMReg:0x7D68]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_RW1	0	0x0	Read/write indicator for second transaction - set to 0 for write, 1 for read. This bit only controls DC_I2C behaviour - the R/W bit in the transaction is programmed into the I2C buffer as the LSB of the address byte. 0=WRITE 1=READ
DC_I2C_STOP_ON_NACK1	8	0x0	Determines whether the current transfer will stop if a NACK is received during the second transaction (current transaction always stops). 0=STOP CURRENT TRANSACTION, GO TO NEXT TRANSACTION 1=STOP ALL TRANSACTIONS, SEND STOP BIT
DC_I2C_ACK_ON_READ1	9	0x0	Determines whether hardware will send an ACK after the last byte on a read in the second transaction. 0=Send NACK 1=Send ACK
DC_I2C_START1	12	0x0	Determines whether a start bit will be sent before the second transaction 0=NO START 1=START

DC_I2C_STOP1	13	0x0	Determines whether a stop bit will be sent after the second transaction 0=NO STOP 1=STOP
DC_I2C_COUNT1	23:16	0x0	Byte count for second transaction (excluding the first byte, which is usually the address).

Configuration for second transaction

DC_I2C_TRANSACTION2 - RW - 32 bits - [GpuF0MMReg:0x7D6C]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_RW2	0	0x0	Read/write indicator for third transaction - set to 0 for write, 1 for read. This bit only controls DC_I2C behaviour - the R/W bit in the transaction is programmed into the I2C buffer as the LSB of the address byte. 0=WRITE 1=READ
DC_I2C_STOP_ON_NACK2	8	0x0	Determines whether the current transfer will stop if a NACK is received during the third transaction (current transaction always stops). 0=STOP CURRENT TRANSACTION, GO TO NEXT TRANSACTION 1=STOP ALL TRANSACTIONS, SEND STOP BIT
DC_I2C_ACK_ON_READ2	9	0x0	Determines whether hardware will send an ACK after the last byte on a read in the third transaction. 0=Send NACK 1=Send ACK
DC_I2C_START2	12	0x0	Determines whether a start bit will be sent before the third transaction 0=NO START 1=START
DC_I2C_STOP2	13	0x0	Determines whether a stop bit will be sent after the third transaction 0=NO STOP 1=STOP
DC_I2C_COUNT2	23:16	0x0	Byte count for third transaction (excluding the first byte, which is usually the address).

Configuration for third transaction

DC_I2C_TRANSACTION3 - RW - 32 bits - [GpuF0MMReg:0x7D70]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_RW3	0	0x0	Read/write indicator for fourth transaction - set to 0 for write, 1 for read. This bit only controls DC_I2C behaviour - the R/W bit in the transaction is programmed into the I2C buffer as the LSB of the address byte. 0=WRITE 1=READ
DC_I2C_STOP_ON_NACK3	8	0x0	Determines whether the current transfer will stop if a NACK is received during the fourth transaction (current transaction always stops). 0=STOP CURRENT TRANSACTION, GO TO NEXT TRANSACTION 1=STOP ALL TRANSACTIONS, SEND STOP BIT

DC_I2C_ACK_ON_READ3	9	0x0	Determines whether hardware will send an ACK after the last byte on a read in the fourth transaction. 0=Send NACK 1=Send ACK
DC_I2C_START3	12	0x0	Determines whether a start bit will be sent before the fourth transaction 0=NO START 1=START
DC_I2C_STOP3	13	0x0	Determines whether a stop bit will be sent after the fourth transaction 0=NO STOP 1=STOP
DC_I2C_COUNT3	23:16	0x0	Byte count for fourth transaction (excluding the first byte, which is usually the address).

Configuration for fourth transaction

DC_I2C_DATA - RW - 32 bits - [GpuF0MMReg:0x7D74]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DATA_RW	0	0x0	Select whether buffer access will be a read or write. For writes, address auto-increments on write to DC_I2C_DATA. For reads, address auto-increments on reads to DC_I2C_DATA. 0=Write 1=Read
DC_I2C_DATA	15:8	0x0	Use to fill or read the I2C buffer
DC_I2C_INDEX	23:16	0x0	Use to set index into I2C buffer for next read or current write, or to read index of current read or next write. Writable only when DC_I2C_INDEX_WRITE=1.
DC_I2C_INDEX_WRITE (W)	31	0x0	To write index field, set this bit to 1 while writing DC_I2C_DATA.

This register is used to read or write the I2C buffer

GENERIC_I2C_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D80]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_GO (W)	0	0x0	Write 1 to start I2C transfer
GENERIC_I2C_SOFT_RESET	1	0x0	Write 1 to reset I2C controller
GENERIC_I2C_SEND_RESET	2	0x0	Set to 1 to send reset sequence (9 clocks with no data) at start of transfer. This sequence is sent after DC_I2C_GO is written to 1

generic i2c control register

GENERIC_I2C_INTERRUPT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D84]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_DONE_INT (R)	0	0x0	GENERIC_I2C_DONE interrupt status
GENERIC_I2C_DONE_ACK (W)	1	0x0	Acknowledge bit for GENERIC_I2C_DONE. Write 1 to clear interrupt.
GENERIC_I2C_DONE_MASK	2	0x0	Mask bit for GENERIC_I2C_DONE. Set to 1 to enable interrupt.

generic i2c interrupt control register

GENERIC_I2C_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D88]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_STATUS (R)	3:0	0x0	Status of the i2c internal state code: 0: idle, 1: sending start, 2: sending address, 3: transmitting/receiving data, 4: transmitting/receiving ack 5: sending stop, 6:N/A, 7:byte delay, 8: wait for GO command
GENERIC_I2C_DONE (R)	4	0x0	Indicates the completion of i2c transfer. Cleared by writing GENERIC_I2C_DONE_ACK or GO
GENERIC_I2C_ABORTED (R)	5	0x0	Indicates that abort request ccurred during i2c transfer, stopping transfer. Cleared on GO.
GENERIC_I2C_TIMEOUT (R)	6	0x0	Indicates that timeout condition occurred during SW transfer, stopping transfer. Cleared on GO.
GENERIC_I2C_STOPPED_ON_NACK (R)	9	0x0	Indicates that SW transfer was interrpted due to NACK when STOP_ON_NACK=1. Cleared on GO.
GENERIC_I2C_NACK (R)	10	0x0	Indicates that I2C slave did not issue an acknowledge during the i2c transaction. Cleared on GO.

generic i2c read only status register

GENERIC_I2C_SPEED - RW - 32 bits - [GpuF0MMReg:0x7D8C]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_THRESHOLD	1:0	0x2	Select threshold to use to determine whether value sampled on SDA is a 1 or 0 when SCL is hi. 0: begining of SCL(hi), 1: 1/4 of SCL(hi),2: 1/2 of SCL(hi),3: 3/4 of SCL(hi) 0=>0 1=1/4 of total samples 2=1/2 of total samples 3=3/4 of total samples
GENERIC_I2C_PRESCALE	31:16	0x0	prescale = (m * xtal_frequency) / (4 * desired_i2c_speed), where m is multiply factor, default: m = 1

Generic i2c bus config

GENERIC_I2C_SETUP - RW - 32 bits - [GpuF0MMReg:0x7D90]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_DATA_DRIVE_EN	0	0x0	Select whether SDA pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SDA high 0=Pullup by external resistor 1=I2C pads drive SDA
GENERIC_I2C_DATA_DRIVE_SEL	1	0x0	Select number of clocks to drive SDA high 0:Drive for 10 SCLKs 1:Drive for 20 SCLKs 0=Drive for 10MCLKs 1=20MCLKS

GENERIC_I2C_CLK_DRIVE_EN	7	0x0	Select whether SCL pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SCL high 0=Pullup by external resistor 1=I2C pads drive SCL
GENERIC_I2C_INTRA_BYTE_DELAY	15:8	0x0	Use to specify delay between bytes in units of I2C reference.
GENERIC_I2C_TIME_LIMIT	31:24	0x0	Time limit, in units of 256 I2C fast reference (TOCLK) pulses, to wait before timeout when clock is stalled by external device.

Generic i2c bus config

GENERIC_I2C_TRANSACTION - RW - 32 bits - [GpuF0MMReg:0x7D94]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_RW	0	0x0	Read/write indicator for second transaction - set to 0 for write, 1 for read. This bit only controls DC_I2C behaviour - the R/W bit in the transaction is programmed into the I2C buffer as the LSB of the address byte. 0=WRITE 1=READ
GENERIC_I2C_STOP_ON_NACK	8	0x0	Determines whether the current transfer will stop if a NACK is received during the transaction (current transaction always stops). 0=STOP CURRENT TRANSACTION, GO TO NEXT TRANSACTION 1=STOP ALL TRANSACTIONS, SEND STOP BIT
GENERIC_I2C_ACK_ON_READ	9	0x0	Determines whether hardware will send an ACK after the last byte on a read in the second transaction. 0=Send NACK 1=Send ACK
GENERIC_I2C_START	12	0x0	Determines whether a start bit will be sent before the second transaction 0=NO START 1=START
GENERIC_I2C_STOP	13	0x0	Determines whether a stop bit will be sent after the second transaction 0=NO STOP 1=STOP
GENERIC_I2C_COUNT	19:16	0x0	Byte count for the transaction (excluding the first byte, which is usually the address).

generic i2c transaction setup register

GENERIC_I2C_DATA - RW - 32 bits - [GpuF0MMReg:0x7D98]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_DATA_RW	0	0x0	Select whether buffer access will be a read or write. For writes, address auto-increments on write to DC_I2C_DATA. For reads, address auto-increments on reads to GENERIC_I2C_DATA. 0=Write 1=Read
GENERIC_I2C_DATA	15:8	0x0	Use to fill or read the generic I2C buffer

GENERIC_I2C_INDEX	19:16	0x0	Use to set index into I2C buffer for next read or current write, or to read index of current read or next write. Writable only when GENERIC_I2C_INDEX_WRITE=1.
GENERIC_I2C_INDEX_WRITE (W)	31	0x0	To write index field, set this bit to 1 while writing GENERIC_I2C_DATA

This register is used to read or write the I2C buffer

GENERIC_I2C_PIN_SELECTION - RW - 32 bits - [GpuF0MMReg:0x7D9C]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_SCL_PIN_SEL	6:0	0x0	GPIO pin selection to use for SCL, if GENERIC_I2C_SCL_PIN_SEL == GENERIC_I2C_SDA_PIN_SEL => disable pin selectin. Refer to generic_i2c_programming guide for pin selection details
GENERIC_I2C_SDA_PIN_SEL	14:8	0x0	GPIO pin selection to use for SDA, if GENERIC_I2C_SCL_PIN_SEL == GENERIC_I2C_SDA_PIN_SEL => disable pin selectin. Refer to generic_i2c_programming guide for pin selection details

Pin selection register

GENERIC_I2C_PIN_DEBUG - RW - 32 bits - [GpuF0MMReg:0x7DA0]			
Field Name	**Bits**	**Default**	**Description**
GENERIC_I2C_SCL_OUTPUT	0	0x0	SCL pin output value when GENERIC_I2C_SCL_EN is set
GENERIC_I2C_SCL_INPUT (R)	1	0x0	SCL pin input value when SCL pin is not driving, i.e. GENERIC_I2C_SCL_EN = 0
GENERIC_I2C_SCL_EN	2	0x0	SCL tri-state output control, set to one when SCL needs to drive
GENERIC_I2C_SDA_OUTPUT	4	0x0	SDA pin output value when GENERIC_I2C_SDA_EN is set
GENERIC_I2C_SDA_INPUT (R)	5	0x0	SDA pin input value when SDA pin is not driving, i.e. GENERIC_I2C_SCL_EN = 0
GENERIC_I2C_SDA_EN	6	0x0	SCL tri-state output control, set to one when SCL needs to drive

Generic i2c pin debug register, allow software to control the selected pins directly

DC_I2C_DDC4_HW_STATUS - RW - 32 bits - [GpuF0MMReg:0x7DB0]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC4_HW_STATUS (R)	1:0	0x0	Current HW status of DC_I2C 0=Idle 1=In use by HW 2=In use by HW 3=Reserved
DC_I2C_DDC4_HW_DONE (R)	3	0x0	Set on completion of HW transfer. Cleared by writing DC_I2C_HW_DONE_ACK to 1
DC_I2C_DDC4_HW_REQ (R)	16	0x0	Hardware requests use of DC_I2C interface (indicates that request is pending - i.e. queued). Cleared when request becomes active or by DC_I2C_ABORT_HW_XFER.

DC_I2C_DDC4_HW_URG (R)	17	0x0	Indicates that hardware I2C request is urgent (used by arbitration logic).

Status fields for DC_I2C engine

DC_I2C_DDC4_SPEED - RW - 32 bits - [GpuF0MMReg:0x7DB4]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC4_THRESHOLD	1:0	0x2	Select threshold to use to determine whether value sampled on SDA is a 1 or 0. Specified in terms of the ratio between the number of sampled ones and the total number of times SDA is sampled. 0=>0 1=1/4 of total samples 2=1/2 of total samples 3=3/4 of total samples
DC_I2C_DDC4_PRESCALE	31:16	0x0	prescale = (m * xtal_frequency) / (desired_i2c_speed), where m is multiply factor, default: m = 1

DDC4 speed setting

DC_I2C_DDC4_SETUP - RW - 32 bits - [GpuF0MMReg:0x7DBC]			
Field Name	**Bits**	**Default**	**Description**
DC_I2C_DDC4_DATA_DRIVE_EN	0	0x0	Select whether SDA pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SDA high 0=Pullup by external resistor 1=I2C pads drive SDA
DC_I2C_DDC4_DATA_DRIVE_SEL	1	0x0	Select number of clocks to drive SDA high 0:Drive for 10 SCLKs 1:Drive for 20 SCLKs 0=Drive for 10MCLKs 1=20MCLKS
DC_I2C_DDC4_CLK_DRIVE_EN	7	0x0	Select whether SCL pad is pulled up or driven high 0:Pullup by external resistor 1:I2C pads drive SCL high 0=Pullup by external resistor 1=I2C pads drive SCL
DC_I2C_DDC4_INTRA_BYTE_DELAY	15:8	0x0	Use to specify delay between bytes in units of I2C reference.
DC_I2C_DDC4_INTRA_TRANSACTION_DELAY	23:16	0x0	Use to specify delay between transactions in units of I2C reference.
DC_I2C_DDC4_TIME_LIMIT	31:24	0x0	Time limit, in units of 256 I2C fast reference (TOCLK) pulses, to wait before timeout when clock is stalled by external device.

DDC4 SETUP

2.5.2 Video Interface Port Host Port Registers

VIPH_REG_ADDR - RW - 32 bits - [GpuF0MMReg:0xC80]			
Field Name	**Bits**	**Default**	**Description**

VIPH_REG_AD	15:0	0x0	Bits (11:0): Slave registers address. Bits(12): 0 = register access, 1= FIFO access. Bits(13): 0= register write, 1 = register read. Bits(15:14): Slave device ID.

VIP Host register access command and address.

VIPH_REG_DATA - RW - 32 bits - [GpuF0MMReg:0xC84]			
Field Name	**Bits**	**Default**	**Description**
VIPH_REG_DT_R (R)	31:0	0x0	Read from VIP Host Port register data port
VIPH_REG_DT_W (W)	31:0	0x0	Write to VIP Host Port register data port

VIP Host Port register data port

VIPH_CH0_DATA - RW - 32 bits - [GpuF0MMReg:0xC00]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH0_DT	31:0	0x0	VIPH0 data interface

VIPH0 data interface

VIPH_CH1_DATA - RW - 32 bits - [GpuF0MMReg:0xC04]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH1_DT	31:0	0x0	VIPH0 data interface

VIPH0 data interface

VIPH_CH2_DATA - RW - 32 bits - [GpuF0MMReg:0xC08]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH2_DT	31:0	0x0	VIPH0 data interface

VIPH0 data interface

VIPH_CH3_DATA - RW - 32 bits - [GpuF0MMReg:0xC0C]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH3_DT	31:0	0x0	VIPH0 data interface

VIPH0 data interface

VIPH_CH0_ADDR - RW - 32 bits - [GpuF0MMReg:0xC10]			
Field Name	**Bits**	**Default**	**Description**

VIPH_CH0_AD	7:0	0x0	Bit(3:0): FIFO address Bit(4): 0= register access, 1 = FIFO access. Bit(5): 0= register write, 1= register read. Bits(7:6): Slave device ID.

VIPH0 command + address.

VIPH_CH1_ADDR - RW - 32 bits - [GpuF0MMReg:0xC14]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH1_AD	7:0	0x0	Bit(3:0): FIFO address Bit(4): 0= register access, 1 = FIFO access. Bit(5): 0= register write, 1= register read. Bits(7:6): Slave device ID.

VIPH1 command + address.

VIPH_CH2_ADDR - RW - 32 bits - [GpuF0MMReg:0xC18]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH2_AD	7:0	0x0	Bit(3:0): FIFO address Bit(4): 0= register access, 1 = FIFO access. Bit(5): 0= register write, 1= register read. Bits(7:6): Slave device ID.

VIPH2 command + address.

VIPH_CH3_ADDR - RW - 32 bits - [GpuF0MMReg:0xC1C]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH3_AD	7:0	0x0	Bit(3:0): FIFO address Bit(4): 0= register access, 1 = FIFO access. Bit(5): 0= register write, 1= register read. Bits(7:6): Slave device ID.

VIPH3 command + address.

VIPH_CH0_SBCNT - RW - 32 bits - [GpuF0MMReg:0xC20]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH0_SCNT	19:0	0x0	Write non-zero byte count will trigger DMA. Maximum 2 jobs can be loaded into the queue any one time.

Byte count of transfer requested.

VIPH_CH1_SBCNT - RW - 32 bits - [GpuF0MMReg:0xC24]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH1_SCNT	19:0	0x0	Write non-zero byte count will trigger DMA. Maximum 2 jobs can be loaded into the queue any one time.

Byte count of transfer requested.

VIPH_CH2_SBCNT - RW - 32 bits - [GpuF0MMReg:0xC28]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH2_SCNT	19:0	0x0	Write non-zero byte count will trigger DMA. Maximum 2 jobs can be loaded into the queue any one time.

Byte count of transfer requested.

VIPH_CH3_SBCNT - RW - 32 bits - [GpuF0MMReg:0xC2C]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH3_SCNT	19:0	0x0	Write non-zero byte count will trigger DMA. Maximum 2 jobs can be loaded into the queue any one time.

Byte count of transfer requested.

VIPH_CH0_ABCNT - RW - 32 bits - [GpuF0MMReg:0xC30]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH0_ACNT (R)	19:0	0x0	Keep track of active byte-count remaining.

Read back of remaining byte count.

VIPH_CH1_ABCNT - RW - 32 bits - [GpuF0MMReg:0xC34]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH1_ACNT (R)	19:0	0x0	Keep track of active byte-count remaining.

Read back of remaining byte count.

VIPH_CH2_ABCNT - RW - 32 bits - [GpuF0MMReg:0xC38]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH2_ACNT (R)	19:0	0x0	Keep track of active byte-count remaining.

Read back of remaining byte count.

VIPH_CH3_ABCNT - RW - 32 bits - [GpuF0MMReg:0xC3C]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH3_ACNT (R)	19:0	0x0	Keep track of active byte-count remaining.

Read back of remaining byte count.

VIPH_CONTROL - RW - 32 bits - [GpuF0MMReg:0xC40]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CLK_SEL	7:0	0x0	VIPH clock select, only even divider is permitted. Which means VIPH_CLK_SEL(0) must be set to 1. 0=reserved 1=reserved 2=reserved 3=xclkby4 4=reserved 5=xclkby6 6=... (Only EVEN divider is permitted
VIPH_REG_RDY (R)	13	0x0	0= VIPH is ready for next register access. 1= VIPH is busy for current VIPH register access.
VIPH_MAX_WAIT	19:16	0x0	Number of VIP phases before issuing time out. Set to zero means no time out
VIPH_DMA_MODE	20	0x0	0= No DMA. 1= DMA
VIPH_EN	21	0x0	VIP Host port Enable
VIP_DEVICE_DESKTOP (R)	22	0x0	0=VIP_DEVICE present, 1=No VIP device attached, valid only with MOBILE_DIS=1 and VIP_DEVICE_STRAP_DIS=0
VIP_DEVICE_MOBILE (R)	23	0x0	0=VIP_DEVICE present, 1=No VIP device attached, valid only with MOBILE_DIS=0 and VIP_DEVICE_STRAP_DIS=0
VIPH_DV0_WID	24	0x0	VIPH0 bus width 0=2-bit vipbus 1=4-bit vipbus
VIPH_DV1_WID	25	0x0	VIPH1 bus width 0=2-bit vipbus 1=4-bit vipbus
VIPH_DV2_WID	26	0x0	VIPH2 bus width 0=2-bit vipbus 1=4-bit vipbus
VIPH_DV3_WID	27	0x0	VIPH3 bus width 0=2-bit vipbus 1=4-bit vipbus
VIPH_PWR_DOWN (R)	28	0x0	'1' to wake up PCICLK. 0=Normal 1=STARTUP PCICLK
VIPH_PWR_DOWN_AK (W)	28	0x0	Clear PWR_DOWN by writing a 1. In order to support PCICLK power down mode, it is important to clear this bit every time there is an interrupt from any part of VIP 0=Normal 1=Allow the host bus to go back to power down state
VIPH_VIPCLK_DIS	29	0x0	'0' will supply VIP clock to slave. '1' will stops VIP clock to save power. 0= 1=turn off VIPCLK for power saving
VIPH_INT_SEL	30	0x0	0=If VIP host port interrupt using input instead of polling, then AUXWIN pin used as interrupt input. 1=If VIP host port interrupt using input instead of polling, then I2C clock pin used as interrupt input.
VIP_DEVICE_STRAP_DIS (R)	31	0x0	0=VIP_DEVICE strap must be checked, 1=VIP_DEVICE strap must be ignored

VIP Host Port Control

VIPH_DV_LAT - RW - 32 bits - [GpuF0MMReg:0xC44]

Field Name	Bits	Default	Description
VIPH_TIME_UNIT	11:0	0x0	Basic time slice
VIPH_DV0_LAT	19:16	0x0	How many time slice port 0 gets
VIPH_DV1_LAT	23:20	0x0	How many time slice port 1 gets
VIPH_DV2_LAT	27:24	0x0	How many time slice port 2 gets
VIPH_DV3_LAT	31:28	0x0	How many time slice port 3 gets

Time slice partition

VIPH_DMA_CHUNK - RW - 32 bits - [GpuF0MMReg:0xC48]			
Field Name	**Bits**	**Default**	**Description**
VIPH_CH0_CHUNK	3:0	0x0	Chunk size between VIP host port and DMA for port 0
VIPH_CH1_CHUNK	5:4	0x0	Chunk size between VIP host port and DMA for port 1
VIPH_CH2_CHUNK	7:6	0x0	Chunk size between VIP host port and DMA for port 2
VIPH_CH3_CHUNK	9:8	0x0	Chunk size between VIP host port and DMA for port 3
VIPH_CH0_ABORT	16	0x0	Abort DMA operation through port 0
VIPH_CH1_ABORT	17	0x0	Abort DMA operation through port 1
VIPH_CH2_ABORT	18	0x0	Abort DMA operation through port 2
VIPH_CH3_ABORT	19	0x0	Abort DMA operation through port 3

DMA transfer chunk size and abort control

VIPH_DV_INT - RW - 32 bits - [GpuF0MMReg:0xC4C]			
Field Name	**Bits**	**Default**	**Description**
VIPH_DV0_INT_EN	0	0x0	Interrupt polling enable for VIP slave device 0
VIPH_DV1_INT_EN	1	0x0	Interrupt polling enable for VIP slave device 1
VIPH_DV2_INT_EN	2	0x0	Interrupt polling enable for VIP slave device 2
VIPH_DV3_INT_EN	3	0x0	Interrupt polling enable for VIP slave device 3
VIPH_DV0_INT (R)	4	0x0	Interrupt
VIPH_DV0_AK (W)	4	0x0	Clear interrupt with a '1'
VIPH_DV1_INT (R)	5	0x0	Interrupt
VIPH_DV1_AK (W)	5	0x0	Clear interrupt with a '1'
VIPH_DV2_INT (R)	6	0x0	Interrupt
VIPH_DV2_AK (W)	6	0x0	Clear interrupt with a '1'
VIPH_DV3_INT (R)	7	0x0	Interrupt
VIPH_DV3_AK (W)	7	0x0	Clear interrupt with a '1'

VIP Host port interrupt control

VIPH_TIMEOUT_STAT - RW - 32 bits - [GpuF0MMReg:0xC50]			
Field Name	**Bits**	**Default**	**Description**
VIPH_FIFO0_STAT (R)	0	0x0	'1' if port 0 time out or hung.
VIPH_FIFO0_AK (W)	0	0x0	Clear FIFO0_STAT with a '1'
VIPH_FIFO1_STAT (R)	1	0x0	'1' if port 1 time out or hung.
VIPH_FIFO1_AK (W)	1	0x0	Clear FIFO1_STAT with a '1'
VIPH_FIFO2_STAT (R)	2	0x0	'1' if port 2 time out or hung.
VIPH_FIFO2_AK (W)	2	0x0	Clear FIFO2_STAT with a '1'
VIPH_FIFO3_STAT (R)	3	0x0	'1' if port 3 time out or hung.
VIPH_FIFO3_AK (W)	3	0x0	Clear FIFO3_STAT with a '1'
VIPH_REG_STAT (R)	4	0x0	'1' if register port time out or hung.
VIPH_REG_AK (W)	4	0x0	Clear REG_STAT with a '1'
VIPH_AUTO_INT_STAT (R)	5	0x0	'1' if auto interrupt polling time out or hung.
VIPH_AUTO_INT_AK (W)	5	0x0	Clear AUTO_INT_STAT with a '1'

VIPH_FIFO0_MASK	8	0x0	'0' disable interrupt.
VIPH_FIFO1_MASK	9	0x0	'0' disable interrupt.
VIPH_FIFO2_MASK	10	0x0	'0' disable interrupt.
VIPH_FIFO3_MASK	11	0x0	'0' disable interrupt.
VIPH_REG_MASK	12	0x0	'0' disable interrupt.
VIPH_AUTO_INT_MASK	13	0x0	'0' disable interrupt.
VIPH_DV0_INT_MASK	16	0x0	'0' disable interrupt.
VIPH_DV1_INT_MASK	17	0x0	'0' disable interrupt.
VIPH_DV2_INT_MASK	18	0x0	'0' disable interrupt.
VIPH_DV3_INT_MASK	19	0x0	'0' disable interrupt.
VIPH_INTPIN_EN	20	0x0	'0' means no physical pins used for VIP interrupt. 1= physical pins used.
VIPH_INTPIN_INT (R)	21	0x0	'1' if physical pins has interrupt.
VIPH_REGR_DIS	24	0x0	'0'= any host read from VIPH_REG_DATA will trigger VIP register cycle. 1= Read from VIPH_REG_DATA will not trigger VIP register cycle.
VIP_RBBMIF_RDWR_TIMEOUT_DIS	31	0x0	This bit is unused because VIP doesn't have its own decode. '0'= enable RBBMIF read/write timeout logic. 1= disable RBBMIF read/write timeout logic.

VIP Host Port Time Out Status

2.5.3 Capture Registers

VID_BUFFER_CONTROL - RW - 32 bits - [GpuF0MMReg:0xB00]			
Field Name	**Bits**	**Default**	**Description**
CAP0_BUFFER_WATER_MARK	9:0	0x10	Capture 0 buffer water mark.
FULL_BUFFER_EN	16	0x0	1= The shared buffer is dedicated to one capture only. 0=DISABLE 1=ENABLE
CAP0_ANC_VBI_QUAD_BUF	17	0x0	0=Dual buffer 1=Quaduple buffer
VID_BUFFER_RESET	20	0x0	Reset the buffer pointers. 0=NOT RESET 1=RESET
CAP_SWAP	22:21	0x0	Capture Port Swap control.
CAP0_BUFFER_EMPTY (R)	24	0x0	Capture 0's buffer empty status. 0=EMPTY 1=NOT EMPTY
CAP_URGENT_EN	31	0x1	Enable urgent signal to MH when water mark is reached.

Video Capture port buffer control.

CAP_INT_CNTL - RW - 32 bits - [GpuF0MMReg:0xB08]			
Field Name	**Bits**	**Default**	**Description**
CAP0_BUF0_INT_EN	0	0x0	Capture 0 Buffer 0 Interrupt enable. 0=Disable 1=Enable
CAP0_BUF0_EVEN_INT_EN	1	0x0	Capture 0 Buffer 0 even frame Interrupt enable. 0=Disable 1=Enable

CAP0_BUF1_INT_EN	2	0x0	Capture 0 Buffer 1 Interrupt enable. 0=Disable 1=Enable
CAP0_BUF1_EVEN_INT_EN	3	0x0	Capture 0 Buffer 1 even frame Interrupt enable. 0=Disable 1=Enable
CAP0_VBI0_INT_EN	4	0x0	Capture 0 VBI Buffer 0 Interrupt enable. 0=Disable 1=Enable
CAP0_VBI1_INT_EN	5	0x0	Capture 0 VBI Buffer 1 Interrupt enable. 0=Disable 1=Enable
CAP0_ONESHOT_INT_EN	6	0x0	Capture 0 ONESHOT Buffer Interrupt enable. 0=Disable 1=Enable
CAP0_ANC0_INT_EN	7	0x0	Capture 0 ANC Buffer 0 Interrupt enable. 0=Disable 1=Enable
CAP0_ANC1_INT_EN	8	0x0	Capture 0 ANC Buffer 1 Interrupt enable. 0=Disable 1=Enable
CAP0_VBI2_INT_EN	9	0x0	Capture 0 VBI Buffer 2 Interrupt enable. 0=Disable 1=Enable
CAP0_VBI3_INT_EN	10	0x0	Capture 0 VBI Buffer 3 Interrupt enable. 0=Disable 1=Enable
CAP0_ANC2_INT_EN	11	0x0	Capture 0 ANC Buffer 2 Interrupt enable. 0=Disable 1=Enable
CAP0_ANC3_INT_EN	12	0x0	Capture 0 ANC Buffer 3 Interrupt enable. 0=Disable 1=Enable
CAP0_BUF_INT_MUX	13	0x0	Wait for MH ack before setting capture interrupt. 0=Disable 1=Enable

Video Capture port interrupt control register

CAP_INT_STATUS - RW - 32 bits - [GpuF0MMReg:0xB0C]			
Field Name	**Bits**	**Default**	**Description**
CAP0_BUF0_INT (R)	0	0x0	Read only. Buffer 0 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_BUF0_INT_AK (W)	0	0x0	Buf0 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_BUF0_EVEN_INT (R)	1	0x0	Read only. Buffer 0 even frame interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_BUF0_EVEN_INT_AK (W)	1	0x0	Buf0 even frame buffer interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_BUF1_INT (R)	2	0x0	Read only. Buffer 1 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_BUF1_INT_AK (W)	2	0x0	Buf1 interrupt acknowledgment. 0=No effect 1=Clear status

CAP0_BUF1_EVEN_INT (R)	3	0x0	Read only. Buffer 1 even frame interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_BUF1_EVEN_INT_AK (W)	3	0x0	Buf1 even frame buffer interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_VBI0_INT (R)	4	0x0	Read only. VBI buffer 0 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_VBI0_INT_AK (W)	4	0x0	VBI buffer 0 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_VBI1_INT (R)	5	0x0	Read only. VBI buffer 1 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_VBI1_INT_AK (W)	5	0x0	VBI buffer 1 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_ONESHOT_INT (R)	6	0x0	Read only. ONESHOT buffer interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_ONESHOT_INT_AK (W)	6	0x0	ONESHOT buffer interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_ANC0_INT (R)	7	0x0	Read only. ANC buffer 0 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_ANC0_INT_AK (W)	7	0x0	ANC buffer 0 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_ANC1_INT (R)	8	0x0	Read only. ANC buffer 1 nterrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_ANC1_INT_AK (W)	8	0x0	ANC buffer 1 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_VBI2_INT (R)	9	0x0	Read only. VBI buffer 2 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_VBI2_INT_AK (W)	9	0x0	VBI buffer 2 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_VBI3_INT (R)	10	0x0	Read only. VBI buffer 3 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_VBI3_INT_AK (W)	10	0x0	VBI buffer 3 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_ANC2_INT (R)	11	0x0	Read only. ANC buffer 2 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_ANC2_INT_AK (W)	11	0x0	ANC buffer 2 interrupt acknowledgment. 0=No effect 1=Clear status
CAP0_ANC3_INT (R)	12	0x0	Read only. ANC buffer 3 interrupt status. 0=No event 1=Event has occurred, interrupting if enabled
CAP0_ANC3_INT_AK (W)	12	0x0	ANC buffer 3 interrupt acknowledgment. 0=No effect 1=Clear status

Capture port interrupt control.

CAP0_BUF0_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB20]			
Field Name	**Bits**	**Default**	**Description**
CAP_BUF0_OFFSET	31:0	0x0	Capture Port 0 Buffer 0 starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture Port 0 Buffer 0 starting address

CAP0_BUF1_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB24]			
Field Name	**Bits**	**Default**	**Description**
CAP_BUF1_OFFSET	31:0	0x0	Capture Port 0 Buffer 1 starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture Port 0 Buffer 1 starting address

CAP0_BUF0_EVEN_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB28]			
Field Name	**Bits**	**Default**	**Description**
CAP_BUF0_EVEN_OFFSET	31:0	0x0	Capture Port 0 Buffer 0 even frame starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture Port 0 Buffer 0 even frame starting address

CAP0_BUF1_EVEN_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB2C]			
Field Name	**Bits**	**Default**	**Description**
CAP_BUF1_EVEN_OFFSET	31:0	0x0	Capture Port 0 Buffer 1 even frame starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture Port 0 Buffer 1 even frame starting address

CAP0_BUF_PITCH - RW - 32 bits - [GpuF0MMReg:0xB30]			
Field Name	**Bits**	**Default**	**Description**
CAP_BUF_PITCH	11:0	0x0	Capture 0 buffer's pitch. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 buffer's pitch.

CAP0_V_WINDOW - RW - 32 bits - [GpuF0MMReg:0xB34]			
Field Name	**Bits**	**Default**	**Description**
CAP_V_START	11:0	0x0	Vertical window starting line number.
CAP_V_END	27:16	0x0	Vertical window end line number.

Capture 0's Vertical window.

CAP0_H_WINDOW - RW - 32 bits - [GpuF0MMReg:0xB38]			
Field Name	**Bits**	**Default**	**Description**
CAP_H_START	11:0	0x0	Horizontal window's start.
CAP_H_WIDTH	27:16	0x0	Horizontal window's width. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0's Horizontal window.

CAP0_VBI0_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB3C]			
Field Name	**Bits**	**Default**	**Description**
CAP_VBI0_OFFSET	31:0	0x0	Capture 0 VBI 0 buffer's starting address. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 VBI 0 buffer's starting address.

CAP0_VBI1_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB40]			
Field Name	**Bits**	**Default**	**Description**
CAP_VBI1_OFFSET	31:0	0x0	Capture 0 VBI 1 buffer's starting address. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 VBI 1 buffer's starting address.

CAP0_VBI_V_WINDOW - RW - 32 bits - [GpuF0MMReg:0xB44]			
Field Name	**Bits**	**Default**	**Description**
CAP_VBI_V_START	11:0	0x0	Capture 0 VBI's Vertical start.
CAP_VBI_V_END	27:16	0x0	Capture 0 VBI's Vertical End.

Capture 0 VBI's vertical window

CAP0_VBI_H_WINDOW - RW - 32 bits - [GpuF0MMReg:0xB48]			
Field Name	**Bits**	**Default**	**Description**
CAP_VBI_H_START	11:0	0x0	Capture 0 VBI's Horizontal start.

CAP_VBI_H_WIDTH	27:16	0x0	Capture 0 VBI's Horizontal Width. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 VBI's horizontal window

CAP0_PORT_MODE_CNTL - RW - 32 bits - [GpuF0MMReg:0xB4C]			
Field Name	**Bits**	**Default**	**Description**
CAP_PORT_WIDTH	1	0x0	Capture 0 port width. 0=8 bits 1=16 bits
CAP_PORT_BYTE_USED	2	0x0	In 8 bit width mode, which byte used. 0=lower byte used 1=upper byte used
CAP_DDR_MODE	3	0x0	Capture DDR mode. 0=DDR mode off 1=DDR mode on
CAP_DDR_SYNC	4	0x0	Embedded sync words DDR mode. 0=Sync on rising edge 1=Sync on both edges
MOBILE_DIS	5	0x1	Mobile/Desktop configuration. 0=Mobile 1=Desktop

Capture 0 mode control register.

CAP0_TRIG_CNTL - RW - 32 bits - [GpuF0MMReg:0xB50]			
Field Name	**Bits**	**Default**	**Description**
CAP_TRIGGER_R (R)	1:0	0x0	Read only. Capture status. 0=capture complete 1=capture pending 2=capture in progress
CAP_TRIGGER_W (W)	0	0x0	Write only. Start capture next frame. 0=no action 1=capture next field/frame
CAP_EN	4	0x0	Capture 0 enable. 0=disable 1=enable
CAP_VSYNC_CNT (R)	15:8	0x0	Read only. VSYNC counter.
CAP_VSYNC_CLR	16	0x0	Reset the VSYNC counter.

Capture 0 trigger control.

CAP0_DEBUG - RW - 32 bits - [GpuF0MMReg:0xB54]			
Field Name	**Bits**	**Default**	**Description**
CAP_H_STATUS (R)	11:0	0x0	Capture 0 Horizontal status.
CAP_V_STATUS (R)	27:16	0x0	Capture 0 vertical status.
CAP_V_SYNC (R)	28	0x0	Capture 0 VSYNC status.

Capture 0 debug status register.

CAP0_CONFIG - RW - 32 bits - [GpuF0MMReg:0xB58]			
Field Name	**Bits**	**Default**	**Description**
CAP_INPUT_MODE	0	0x0	Input mode. 0=OneShot trigger mode 1=Enable continuous capture
CAP_START_FIELD	1	0x0	Starting field. 0=Odd 1=Even
CAP_START_BUF_R (R)	2	0x0	Read only. Current starting buffer. 0=Buffer 0 1=Buffer 1
CAP_START_BUF_W (W)	3	0x0	Write only. Control starting buffer. 0=Buffer 0 1=Buffer 1
CAP_BUF_TYPE	5:4	0x0	Buffer type. 0=Field 1=Alternating 2=Frame
CAP_ONESHOT_MODE	6	0x0	ONESHOT mode. 0=FIELD 1=FRAME
CAP_BUF_MODE	8:7	0x0	Capture 0 buffer mode. 0=Single 1=Double 2=Triple
CAP_MIRROR_EN	9	0x0	Capture 0 mirroring function enable. 0=Normal 1=Mirror
CAP_ONESHOT_MIRROR_EN	10	0x0	ONESHOT buffer mirroring function enable. 0=Normal 1=Mirror
CAP_VIDEO_SIGNED_UV	11	0x0	Enable conversion to signed value. 1=Convert to signed
CAP_ANC_DECODE_EN	12	0x0	ANC enable. 0=disable 1=enable
CAP_VBI_EN	13	0x0	VBI enable. 0=disable 1=enable
CAP_SOFT_PULL_DOWN_EN	14	0x0	Software pull down enable. 0=disable 1=enable
CAP_VIP_EXTEND_FLAG_EN	15	0x0	Extended flag enable. 0=DISABLE 1=ENABLE
CAP_FAKE_FIELD_EN	16	0x1	Fake field enable. 0=DISABLE 1=ENABLE
CAP_FIELD_START_LINE_DIFF	18:17	0x0	Odd, Even frame line number differences. 0=EQUAL 1=ODD_ONE_MORE_LINE 2=EVEN_ONE_MORE_LINE
CAP_HORZ_DOWN	20:19	0x0	Horizontal decimation. 0=Normal 1=x2 2=x4

CAP_VERT_DOWN	22:21	0x0	Vertical decimation. 0=Normal 1=x2 2=x4
CAP_STREAM_FORMAT	25:23	0x0	Video stream format. 0=Brooktree 1=CCIR 656 2=ZV 3=16bit VIP 4=TRANSPORT STREAM
CAP_HDWNS_DEC	26	0x1	Horizontal downscaler or decimator. 0=downscaler 1=decimator
CAP_IMAGE_FLIP_EN	27	0x0	0=Normal 1=Flip
CAP_ONESHOT_IMAGE_FLIP_EN	28	0x0	0=Normal 1=Flip
CAP_VIDEO_IN_FORMAT	29	0x0	Input format. 0=YVYU422 1=VYUY422
VBI_HORZ_DOWN	31:30	0x0	0=Normal 1=x2 2=x4

Capture 0 configuration register.

CAP0_ANC0_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB5C]			
Field Name	**Bits**	**Default**	**Description**
CAP_ANC0_OFFSET	31:0	0x0	Starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 ANC 0 starting address.

CAP0_ANC1_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB60]			
Field Name	**Bits**	**Default**	**Description**
CAP_ANC1_OFFSET	31:0	0x0	Starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 ANC 1 starting address.

CAP0_ANC_H_WINDOW - RW - 32 bits - [GpuF0MMReg:0xB64]			
Field Name	**Bits**	**Default**	**Description**
CAP_ANC_WIDTH	11:0	0x0	Window width. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 ANC horizontal window.

CAP0_VIDEO_SYNC_TEST - RW - 32 bits - [GpuF0MMReg:0xB68]			
Field Name	**Bits**	**Default**	**Description**
CAP_TEST_VID_SOF	0	0x0	Start of field.
CAP_TEST_VID_EOF	1	0x0	End of field.
CAP_TEST_VID_EOL	2	0x0	End of line.
CAP_TEST_VID_FIELD	3	0x0	Odd/Even field. 0=Even Field 1=Odd Field
CAP_TEST_SYNC_EN	5	0x0	Test sync enable. 0=Normal 1=Test Mode

Capture port 0 sync test.

CAP0_ONESHOT_BUF_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB6C]			
Field Name	**Bits**	**Default**	**Description**
CAP_ONESHOT_BUF_OFFSET	31:0	0x0	ONESHOT buffer starting address. NOTE: Bits 0:1 of this field are hardwired to ZERO.

ONESHOT buffer starting address.

CAP0_BUF_STATUS - RW - 32 bits - [GpuF0MMReg:0xB70]			
Field Name	**Bits**	**Default**	**Description**
CAP_PRE_VID_BUF (R)	1:0	0x0	Read only. Previous capture buffer.
CAP_CUR_VID_BUF (R)	3:2	0x0	Read only. Current Capture buffer.
CAP_PRE_FIELD (R)	4	0x0	Read only. Previous field.
CAP_CUR_FIELD (R)	5	0x0	Read only. Current field.
CAP_PRE_VBI_BUF (R)	7:6	0x0	Read only. Previous VBI buffer.
CAP_CUR_VBI_BUF (R)	9:8	0x0	Read only. Current VBI buffer.
CAP_VBI_BUF_STATUS (R)	10	0x0	Read only. VBI busy status. 0=done 1=busy
CAP_PRE_ANC_BUF (R)	12:11	0x0	Read only. Previous ANC buffer.
CAP_CUR_ANC_BUF (R)	14:13	0x0	Read only. Current ANC buffer.
CAP_ANC_BUF_STATUS (R)	15	0x0	Read only. Buffer busy status. 0=done 1=busy
CAP_ANC_PRE_BUF_CNT (R)	27:16	0x0	Read only. Buffer count.
CAP_VIP_INC (R)	28	0x0	Read only. Interlaced or not. 0=INTERLACED 1=NON_INTERLACED
CAP_VIP_PRE_REPEAT_FIELD (R)	29	0x0	Read only. Previous buffer is new/repeat field. 0=new_field 1=repeated_field
CAP_CAP_BUF_STATUS (R)	30	0x0	Read only. Capture buffer busy status. 0=done 1=busy
CAP_VIP_STATUS_STROBE (R)	31	0x0	Read only. Status strobe changes polarity when there is a buffer change.

Capture 0 buffer status.

CAP0_ANC_BUF01_BLOCK_CNT - RW - 32 bits - [GpuF0MMReg:0xB74]			
Field Name	**Bits**	**Default**	**Description**
CAP0_ANC_BUF0_BLOCK_CNT (R)	11:0	0x0	
CAP0_ANC_BUF1_BLOCK_CNT (R)	27:16	0x0	

CAP0_ANC_BUF23_BLOCK_CNT - RW - 32 bits - [GpuF0MMReg:0xB7C]			
Field Name	**Bits**	**Default**	**Description**
CAP0_ANC_BUF2_BLOCK_CNT (R)	11:0	0x0	
CAP0_ANC_BUF3_BLOCK_CNT (R)	27:16	0x0	

CAP0_VBI2_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB80]			
Field Name	**Bits**	**Default**	**Description**
CAP_VBI2_OFFSET	31:0	0x0	Capture 0 VBI 2 buffer's starting address. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 VBI 2 buffer's starting address.

CAP0_VBI3_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB84]			
Field Name	**Bits**	**Default**	**Description**
CAP_VBI3_OFFSET	31:0	0x0	Capture 0 VBI 3 buffer's starting address. NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 VBI 3 buffer's starting address.

CAP0_ANC2_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB88]			
Field Name	**Bits**	**Default**	**Description**
CAP_ANC2_OFFSET	31:0	0x0	Starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 ANC 2 starting address.

CAP0_ANC3_OFFSET - RW - 32 bits - [GpuF0MMReg:0xB8C]			
Field Name	**Bits**	**Default**	**Description**

CAP_ANC3_OFFSET	31:0	0x0	Starting address NOTE: Bits 0:1 of this field are hardwired to ZERO.

Capture 0 ANC 3 starting address.

2.5.4 VIP Host Port DMA Registers

DMA_VIPH0_COMMAND - R - 32 bits - [GpuF0MMReg:0xA00]			
Field Name	**Bits**	**Default**	**Description**
BYTE_COUNT	20:0	0x0	Byte Count of transfer size.
SWAP_CONTROL	25:24	0x0	Endian's swap control. 0=No Swapping 1=[15:0]=[31:16], [31:16]=[15:0] 2=[7:0]=[31:24], [15:8]=[23:16], [23:16]=[15:8], [31:24]=[7:0] 3=Undefined
TRANSFER_SOURCE	26	0x0	Address space of source data. 0=Transfer from memory 1=Transfer from VIPH
TRANSFER_DEST	27	0x0	Address space of destination data. 0=Transfer to memory 1=Transfer to VIPH
SOURCE_OFFSET_HOLD	28	0x0	Hold the source address without increase. 0=Increment 1=Hold
DEST_OFFSET_HOLD	29	0x0	Hold the destination address without increase. 0=Increment 1=Hold
INTERRUPT_DIS	30	0x0	End of DMA command table interrupt control. 0=Normal 1=Disable the end of list interrupt
END_OF_LIST_STATUS	31	0x0	Status bit show the last command of the DMA table. 0=Normal 1=End of Descriptor List

VIPH channel0 DMA command read back.

DMA_VIPH1_COMMAND - R - 32 bits - [GpuF0MMReg:0xA04]			
Field Name	**Bits**	**Default**	**Description**
BYTE_COUNT	20:0	0x0	Byte Count of transfer size.
SWAP_CONTROL	25:24	0x0	Endian's swap control. 0=No Swapping 1=[15:0]=[31:16], [31:16]=[15:0] 2=[7:0]=[31:24], [15:8]=[23:16], [23:16]=[15:8], [31:24]=[7:0] 3=Undefined
TRANSFER_SOURCE	26	0x0	Address space of source data. 0=Transfer from memory 1=Transfer from VIPH
TRANSFER_DEST	27	0x0	Address space of destination data. 0=Transfer to memory 1=Transfer to VIPH

SOURCE_OFFSET_HOLD	28	0x0	Hold the source address without increase. 0=Increment 1=Hold
DEST_OFFSET_HOLD	29	0x0	Hold the destination address without increase. 0=Increment 1=Hold
INTERRUPT_DIS	30	0x0	End of DMA command table interrupt control. 0=Normal 1=Disable the end of list interrupt
END_OF_LIST_STATUS	31	0x0	Status bit show the last command of the DMA table. 0=Normal 1=End of Descriptor List

VIPH channel1 DMA command read back.

DMA_VIPH2_COMMAND - R - 32 bits - [GpuF0MMReg:0xA08]			
Field Name	**Bits**	**Default**	**Description**
BYTE_COUNT	20:0	0x0	Byte Count of transfer size.
SWAP_CONTROL	25:24	0x0	Endian's swap control. 0=No Swapping 1=[15:0]=[31:16], [31:16]=[15:0] 2=[7:0]=[31:24], [15:8]=[23:16], [23:16]=[15:8], [31:24]=[7:0] 3=Undefined
TRANSFER_SOURCE	26	0x0	Address space of source data. 0=Transfer from memory 1=Transfer from VIPH
TRANSFER_DEST	27	0x0	Address space of destination data. 0=Transfer to memory 1=Transfer to VIPH
SOURCE_OFFSET_HOLD	28	0x0	Hold the source address without increase. 0=Increment 1=Hold
DEST_OFFSET_HOLD	29	0x0	Hold the destination address without increase. 0=Increment 1=Hold
INTERRUPT_DIS	30	0x0	End of DMA command table interrupt control. 0=Normal 1=Disable the end of list interrupt
END_OF_LIST_STATUS	31	0x0	Status bit show the last command of the DMA table. 0=Normal 1=End of Descriptor List

VIPH channel2 DMA command read back.

DMA_VIPH3_COMMAND - R - 32 bits - [GpuF0MMReg:0xA0C]			
Field Name	**Bits**	**Default**	**Description**
BYTE_COUNT	20:0	0x0	Byte Count of transfer size.
SWAP_CONTROL	25:24	0x0	Endian's swap control. 0=No Swapping 1=[15:0]=[31:16], [31:16]=[15:0] 2=[7:0]=[31:24], [15:8]=[23:16], [23:16]=[15:8], [31:24]=[7:0] 3=Undefined

TRANSFER_SOURCE	26	0x0	Address space of source data. 0=Transfer from memory 1=Transfer from VIPH
TRANSFER_DEST	27	0x0	Address space of destination data. 0=Transfer to memory 1=Transfer to VIPH
SOURCE_OFFSET_HOLD	28	0x0	Hold the source address without increase. 0=Increment 1=Hold
DEST_OFFSET_HOLD	29	0x0	Hold the destination address without increase. 0=Increment 1=Hold
INTERRUPT_DIS	30	0x0	End of DMA command table interrupt control. 0=Normal 1=Disable the end of list interrupt
END_OF_LIST_STATUS	31	0x0	Status bit show the last command of the DMA table. 0=Normal 1=End of Descriptor List

VIPH channel3 DMA command read back.

DMA_VIPH_STATUS - R - 32 bits - [GpuF0MMReg:0xA10]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH0_AVAIL	3:0	0x3	VIPH DMA channel 0 available job queue number.
DMA_VIPH1_AVAIL	7:4	0x3	VIPH DMA channel 1 available job queue number.
DMA_VIPH2_AVAIL	11:8	0x3	VIPH DMA channel 2 available job queue number.
DMA_VIPH3_AVAIL	15:12	0x3	VIPH DMA channel 3 available job queue number.
DMA_VIPH0_CURRENT	17:16	0x0	VIPH DMA channel 0 current active job queue number
DMA_VIPH1_CURRENT	19:18	0x0	VIPH DMA channel 1 current active job queue number
DMA_VIPH2_CURRENT	21:20	0x0	VIPH DMA channel 2 current active job queue number
DMA_VIPH3_CURRENT	23:22	0x0	VIPH DMA channel 3 current active job queue number
DMA_VIPH0_ACTIVE	24	0x0	VIPH DMA channel 0 active status. 0=All VIP0 queue transfers are all done 1=A VIP0 queue transfer is active
DMA_VIPH1_ACTIVE	25	0x0	VIPH DMA channel 1 active status. 0=All VIP1 queue transfers are all done 1=A VIP1 queue transfer is active
DMA_VIPH2_ACTIVE	26	0x0	VIPH DMA channel 2 active status. 0=All VIP2 queue transfers are all done 1=A VIP2 queue transfer is active
DMA_VIPH3_ACTIVE	27	0x0	VIPH DMA channel 3 active status. 0=All VIP3 queue transfers are all done 1=A VIP3 queue transfer is active
VIP_RBBM_H0DMA_IDLE	28	0x0	0=VIP DMA channel 0 is busy 1=VIP DMA channel 0 is idle
VIP_RBBM_H1DMA_IDLE	29	0x0	0=VIP DMA channel 1 is busy 1=VIP DMA channel 1 is idle
VIP_RBBM_H2DMA_IDLE	30	0x0	0=VIP DMA channel 2 is busy 1=VIP DMA channel 2 is idle
VIP_RBBM_H3DMA_IDLE	31	0x0	0=VIP DMA channel 3 is busy 1=VIP DMA channel 3 is idle

VIPH DMA channels status register.

DMA_VIPH_MISC_CNTL - RW - 32 bits - [GpuF0MMReg:0xA14]			
Field Name	**Bits**	**Default**	**Description**

DMA_VIPH_READ_TIMER	3:0	0xf	
DMA_VIPH_READ_TIMEOUT_TO_PRIORITY_EN	7	0x0	0=Disable 1=Enable
DMA_VIPH_READ_TIMEOUT_STATUS (R)	8	0x0	0=Normal 1=Timeout
DMA_VIPH_URGENT_EN	9	0x1	0=Disable 1=Enable urgent to MH if read times out

DMA_VIPH_CHUNK_0 - RW - 32 bits - [GpuF0MMReg:0xA18]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH3_TABLE_SWAP	1:0	0x0	VIPH DMA Channel 3 Endian swap control. 0=No swap 1=8bit swap 2=16bit swap 3=reserved
DMA_VIPH2_TABLE_SWAP	3:2	0x0	VIPH DMA Channel 2 Endian swap control. 0=No swap 1=8bit swap 2=16bit swap 3=reserved
DMA_VIPH1_TABLE_SWAP	5:4	0x0	VIPH DMA Channel 1 Endian swap control. 0=No swap 1=8bit swap 2=16bit swap 3=reserved
DMA_VIPH0_TABLE_SWAP	7:6	0x0	VIPH DMA Channel 0 Endian swap control. 0=No swap 1=8bit swap 2=16bit swap 3=reserved
DMA_VIPH3_NOCHUNK	28	0x0	VIPH DMA Channel 3 disregard chunk size 0=Use chunk value 1=Use infinity for the chunk value
DMA_VIPH2_NOCHUNK	29	0x0	VIPH DMA Channel 2 disregard chunk size 0=Use chunk value 1=Use infinity for the chunk value
DMA_VIPH1_NOCHUNK	30	0x0	VIPH DMA Channel 1 disregard chunk size 0=Use chunk value 1=Use infinity for the chunk value
DMA_VIPH0_NOCHUNK	31	0x0	VIPH DMA Channel 0 disregard chunk size 0=Use chunk value 1=Use infinity for the chunk value

VIP Host Port DMA Chunk control register.

DMA_VIPH_CHUNK_1_VAL - RW - 32 bits - [GpuF0MMReg:0xA1C]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIP0_CHUNK	7:0	0xf	VIP Host Port DMA channel 0 Chunk size
DMA_VIP1_CHUNK	15:8	0xf	VIP Host Port DMA channel 1 Chunk size
DMA_VIP2_CHUNK	23:16	0xf	VIP Host Port DMA channel 2 Chunk size
DMA_VIP3_CHUNK	31:24	0xf	VIP Host Port DMA channel 3 Chunk size

VIP Host Port DMA Chunk size

DMA_VIP0_TABLE_ADDR - W - 32 bits - [GpuF0MMReg:0xA20]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR VIP Port 0 DMA table starting address	31:0	0x0	This points to first entry in the DMA table.

DMA_VIP1_TABLE_ADDR - W - 32 bits - [GpuF0MMReg:0xA30]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR VIP Port 1 DMA table starting address	31:0	0x0	This points to first entry in the DMA table.

DMA_VIP2_TABLE_ADDR - W - 32 bits - [GpuF0MMReg:0xA40]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR VIP Port 2 DMA table starting address	31:0	0x0	This points to first entry in the DMA table.

DMA_VIP3_TABLE_ADDR - W - 32 bits - [GpuF0MMReg:0xA50]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR VIP Port 3 DMA table starting address	31:0	0x0	This points to first entry in the DMA table.

DMA_VIPH0_ACTIVE - R - 32 bits - [GpuF0MMReg:0xA24]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR_ACT VIP Port 0 DMA Current table address	31:0	0x0	This points to the current active entry in the DMA table.

DMA_VIPH1_ACTIVE - R - 32 bits - [GpuF0MMReg:0xA34]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR_ACT VIP Port 1 DMA Current table address	31:0	0x0	This points to the current active entry in the DMA table.

DMA_VIPH2_ACTIVE - R - 32 bits - [GpuF0MMReg:0xA44]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR_ACT	31:0	0x0	This points to the current active entry in the DMA table.

VIP Port 2 DMA Current table address

DMA_VIPH3_ACTIVE - R - 32 bits - [GpuF0MMReg:0xA54]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH_TABLE_ADDR_ACT	31:0	0x0	This points to the current active entry in the DMA table.

VIP Port 3 DMA Current table address

DMA_VIPH_ABORT - RW - 32 bits - [GpuF0MMReg:0xA88]			
Field Name	**Bits**	**Default**	**Description**
DMA_VIPH0_ABORT_EN	3	0x0	Enable abort action 0=Normal 1=Enable queue abort
DMA_VIPH1_ABORT_EN	7	0x0	Enable abort action 0=Normal 1=Enable queue abort
DMA_VIPH2_ABORT_EN	11	0x0	Enable abort action 0=Normal 1=Enable queue abort
DMA_VIPH3_ABORT_EN	15	0x0	Enable abort action 0=Normal 1=Enable queue abort
DMA_VIPH0_RESET	20	0x0	Soft reset. Reset the DMA and job queue.
DMA_VIPH1_RESET	21	0x0	Soft reset. Reset the DMA and job queue.
DMA_VIPH2_RESET	22	0x0	Soft reset. Reset the DMA and job queue.
DMA_VIPH3_RESET	23	0x0	Soft reset. Reset the DMA and job queue.

VIP Host Port DMA abort control registers

2.5.5 General Purpose I/O Data and Control Registers

Registers responsible for GPIO Pads' Programmability and values.

GPIOPAD_STRENGTH - RW - 32 bits - [GpuF0MMReg:0x1794]			
Field Name	**Bits**	**Default**	**Description**
GPIO_STRENGTH_SN	3:0	0x9	For NMOS of GPIOs. 0=For NMOS of GPIOs.
GPIO_STRENGTH_SP	7:4	0xa	For PMOS of GPIOs. 0=For PMOS of GPIOs.

Pad strength for GPIOs

GPIOPAD_MASK - RW - 32 bits - [GpuF0MMReg:0x1798]			
Field Name	**Bits**	**Default**	**Description**
GPIO_MASK	28:0	0x0	GPIO pads mask. Allows software to control the GPIO pad. POSSIBLE VALUES: 1 - Only software can control GPIO pad. 0 - Allows chip components to control GPIO pad.

GPIO pads mask register

GPIOPAD_A - RW - 32 bits - [GpuF0MMReg:0x179C]			
Field Name	**Bits**	**Default**	**Description**
GPIO_A	28:0	0x0	GPIO pads output. The value to be outputted to the pads if GPIO_EN is '1'.

GPIO pads output register

GPIOPAD_EN - RW - 32 bits - [GpuF0MMReg:0x17A0]			
Field Name	**Bits**	**Default**	**Description**
GPIO_EN	28:0	0x0	GPIO pads output enable. If 1, GPIO pad is in output mode. If 0, GPIO pad accepts inputs from pads.

GPIO pads output enable register

GPIOPAD_Y - RW - 32 bits - [GpuF0MMReg:0x17A4]			
Field Name	**Bits**	**Default**	**Description**
GPIO_Y (R)	28:0	0x0	GPIO pads input (or the values on the GPIO pads).

GPIO pad input read back

GPIOPAD_EXTERN_TRIG_CNTL - RW - 32 bits - [GpuF0MMReg:0x17C4]			
Field Name	**Bits**	**Default**	**Description**
EXTERN_TRIG_SEL	4:0	0x0	Selects whether one of the GPIOs, or a signal from display is used for detecting an external trigger event: 0= GPIO_0 1= GPIO_1 2= GPIO_2 3= GPIO_3 4= GPIO_4 5= GPIO_5 6= GPIO_6 7= GPIO_7 8= GPIO_8 9= GPIO_9 10= GPIO_10 11= GPIO_11 12= GPIO_12 13= GPIO_13 14= GPIO_14 15= GPIO_15 16= GPIO_16 17= GPIO_17 18= GPIO_18 19= GPIO_19 20= GPIO_20 21= Display pin 22= Disable external trigger source event for both GPIO pad and Display pin
EXTERN_TRIG_CLR (W)	5	0x0	Clearing External Trigger logic: 0= Write 0 has no affect. 1= Write 1 sets EXTERN_TRIG_READ to 0.
EXTERN_TRIG_READ (R)	6	0x0	Checks the status of an external trigger event: 0= No external trigger event occurred OR an external trigger event that has been acknowledged by writing to EXTERN_TRIG_CLR with a '1'. 1= An external trigger event has occurred and is waiting to be acknowledged.

External Trigger register

2.5.6 VIP Miscellaneous Registers

VIPPAD_MASK - RW - 32 bits - [GpuF0MMReg:0xAC0]			
Field Name	**Bits**	**Default**	**Description**
VIPPAD_MASK_SCL	0	0x0	Desktop: GPIO override for SCL. Mobile: GPIO override for GPIO[19]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_SDA	1	0x0	Desktop: GPIO override for SDA. Mobile: GPIO override for GPIO[18]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.

VIPPAD_MASK_VHAD	3:2	0x0	Desktop: GPIO override for VHAD[1:0]. Mobile: GPIO override for GPIO[23:22]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_VPHCTL	4	0x0	Desktop: GPIO override for VPHCTL. Mobile: GPIO override for GPIO[21]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_VIPCLK	5	0x0	Desktop: GPIO override for VIPCLK. Mobile: GPIO override for GPIO[20]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_VID	15:8	0x0	Desktop: GPIO override for VID[7:0]. Mobile: GPIO override for GPIO[34:27]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_VPCLK0	16	0x0	Desktop: GPIO override for VPCLK0. Mobile: GPIO override for GPIO[24]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_DVALID	17	0x0	Desktop: GPIO override for DVALID. Mobile: GPIO override for GPIO[26]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.
VIPPAD_MASK_PSYNC	18	0x0	Desktop: GPIO override for PSYNC. Mobile: GPIO override for GPIO[25]. 0=Pin not enabled for GPIO 1=Pin enabled for GPIO. Normal function overridden.

Desktop: Multimedia Interface GPIO Mask Control. Mobile: Additional GPIO Interface Mask Control

VIPPAD_A - RW - 32 bits - [GpuF0MMReg:0xAC4]			
Field Name	**Bits**	**Default**	**Description**
VIPPAD_A_SCL	0	0x0	Desktop: Output for SCL. Mobile: Output for GPIO[19]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_SDA	1	0x0	Desktop: Output for SDA. Mobile: Output for GPIO[18]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_VHAD	3:2	0x0	Desktop: Output for VHAD[1:0]. Mobile: Output for GPIO[23:22]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_VPHCTL	4	0x0	Desktop: Output for VPHCTL. Mobile: Output for GPIO[21]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_VIPCLK	5	0x0	Desktop: Output for VIPCLK. Mobile: Output for GPIO[20]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.

VIPPAD_A_VID	15:8	0x0	Desktop: Output for VID[7:0]. Mobile: Output for GPIO[34:27]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_VPCLK0	16	0x0	Desktop: Output for VPCLK0. Mobile: Output for GPIO[24]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_DVALID	17	0x0	Desktop: Output for DVALID. Mobile: Output for GPIO[26]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.
VIPPAD_A_PSYNC	18	0x0	Desktop: Output for PSYNC. Mobile: Output for GPIO[25]. 0=GPIO output is low for this pin, if mask and output are enabled. 1=GPIO output is high for this pin, if mask and output are enabled.

Desktop: Multimedia Interface GPIO Output Control; Mobile: Additional GPIO Interface Output Control

VIPPAD_EN - RW - 32 bits - [GpuF0MMReg:0xAC8]			
Field Name	**Bits**	**Default**	**Description**
VIPPAD_EN_SCL	0	0x0	Desktop: Output enable for SCL. Mobile: Output enable for GPIO[19]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_SDA	1	0x0	Desktop: Output enable for SDA. Mobile: Output enable for GPIO[18]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_VHAD	3:2	0x0	Desktop: Output enable for VHAD[1:0]. Mobile: Output enable for GPIO[23:22]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_VPHCTL	4	0x0	Desktop: Output enable for VPHCTL. Mobile: Output enable for GPIO[21]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_VIPCLK	5	0x0	Desktop: Output enable for VIPCLK. Mobile: Output enable for GPIO[20]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_VID	15:8	0x0	Desktop: Output enable for VID[7:0]. Mobile: Output enable for GPIO[34:27]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_VPCLK0	16	0x0	Desktop: Output enable for VPCLK0. Mobile: Output enable for GPIO[24]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.
VIPPAD_EN_DVALID	17	0x0	Desktop: Output enable for DVALID. Mobile: Output enable for GPIO[26]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.

VIPPAD_EN_PSYNC	18	0x0	Desktop: Output enable for PSYNC. Mobile: Output enable for GPIO[25]. 0=GPIO output is disabled for this pin. 1=GPIO output is enabled for this pin.

Desktop: Multimedia Interface GPIO Output Enable Control; Mobile: Additional GPIO Interface Output Enable Control

VIPPAD_Y - R - 32 bits - [GpuF0MMReg:0xACC]			
Field Name	**Bits**	**Default**	**Description**
VIPPAD_Y_SCL	0	0x0	Desktop: Input readback of SCL. Mobile: Input readback of GPIO[19]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_SDA	1	0x0	Desktop: Input readback of SDA. Mobile: Input readback of GPIO[18]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_VHAD	3:2	0x0	Desktop: Input readback of VHAD[1:0]. Mobile: Input readback of GPIO[23:22]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_VPHCTL	4	0x0	Desktop: Input readback of VPHCTL. Mobile: Input readback of GPIO[21]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_VIPCLK	5	0x0	Desktop: Input readback of VIPCLK. Mobile: Input readback of GPIO[20]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_VID	15:8	0x0	Desktop: Input readback of VID. Mobile: Input readback of GPIO[34:27]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_VPCLK0	16	0x0	Desktop: Input readback of VPCLK0. Mobile: Input readback of GPIO[24]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_DVALID	17	0x0	Desktop: Input readback of DVALID. Mobile: Input readback of GPIO[26]. 0=This pin was low at time of read. 1=This pin was high at time of read.
VIPPAD_Y_PSYNC	18	0x0	Desktop: Input readback of PSYNC. Mobile: Input readback of GPIO[25]. 0=This pin was low at time of read. 1=This pin was high at time of read.

Desktop: Multimedia Interface GPIO Input Readback; Mobile: Additional GPIO Interface Input Readback

VIPPAD_STRENGTH - RW - 32 bits - [GpuF0MMReg:0xAD0]			
Field Name	**Bits**	**Default**	**Description**
I2C_STRENGTH_SN	3:0	0x7	Desktop: NMOS of SCL and SDA. Mobile: NMOS of GPIO[19:18].
I2C_STRENGTH_SP	7:4	0x4	Desktop: PMOS of SCL and SDA. Mobile: PMOS of GPIO[19:18].
VIPHDAT_STRENGTH_SN	11:8	0x7	Desktop: NMOS of VHAD[1:0] and VPHCTL. Mobile: NMOS of GPIO[23:21].

VIPHDAT_STRENGTH_SP	15:12	0x4	Desktop: PMOS of VHAD[1:0] and VPHCTL. Mobile: PMOS of GPIO[23:21].
VIPHCLK_STRENGTH_SN	19:16	0x7	Desktop: NMOS of VIPCLK. Mobile: NMOS of GPIO[20].
VIPHCLK_STRENGTH_SP	23:20	0x4	Desktop: PMOS of VIPCLK. Mobile: PMOS of GPIO[20].
VIDCAP_STRENGTH_SN	27:24	0x7	Desktop: NMOS of VID, VPCLK0, PSYNC, and DVALID. Mobile: NMOS of GPIO[34:24].
VIDCAP_STRENGTH_SP	31:28	0x4	Desktop: PMOS of VID, VPCLK0, PSYNC, and DVALID. Mobile: PMOS of GPIO[34:24].

Desktop: Multimedia Interface GPIO Output Driver Strength; Mobile: Additional GPIO Interface Output Driver Strength

EXTERN_TRIG_CNTL - RW - 32 bits - [GpuF0MMReg:0xE54]			
Field Name	**Bits**	**Default**	**Description**
EXTERN_TRIG_CLR (W)	0	none	External Trigger Clear: Write 0 has no affect. Write 1 sets the external trigger to 0.
EXTERN_TRIG_READ (R)	1	none	External Trigger Status: 0 - Indicates WAIT condition is active. 1 - Indicates WAIT condition is not active.

External Trigger Control

ROM_CNTL - RW - 32 bits - [GpuF0MMReg:0x1600]			
Field Name	**Bits**	**Default**	**Description**
SCK_OVERWRITE	1	0x0	Overwirte the default SCK clock source. 0=SCK sourced from sclk. 1=SCK sourced from crystal clock.
CLOCK_GATING_EN	2	0x0	ROM read controller dynamic clock gating enable. 0=Software disable the dynamic clock going to the read controller 1=Software enable the dynamic clock going to the read controller
CSB_ACTIVE_TO_SCK_SETUP_TIME	15:8	0x3	CSb active to SCK setup time. Programmable delay in number of SCK cycles.
CSB_ACTIVE_TO_SCK_HOLD_TIME	23:16	0x3	CSb active to SCK hold time. Programmable delay in number of SCK cycles. Actual hold time is (this delay + one SCK cycle).
SCK_PRESCALE_REFCLK	27:24	0x1	Programmable SCK divider when clock source is PCIE REFCLK.
SCK_PRESCALE_CRYSTAL_CLK	31:28	0x1	Programmable SCK divider when clock source is on-board crystal clock.

ROM controller control registers.

ROM_STATUS - R - 32 bits - [GpuF0MMReg:0x1608]			
Field Name	**Bits**	**Default**	**Description**
ROM_BUSY	0	0x0	The ROM SPI interface is busy doing transaction.

ROM controller status registers.

ROM_INDEX - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0xA8]			
Field Name	**Bits**	**Default**	**Description**
ROM_INDEX	23:0	0x0	Address in the ROM aperture space. The ROM device physical address is calculated based on the ROM_START register, plus this ROM_INDEX field. The ROM controller will read 4 bytes starting from this address.

Address for indirect read access to ROM.

ROM_DATA - R - 32 bits - [GpuF0MMReg,GpuIOReg:0xAC]			
Field Name	**Bits**	**Default**	**Description**
ROM_DATA	31:0	0x0	Four bytes of data from indirect read access to ROM.

Data from indirect read access to ROM.

ROM_START - RW - 32 bits - [GpuF0MMReg:0x1614]			
Field Name	**Bits**	**Default**	**Description**
ROM_START	23:0	0x0	ROM device starting address that points to the starting of the ROM aperture. This is used by software to read the whole ROM device via ROM aperture when the device size is larger than the aperture size.

ROM device starting address that points to the starting of the ROM aperture. Default to 0x0.

2.6 Video Graphics Array (VGA) Registers

VGA Control Registers

2.6.1 VGA Control/Status Registers

General purpose status VGA

GENMO_WT - W - 8 bits - [GpuF0MMReg,VGA_IO:0x3C2]			
Field Name	**Bits**	**Default**	**Description**
GENMO_MONO_ADDRESS_B	0	0x0	0=Monochrome emulation, regs at 0x3Bx 1=Color/Graphic emulation, regs at 0x3Dx
VGA_RAM_EN	1	0x0	0=Disable 1=Enable
VGA_CKSEL	3:2	0x0	0=25.1744MHz (640 Pels) 1=28.3212MHz (720 Pels) 2=Reserved 3=Reserved
ODD_EVEN_MD_PGSEL	5	0x0	0=Selects odd (high) memory locations 1=Selects even (low) memory locations
VGA_HSYNC_POL	6	0x0	
VGA_VSYNC_POL	7	0x0	

GENMO_RD - R - 8 bits - [GpuF0MMReg,VGA_IO:0x3CC]			
Field Name	**Bits**	**Default**	**Description**
GENMO_MONO_ADDRESS_B *(mirror of GENMO_WT:GENMO_MONO_ADDRESS_B)*	0	0x0	0=Monochrome emulation, regs at 0x3Bx 1=Color/Graphic emulation, regs at 0x3Dx
VGA_RAM_EN *(mirror of GENMO_WT:VGA_RAM_EN)*	1	0x0	0=Disable 1=Enable
VGA_CKSEL *(mirror of GENMO_WT:VGA_CKSEL)*	3:2	0x0	0=25.1744MHz (640 Pels) 1=28.3212MHz (720 Pels) 2=Reserved 3=Reserved
ODD_EVEN_MD_PGSEL *(mirror of GENMO_WT:ODD_EVEN_MD_PGSEL)*	5	0x0	0=Selects odd (high) memory locations 1=Selects even (low) memory locations
VGA_HSYNC_POL *(mirror of GENMO_WT:VGA_HSYNC_POL)*	6	0x0	
VGA_VSYNC_POL *(mirror of GENMO_WT:VGA_VSYNC_POL)*	7	0x0	

GENENB - R - 8 bits - [GpuF0MMReg,VGA_IO:0x3C3]			
Field Name	**Bits**	**Default**	**Description**
BLK_IO_BASE	7:0	0x0	

GENFC_WT - W - 8 bits - [GpuF0MMReg:0x3BA] [GpuF0MMReg:0x3DA] [VGA_IO:0x3BA] [VGA_IO:0x3DA]			
Field Name	**Bits**	**Default**	**Description**
VSYNC_SEL_W	3	0x0	Vertical sync select (write). 0=Normal vertical sync 1=Sync is 'vertical sync' ORed with 'vertical display enable'

Feature Control Register (Read)

GENFC_RD - R - 8 bits - [GpuF0MMReg,VGA_IO:0x3CA]			
Field Name	**Bits**	**Default**	**Description**
VSYNC_SEL_R *(mirror of GENFC_WT:VSYNC_SEL_W)*	3	0x0	Veritcal sync select (read). 0=Normal vertical sync 1=Sync is 'vertical sync' ORed with 'vertical display enable'

Feature Control Regsiter (Read)

GENS0 - R - 8 bits - [GpuF0MMReg,VGA_IO:0x3C2]			
Field Name	**Bits**	**Default**	**Description**
SENSE_SWITCH	4	0x0	DAC comparator read back. Used for monitor detection. Mirror of DAC_CMP_OUTPUT@DAC_CNTL. See description there.
CRT_INTR	7	0x0	CRT Interrupt: 0=Vertical retrace interrupt is cleared 1=Vertical retrace interrupt is pending

Input Status 0 Register

GENS1 - R - 8 bits - [GpuF0MMReg:0x3BA] [GpuF0MMReg:0x3DA] [VGA_IO:0x3BA] [VGA_IO:0x3DA]			
Field Name	**Bits**	**Default**	**Description**
NO_DISPLAY	0	0x0	Display enable. 0=Enable 1=Disable
VGA_VSTATUS	3	0x0	Vertical Retrace Status. 0=Vertical retrace not active 1=Vertical retrace active
PIXEL_READ_BACK	5:4	0x0	Diagnostic bits 0, 1 respectively. These two bits are connected to two of the eight colour 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

Input Status 1 Register

2.6.2 VGA DAC Control Registers

DAC_DATA - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C9]			
Field Name	**Bits**	**Default**	**Description**
DAC_DATA	5:0	0x0	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.

VGA Palette (DAC) Data

DAC_MASK - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C6]			
Field Name	**Bits**	**Default**	**Description**
DAC_MASK	7:0	0x0	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.

Palette index mask for VGA emulation modes.

DAC_R_INDEX - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C7]			
Field Name	**Bits**	**Default**	**Description**
DAC_R_INDEX	7:0	0x0	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). Aslo see DAC_W_INDEX.

Palette (DAC) Read Index

DAC_W_INDEX - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C8]			
Field Name	**Bits**	**Default**	**Description**
DAC_W_INDEX	7:0	0x0	Sets the index for a palette (DAC) write operation. Index auto-increments after every third write of DAC_DATA. Aslo see DAC_R_INDEX.

Palette (DAC) Write Index

2.6.3 VGA Sequencer Registers

SEQ8_IDX - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C4]			
Field Name	**Bits**	**Default**	**Description**
SEQ_IDX	2:0	0x0	

SEQ8_DATA - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C5]			
Field Name	**Bits**	**Default**	**Description**
SEQ_DATA	7:0	0x0	

SEQ00 - RW - 8 bits - VGASEQIND:0x0			
Field Name	**Bits**	**Default**	**Description**
SEQ_RST0B	0	0x1	Synchronous reset bit 0: 0=Follows SEQ_RST1B 1=Sequencer runs unless SEQ_RST1B=0
SEQ_RST1B	1	0x1	Synchronous reset bit 1: 0=Disable character clock, display requests, and H/V syncs 1=Sequencer runs unless SEQ_RST0B=0

Reset Register

SEQ01 - RW - 8 bits - VGASEQIND:0x1			
Field Name	**Bits**	**Default**	**Description**
SEQ_DOT8	0	0x1	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.
SEQ_SHIFT2	2	0x0	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	0x0	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	0x0	Shift load bits. 0=SEQ_SHIFT2 determines serializer loading 1=Load video serializer every fourth clock. Ignore SEQ_SHIFT2
SEQ_MAXBW	5	0x1	Screen off: 0=Normal. Screen on 1=Sreen off and blanked. CPU has uninterrupted access to frame buffer

Clock Mode Register

SEQ02 - RW - 8 bits - VGASEQIND:0x2			
Field Name	**Bits**	**Default**	**Description**
SEQ_MAP0_EN	0	0x0	0=Disable write to memory map 0 1=Enable write to memory map 0
SEQ_MAP1_EN	1	0x0	0=Disable write to memory map 1 1=Enable write to memory map 1
SEQ_MAP2_EN	2	0x0	0=Disable write to memory map 2 1=Enable write to memory map 2
SEQ_MAP3_EN	3	0x0	0=Disable write to memory map 3 1=Enable write to memory map 3

SEQ03 - RW - 8 bits - VGASEQIND:0x3			
Field Name	**Bits**	**Default**	**Description**
SEQ_FONT_B1	0	0x0	Character Map Select B Bit 1
SEQ_FONT_B2	1	0x0	Character Map Select B Bit 2
SEQ_FONT_A1	2	0x0	Character Map Select A Bit 1
SEQ_FONT_A2	3	0x0	Character Map Select A Bit 2
SEQ_FONT_B0	4	0x0	Character Map Select B Bit 0
SEQ_FONT_A0	5	0x0	Character Map Select A Bit 0

Character Map Select Register

SEQ04 - RW - 8 bits - VGASEQIND:0x4			
Field Name	**Bits**	**Default**	**Description**
SEQ_256K	1	0x0	0=64KB memory present. Has no effect since 256KB always available 1=256KB memory present
SEQ_ODDEVEN	2	0x0	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	0x0	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

2.6.4 VGA CRT Registers

CRTC8_IDX - RW - 8 bits - [GpuF0MMReg:0x3B4] [GpuF0MMReg:0x3D4] [VGA_IO:0x3B4] [VGA_IO:0x3D4]

Field Name	Bits	Default	Description
VCRTC_IDX	5:0	0x0	

CRTC8_DATA - RW - 8 bits - [GpuF0MMReg:0x3B5] [GpuF0MMReg:0x3D5] [VGA_IO:0x3B5] [VGA_IO:0x3D5]			
Field Name	**Bits**	**Default**	**Description**
VCRTC_DATA	7:0	0x0	

CRT00 - RW - 8 bits - VGACRTIND:0x0			
Field Name	**Bits**	**Default**	**Description**
H_TOTAL	7:0	0x0	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.

Horizontal Total Register

CRT01 - RW - 8 bits - VGACRTIND:0x1			
Field Name	**Bits**	**Default**	**Description**
H_DISP_END	7:0	0x0	These bits define the active horizontal dispaly in a scan line. The value is one less than the total number of displayed characters in a scan line.

Horizontal Display Enable End Register

CRT02 - RW - 8 bits - VGACRTIND:0x2			
Field Name	**Bits**	**Default**	**Description**
H_BLANK_START	7:0	0x0	These bits define the horizontal character count that represents the character coune in the active display area plus the right borger. In other words, the count is from the start of active display to the start of triggering of the H blanking pulse.

Start Horizontal Blanking Register

CRT03 - RW - 8 bits - VGACRTIND:0x3			
Field Name	**Bits**	**Default**	**Description**
H_BLANK_END	4:0	0x0	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	0x0	Display-enable skew: 0=0Skew 1=1Skew 2=2Skew 3=3Skew
CR10CR11_R_DIS_B	7	0x0	Comptibility Read: 0=WrtOnlyToCRT10-11 1=WrtRdToCRT10-11

End Horizontal Blanking Register

CRT04 - RW - 8 bits - VGACRTIND:0x4			
Field Name	**Bits**	**Default**	**Description**
H_SYNC_START	7:0	0x0	These bits define the horizontal character count at which the horizontal retrace pulse becomes active.

Start Horizontal Retrace Register

CRT05 - RW - 8 bits - VGACRTIND:0x5			
Field Name	**Bits**	**Default**	**Description**
H_SYNC_END	4:0	0x0	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	0x0	H Retrace Delay bits (these two bits skew the horizontal retrace pulse).
H_BLANK_END_B5	7	0x0	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].

End Horizontal Retrace Register

CRT06 - RW - 8 bits - VGACRTIND:0x6			
Field Name	**Bits**	**Default**	**Description**
V_TOTAL	7:0	0x0	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.

Vertical Total Register

CRT07 - RW - 8 bits - VGACRTIND:0x7			
Field Name	**Bits**	**Default**	**Description**
V_TOTAL_B8	0	0x0	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	0x0	End V Display Bit 8 (CRT12). Bit 8 of 10-bit vertical count for V Display enable. For functional desription see CRT12 register.
V_SYNC_START_B8	2	0x0	Start V Retrace Bit 8 (CRT10). Bit 8 of 10-bit veritcal count for V Retrace start. For functional description see CRT10 register.
V_BLANK_START_B8	3	0x0	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	0x0	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	0x0	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	0x0	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	0x0	Start V Retrace Bit (CRT10). Bit 9 of 10-bit vertical count for V Retrace start. For functional description see CRT10 register.

CRTC Overflow Register

CRT08 - RW - 8 bits - VGACRTIND:0x8			
Field Name	**Bits**	**Default**	**Description**
ROW_SCAN_START	4:0	0x0	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	0x0	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).

Preset Row Scan Register

CRT09 - RW - 8 bits - VGACRTIND:0x9			
Field Name	**Bits**	**Default**	**Description**
MAX_ROW_SCAN	4:0	0x0	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	0x0	Start V Blanking bit 9 (CRT15). Bit 9 of 10-bit veritcal count for line compare. For functional description see CRT18 register.
LINE_CMP_B9	6	0x0	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	0x0	200/400 line scan. NOTE H/V display and blanking timings etc. (in CRT00-CRT06 registers) are not affected. 0=200LineScan 1=400LineScan

Maximum Scan Line Register

CRT0A - RW - 8 bits - VGACRTIND:0xA			
Field Name	**Bits**	**Default**	**Description**
CURSOR_START	4:0	0x0	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	0x0	Cursor on/off. 0=on 1=off

Cursor Start Register

CRT0B - RW - 8 bits - VGACRTIND:0xB			
Field Name	**Bits**	**Default**	**Description**
CURSOR_END	4:0	0x0	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	0x0	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.

Cursor End Register

CRT0C - RW - 8 bits - VGACRTIND:0xC			
Field Name	**Bits**	**Default**	**Description**
DISP_START	7:0	0x0	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.

Start Address (High Byte) Register

CRT0D - RW - 8 bits - VGACRTIND:0xD

Field Name	Bits	Default	Description
DISP_START	7:0	0x0	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 creen mode, CRT0C + CRT0D points to the starting location of screen A (top half.) The starting address for screen B is always zero.

Start Address (Low Byte) Register

CRT0E - RW - 8 bits - VGACRTIND:0xE			
Field Name	**Bits**	**Default**	**Description**
CURSOR_LOC_HI	7:0	0x0	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.

Cursor Location (High Byte) Register

CRT0F - RW - 8 bits - VGACRTIND:0xF			
Field Name	**Bits**	**Default**	**Description**
CURSOR_LOC_LO	7:0	0x0	CA bits 7:0- 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

Cursor Location (Low Byte) Register

CRT10 - RW - 8 bits - VGACRTIND:0x10			
Field Name	**Bits**	**Default**	**Description**
V_SYNC_START	7:0	0x0	Bits CRT10[7:0] are the eight low-order bits of the 10-bit vertical retrace start count. The two high-order bits are CRTt07[2:7], located in the CRTC overflow register.- These bits define the horizontal scan count that triggers the V retrace pulse.

Start Vertical Retrace Register

CRT11 - RW - 8 bits - VGACRTIND:0x11			
Field Name	**Bits**	**Default**	**Description**
V_SYNC_END	3:0	0x0	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	0x0	V Retrace Interrupt Set: 0=VRetraceIntCleared 1=Not Cleared
V_INTR_EN	5	0x0	V Retrace Interrupt Disabled: 0=VRetraceIntEna 1=Disable
SEL5_REFRESH_CYC	6	0x0	0=3 DRAM Refresh/Horz Line 1=5 DRAM Refresh/Horz Line
C0T7_WR_ONLY	7	0x0	Write Protect (CRT00-CRT06). All register bits except CRTO7[4] are write protected. 0=EnaWrtToCRT00-07 1=C0T7B4WrtOnly

End Vertical Retrace Register

CRT12 - RW - 8 bits - VGACRTIND:0x12			
Field Name	**Bits**	**Default**	**Description**
V_DISP_END	7:0	0x0	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.

Vertical Display Enable End Register

CRT13 - RW - 8 bits - VGACRTIND:0x13			
Field Name	**Bits**	**Default**	**Description**
DISP_PITCH	7:0	0x0	- 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.

Offset Register

CRT14 - RW - 8 bits - VGACRTIND:0x14			
Field Name	**Bits**	**Default**	**Description**
UNDRLN_LOC	4:0	0x0	
ADDR_CNT_BY4	5	0x0	0=Char. Clock 1=CountBy4
DOUBLE_WORD	6	0x0	0=Disable 1=DoubleWordMdEna

CRT15 - RW - 8 bits - VGACRTIND:0x15			
Field Name	**Bits**	**Default**	**Description**
V_BLANK_START	7:0	0x0	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 blaning pulse, in units of horizontal scan lines. The value is equal to the actual number of displayed lines minus one.

Start Vertical Blanking Register

CRT16 - RW - 8 bits - VGACRTIND:0x16			
Field Name	**Bits**	**Default**	**Description**
V_BLANK_END	7:0	0x0	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 storeed 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.

End Vertical Blanking Register

CRT17 - RW - 8 bits - VGACRTIND:0x17			
Field Name	**Bits**	**Default**	**Description**
RA0_AS_A13B	0	0x0	
RA1_AS_A14B	1	0x0	
VCOUNT_BY2	2	0x0	
ADDR_CNT_BY2	3	0x0	
WRAP_A15TOA0	5	0x0	
BYTE_MODE	6	0x0	0=WordMode 1=ByteMode
CRTC_SYNC_EN	7	0x0	0=Disable HVSync 1=EnaHVSync

CRT18 - RW - 8 bits - VGACRTIND:0x18			
Field Name	**Bits**	**Default**	**Description**
LINE_CMP	7:0	0x0	- 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].)

Line Compare Register

CRT1E - R - 8 bits - VGACRTIND:0x1E			
Field Name	Bits	Default	Description
GRPH_DEC_RD1	1	0x0	

CRT1F - R - 8 bits - VGACRTIND:0x1F			
Field Name	Bits	Default	Description
GRPH_DEC_RD0	7:0	0x0	

CRT22 - R - 8 bits - VGACRTIND:0x22			
Field Name	Bits	Default	Description
GRPH_LATCH_DATA	7:0	0x0	

2.6.5 VGA Graphics Registers

GRPH8_IDX - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3CE]			
Field Name	Bits	Default	Description
GRPH_IDX	3:0	0x0	

GRPH8_DATA - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3CF]			
Field Name	Bits	Default	Description
GRPH_DATA	7:0	0x0	

GRA00 - RW - 8 bits - VGAGRPHIND:0x0			
Field Name	Bits	Default	Description
GRPH_SET_RESET0	0	0x0	

GRPH_SET_RESET1	1	0x0	
GRPH_SET_RESET2	2	0x0	
GRPH_SET_RESET3	3	0x0	

GRA01 - RW - 8 bits - VGAGRPHIND:0x1			
Field Name	**Bits**	**Default**	**Description**
GRPH_SET_RESET_ENA0	0	0x0	
GRPH_SET_RESET_ENA1	1	0x0	
GRPH_SET_RESET_ENA2	2	0x0	
GRPH_SET_RESET_ENA3	3	0x0	

GRA02 - RW - 8 bits - VGAGRPHIND:0x2			
Field Name	**Bits**	**Default**	**Description**
GRPH_CCOMP	3:0	0x0	

GRA03 - RW - 8 bits - VGAGRPHIND:0x3			
Field Name	**Bits**	**Default**	**Description**
GRPH_ROTATE	2:0	0x0	
GRPH_FN_SEL	4:3	0x0	0=Replace 1=AND 2=OR 3=XOR

GRA04 - RW - 8 bits - VGAGRPHIND:0x4			
Field Name	**Bits**	**Default**	**Description**
GRPH_RMAP	1:0	0x0	

GRA05 - RW - 8 bits - VGAGRPHIND:0x5			
Field Name	**Bits**	**Default**	**Description**
GRPH_WRITE_MODE	1:0	0x0	0=Write mode 0 1=Write mode 1 2=Write mode 2 3=Write mode 3
GRPH_READ1	3	0x0	0=Read mode 0, byte oriented 1=Read mode 1, pixel oriented

CGA_ODDEVEN	4	0x0	0=Disable Odd/Even Addressing 1=Enable Odd/Even Addressing
GRPH_OES	5	0x0	0=Linear shift mode 1=Tiled shift mode
GRPH_PACK	6	0x0	0=Use shift register mode as per GRPH_OES 1=256 color mode, read as packed pixels, ignore GRPH_OES

GRA06 - RW - 8 bits - VGAGRPHIND:0x6			
Field Name	**Bits**	**Default**	**Description**
GRPH_GRAPHICS	0	0x0	0=Alpha Numeric Mode 1=Graphics Mode
GRPH_ODDEVEN	1	0x0	0=Normal 1=Chain Odd maps to Even
GRPH_ADRSEL	3:2	0x0	0=A0000-128K 1=A0000-64K 2=B0000-32K 3=B8000-32K

GRA07 - RW - 8 bits - VGAGRPHIND:0x7			
Field Name	**Bits**	**Default**	**Description**
GRPH_XCARE0	0	0x0	0=Ignore map 0 1=Use map 0 for read mode 1
GRPH_XCARE1	1	0x0	0=Ignore map 1 1=Use map 1 for read mode 1
GRPH_XCARE2	2	0x0	0=Ignore map 2 1=Use map 2 for read mode 1
GRPH_XCARE3	3	0x0	0=Ignore map 3 1=Use map 3 for read mode 1

GRA08 - RW - 8 bits - VGAGRPHIND:0x8			
Field Name	**Bits**	**Default**	**Description**
GRPH_BMSK	7:0	0x0	

2.6.6 VGA Attribute Registers

ATTRX - RW - 8 bits - [GpuF0MMReg,VGA_IO:0x3C0]			
Field Name	**Bits**	**Default**	**Description**
ATTR_IDX	4:0	0x0	ATTR Index. This index points to one of the internal registers of the attribute controller (ATTR) at addresses 0x3C1/0x3C0, 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 cna be initialized to point to the index register by a read of GENS1.
ATTR_PAL_RW_ENB	5	0x0	Palette Address Source. After loading the colour palette, this bit should be set to logical 1. 0=Processor to load 1=Memory data to access

Attribute Index Register

ATTRDW - W - 8 bits - [GpuF0MMReg,VGA_IO:0x3C0]			
Field Name	**Bits**	**Default**	**Description**
ATTR_DATA	7:0	0x0	Attribute Data Write

Attribute Data Write Register

ATTRDR - R - 8 bits - [GpuF0MMReg,VGA_IO:0x3C1]			
Field Name	**Bits**	**Default**	**Description**
ATTR_DATA	7:0	0x0	Attribute Data Read

Attribute Data Read Register

ATTR00 - RW - 8 bits - VGAATTRIND:0x0			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 0

ATTR01 - RW - 8 bits - VGAATTRIND:0x1			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 1

ATTR02 - RW - 8 bits - VGAATTRIND:0x2			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 2

ATTR03 - RW - 8 bits - VGAATTRIND:0x3			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 3

ATTR04 - RW - 8 bits - VGAATTRIND:0x4			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 4

ATTR05 - RW - 8 bits - VGAATTRIND:0x5			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 5

ATTR06 - RW - 8 bits - VGAATTRIND:0x6			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 6

ATTR07 - RW - 8 bits - VGAATTRIND:0x7			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 7

ATTR08 - RW - 8 bits - VGAATTRIND:0x8			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 8

ATTR09 - RW - 8 bits - VGAATTRIND:0x9			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register 9

ATTR0A - RW - 8 bits - VGAATTRIND:0xA			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register Ah (10)

ATTR0B - RW - 8 bits - VGAATTRIND:0xB			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register Bh (11)

ATTR0C - RW - 8 bits - VGAATTRIND:0xC			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register Ch (12)

ATTR0D - RW - 8 bits - VGAATTRIND:0xD			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register Dh (13)

ATTR0E - RW - 8 bits - VGAATTRIND:0xE			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register Eh (14)

ATTR0F - RW - 8 bits - VGAATTRIND:0xF			
Field Name	**Bits**	**Default**	**Description**
ATTR_PAL	5:0	0x0	Colour Bits 5:0 map the text attribute or graphics colour input value to a display colour on the screen. Colour is disabled for those bits that are set to logical 0; enabled for those bits set to logical 1.

Palette Register Fh (15)

ATTR10 - RW - 8 bits - VGAATTRIND:0x10			
Field Name	**Bits**	**Default**	**Description**

ATTR_GRPH_MODE	0	0x0	Graphics/Alphanumeric Mode. 0=Alphanumeric Mode 1=Graphic Mode
ATTR_MONO_EN	1	0x0	Monochrome/Colour Attributes Select: 0=Color Disp 1=Monochrome Disp
ATTR_LGRPH_EN	2	0x0	Line Graphics Enable. Must be 0 for character fonts that do not use line graphics character codes for graphics. Zero will force the 9th dot to the background colour. One will allow the 8th bit of the line graphics characters to be stretched to the 9th dot. 0=Disable line graphics 8th dot stretch 1=Enable line graphics 8th dot stretch
ATTR_BLINK_EN	3	0x0	Blink Enable/Background Intensity: Selects whether bit 7 of the attribute controls intensity or blinking. 0=Intensity control 1=Blink control
ATTR_PANTOPONLY	5	0x0	PEL Panning Compatibility: 0=Pan both halves of the screen 1=Pan only the top half screen
ATTR_PCLKBY2	6	0x0	PEL Clock Select: 0=Shift register clocked every dot clock 1=For mode 13 (256 colour), 8 bits packed to form a pixel
ATTR_CSEL_EN	7	0x0	Alternate Colour Source: 0=Select ATTR00-0F bit 5:4 as P5 and P4 1=Select ATTR14 bit 1:0 as P5 and P4

Mode Control Register

ATTR11 - RW - 8 bits - VGAATTRIND:0x11			
Field Name	**Bits**	**Default**	**Description**
ATTR_OVSC	7:0	0x0	Overscan Colour

Overscan Colour Register

ATTR12 - RW - 8 bits - VGAATTRIND:0x12			
Field Name	**Bits**	**Default**	**Description**
ATTR_MAP_EN	3:0	0x0	Enable Colour 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.
ATTR_VSMUX	5:4	0x0	Video Status Mux bits 1:0. These are control bits for the multiplexer on colour 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

Colour Map Enable Register

ATTR13 - RW - 8 bits - VGAATTRIND:0x13			
Field Name	**Bits**	**Default**	**Description**
ATTR_PPAN	3:0	0x0	Shift Count Bits 3:0. The shift count value (0-8) indicates how many pixle positions to shift left. Shift in respective modes Count Value: 0+,1+,2+,13 / All other 3+,7,7+ 0: 1 / 0 1: 2 / - 2: 3 / 1 3: 4 / - 4: 5 / 2 5: 6 / - 6: 7 / 3 7: 8 / - 8: 0 / -

Horizontal PEL Panning Register

ATTR14 - RW - 8 bits - VGAATTRIND:0x14			
Field Name	**Bits**	**Default**	**Description**
ATTR_CSEL1	1:0	0x0	Colour bits P5 and P4, respectively. These are the colour output bits (instead of bits 5 and 4 of the internal palette registers ATTR00-0F) when alternate colour source, bit ATTR10[7] is logical 1.
ATTR_CSEL2	3:2	0x0	Colour bits P7 and P6, respectively. These two bits are the two high-order bits of the 8-bit colour, used for rapid colour set switching (addressing different parts of the DAC colour lookup table). The lower order bits are in registers ATTR00-0F.

Colour Select Register

2.6.7 VGA Miscellaneous Registers

VGA_RENDER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x300]			
Field Name	**Bits**	**Default**	**Description**
VGA_BLINK_RATE	4:0	0xf	One less than the number of frames that the cursor remains OFF = one less than the number of frames that the cursor remains ON = one less than half the cursor blink period = one less than a quarter of the character blink period. If register set to 0 test mode will happen, blink counter is reset and VGA_BLINK_MODE is followed, if set to 1, as an exception, cursor blink will be ON one frame, OFF one frame, if set to 2, cursor blink will be ON three frames, OFF three frames, etc

VGA_BLINK_MODE	6:5	0x0	Determines whether the blinking sequence starts with blinking characters and cursor visible or invisible. If VGA_BLINK_RATE = 0 the frame remains static at the start of the sequence. 0=Blinking sequence starts with blinking characters visible and cursor visible 1=Blinking sequence starts with blinking characters visible and cursor invisible 2=Blinking sequence starts with blinking characters invisible and cursor visible 3=Blinking sequence starts with blinking characters invisible and cursor invisible
VGA_CURSOR_BLINK_INVERT	7	0x0	Determines if the blinking characters toggle when the cursor toggles from invisible to visible (default) or when the cursor toggles from visible to invisible 0=Sequence is (regardless of where it starts) : blinking chars visible and cursor visible, blinking chars visible and cursor invisible, blinking chars invisible and cursor visible, blinking chars invisible and cursor invisible, blinking chars visible and cursor visible, ... etc . The starting point in the sequence is determined by VGA_BLINK_MODE 1=Sequence is (regardless of where it starts) : blinking chars visible and cursor visible, blinking chars invisible and cursor invisible, blinking chars invisible and cursor visible, blinking chars visible and cursor invisible, blinking chars visible and cursor visible, ... etc. The starting point in the sequence is determined by VGA_BLINK_MODE
VGA_EXTD_ADDR_COUNT_ENABLE	8	0x0	Determines if the render will allow reading beyond 256K 0=Disable 1=Enable Extended Address Counter beyond 256K
VGA_VSTATUS_CNTL	17:16	0x0	controls the main state machine of the VGA render 0=VGA render disable (no VGA engine trigger enabled) 1=Use CRTC1 vblank to trigger VGA engine 2=Use CRTC2 vblank to trigger VGA engine 3=Use both CRTC1 and CRTC2 vblank to trigger VGA engine
VGA_LOCK_8DOT	24	0x0	Determines if 9 dot text characters will be allowed or not 0=respect SEQ_DOT8 value 1=Force SEQ_DOT8 =1, VGA_CKSEL = 0 for functionality
VGAREG_LINECMP_COMPATIBILITY_SEL	25	0x0	Selects point at which line compare is activated 0=line==line_cmp(default). As per VGA specification 1=line>line_cmp. As per legacy ATI VGA controllers

VGA Render control Register

VGA_SEQUENCER_RESET_CONTROL - RW - 32 bits - [GpuF0MMReg:0x304]			
Field Name	**Bits**	**Default**	**Description**
D1_BLANK_DISPLAY_WHEN_SEQUENCER_RESET	0	0x1	controls wheter to blank the display 1 in a sequencer reset 0=Reseting Sequencer (SEQ00:SEQ_RST) has no effect on Display Controller 1 1=Reseting Sequencer (SEQ00:SEQ_RST) blanks the output of Display Controller 1
D2_BLANK_DISPLAY_WHEN_SEQUENCER_RESET	4	0x1	controls wheter to blank the display 1 in a sequencer reset 0=Reseting Sequencer (SEQ00:SEQ_RST) has no effect on Display Controller 2 1=Reseting Sequencer (SEQ00:SEQ_RST) blanks the output of Display Controller 2

D1_DISABLE_SYNCS_AND_DE_WHEN_SEQUENCER_RESET	8	0x1	controls wheter to disable syncs for display 1 in a sequencer reset 0=Reseting Sequencer (SEQ00:SEQ_RST) has no effect on Display Controller 1 1=Reseting Sequencer (SEQ00:SEQ_RST) disables HSync, VSync, and DE on Display Controller 2
D2_DISABLE_SYNCS_AND_DE_WHEN_SEQUENCER_RESET	12	0x1	controls wheter to disable syncs for display 2 in a sequencer reset 0=Reseting Sequencer (SEQ00:SEQ_RST) has no effect on Display Controller 2 1=Reseting Sequencer (SEQ00:SEQ_RST) disables HSync, VSync, and DE on Display Controller 2
VGA_MODE_AUTO_TRIGGER_ENABLE	16	0x0	enables the auto-trigger of the VGA mode in a VGA register write 0=disable the auto-trigger mode 1=enable the auto-trigger mode
VGA_MODE_AUTO_TRIGGER_REGISTER_SELECT	17	0x0	selects which register write to use for VGA mode auto-trigger 0=GENFC_WT is used for auto-trigger 1=CRTC_DATA is used for auto-trigger, see VGA_MODE_ENABLE_AUTO_TRIGGER_INDEX_SELECT
VGA_MODE_AUTO_TRIGGER_INDEX_SELECT	23:18	0x0	Selects which CRTC register write will trigger VGA mode

VGA sequencer reset control Register

VGA_MODE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x308]			
Field Name	**Bits**	**Default**	**Description**
VGA_ATI_LINEAR	0	0x0	Sets linear mode for VESA modes 0=Disable 1=Enable
VGA_LUT_PALETTE_UPDATE_MODE	5:4	0x0	Determines how VGA DAC palette updates affect the LUT palette 0=VGA DAC palette writes do not update LUT palette 1=VGA DAC palette writes updata LUTA palette 2=VGA DAC palette writes update LUTB palette 3=reserved
VGA_128K_APERTURE_PAGING	8	0x0	Controls wether the B0000 to BFFFF aperture will wrap on top of the A0000 to AFFFF aperture 0=Normal 1=Enable
VGA_TEXT_132_COLUMNS_EN	16	0x0	Controls 132 column text 0=inActive 1=Active

VGA mode control register

VGA_SURFACE_PITCH_SELECT - RW - 32 bits - [GpuF0MMReg:0x30C]			
Field Name	**Bits**	**Default**	**Description**

VGA_SURFACE_PITCH_SELECT	1:0	0x2	Selects the pitch of the display buffer 0=768 pixels 1=1024 pixels 2=1280 pixels 3=1408 pixels
VGA_SURFACE_HEIGHT_SELECT	9:8	0x0	Selects the height of the display buffer 0=768 lines 1=1024 lines 2=1280 lines 3=1408 lines

display buffer pitch Register

VGA_MEMORY_BASE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x310]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEMORY_BASE_ADDRESS	31:0	0x0	Base address of the 32 Meg area that the VGAHDP and VGARENDER access NOTE: Bits 0:24 of this field are hardwired to ZERO.

VGA Base address Register

VGA_DISPBUF1_SURFACE_ADDR - RW - 32 bits - [GpuF0MMReg:0x318]			
Field Name	**Bits**	**Default**	**Description**
VGA_DISPBUF1_SURFACE_ADDR	24:0	0x0	Base address of display 1 buffer within the 32 Meg defined by VGA_MEMORY_BASE_ADDRESS NOTE: Bits 0:19 of this field are hardwired to ZERO.

display 1 buffer base address

VGA_DISPBUF2_SURFACE_ADDR - RW - 32 bits - [GpuF0MMReg:0x320]			
Field Name	**Bits**	**Default**	**Description**
VGA_DISPBUF2_SURFACE_ADDR	24:0	0x0	Base address of display 2 buffer within the 32 Meg defined by VGA_MEMORY_BASE_ADDRESS NOTE: Bits 0:19 of this field are hardwired to ZERO.

display 2 buffer base address

VGA_HDP_CONTROL - RW - 32 bits - [GpuF0MMReg:0x328]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_PAGE_SELECT_EN	0	0x0	Enables write and read paging 0=Don't use VGA_MEM_WRITE_PAGE_ADDR and VGA_MEM_READ_PAGE_ADDR registers 1=Use VGA_MEM_WRITE_PAGE_ADDR and VGA_MEM_READPAGE_ADDR registers

VGA_MEMORY_DISABLE	4	0x0	Disables the VGA memory: required by Longhorn 0=Do not disable 1=ignore writes and return zero for the reads without affecting the read latch
VGA_RBBM_LOCK_DISABLE	8	0x0	Disables the lock that holds register writes while the memory pipe is full 0=The RBBM write requests will be held untile the data pipe is idle. 1=The RBBM write requests will not be held.
VGA_SOFT_RESET	16	0x0	Does soft reset for VGA, does not reset the registers 0=VGA running in normal operating mode 1=Soft Reset to VGA
VGA_TEST_RESET_CONTROL	24	0x0	Not used

VGAHDP control register

VGA_CACHE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x32C]			
Field Name	**Bits**	**Default**	**Description**
VGA_WRITE_THROUGH_CACHE_DIS	0	0x0	Disables the snooping of memory writes into the read buffer 0=Writes that hit the read cache will update it 1=Writes will invalidate the read cache
VGA_READ_CACHE_DISABLE	8	0x0	Disables the read buffer 0=reads taken from cache, if possible. 1=reads always sent to memory.
VGA_READ_BUFFER_INVALIDATE	16	0x0	Everytime this bit is written with a '1' the VGA read buffer invalidates for coherency purposes
VGA_DCCIF_W256ONLY	20	0x0	Controls whether the write requests from VGADCC to MH will be always 256 bits or optimized for 128 or 256 bit 0=Optimized for 128 or 256 bits 1=Always 256 bits
VGA_DCCIF_WC_TIMEOUT	29:24	0x0	DCCIF write combiner timeout. If there is write inactivity, this field defines the number of SCLKs to wait before flushing write combiner. Minimun value is 9.

VGAHDP caching and VGADCCIF write combining control register

D1VGA_CONTROL - RW - 32 bits - [GpuF0MMReg:0x330]			
Field Name	**Bits**	**Default**	**Description**
D1VGA_MODE_ENABLE	0	0x0	Controls whether display 1 serves the VGA or not 0=VGA display 1 disabled 1=VGA display 1 enabled
D1VGA_TIMING_SELECT	8	0x0	Controls whether display 1 uses the VGA or extended timing parameters 0=display 1 uses extended timing 1=display 1 uses VGA timing
D1VGA_SYNC_POLARITY_SELECT	9	0x0	Controls whether display 1 uses the VGA or extended sync polarities 0=display 1 uses extended sync polarity 1=display 1 uses VGA sync polarity
D1VGA_OVERSCAN_TIMING_SELECT	10	0x1	Controls whether display 1 uses the VGA or extended overscan timing. Only followed if D1VGA_TIMING_SELECT=1 0=display 1 uses extended overscan timing 1=display 1 uses VGA overscan timing

D1VGA_OVERSCAN_COLOR_EN	16	0x0	Controls whether display 1 uses the VGA or extended overscan color 0=display 1 uses CRTC register for overscan color 1=display 1 uses VGA register for overscan color
D1VGA_ROTATE	25:24	0x0	Controls rotation, only looked at if D1VGA_TIMING_SELECT =0 0=no rotation, displays do not interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters 1=rotation 90 degrees, displays do interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters 2=rotation 180 degrees, displays do not interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters 3=rotation 270 degrees, displays do interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters

VGA-Display1 interface control register

D2VGA_CONTROL - RW - 32 bits - [GpuF0MMReg:0x338]			
Field Name	**Bits**	**Default**	**Description**
D2VGA_MODE_ENABLE	0	0x0	Controls whether display 2 serves the VGA or not 0=VGA display 2 disabled 1=VGA display 2 enabled
D2VGA_TIMING_SELECT	8	0x0	Controls whether display 2 uses the VGA or extended timing parameters 0=display 2 uses extended timing 1=display 2 uses VGA timing
D2VGA_SYNC_POLARITY_SELECT	9	0x0	Controls whether display 2 uses the VGA or extended sync polarities 0=display 2 uses extended sync polarity 1=display 2 uses VGA sync polarity
D2VGA_OVERSCAN_TIMING_SELECT	10	0x1	Controls whether display 2 uses the VGA or extended overscan timing. Only followed if D2VGA_TIMING_SELECT=1 0=display 2 uses extended overscan timing 1=display 2 uses VGA overscan timing
D2VGA_OVERSCAN_COLOR_EN	16	0x0	Controls whether display 2 uses the VGA or extended overscan color 0=display 2 uses CRTC register for overscan color 1=display 2 uses VGA register for overscan color
D2VGA_ROTATE *(mirror of D1VGA_CONTROL:D1VGA_ROTATE)*	25:24	0x0	Controls rotation, only looked at if D2VGA_TIMING_SELECT=0 0=no rotation, displays do not interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters 1=rotation 90 degrees, displays do interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters 2=rotation 180 degrees, displays do not interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters 3=rotation 270 degrees, displays do interchange VGA_DISP_h_disp_width and VGA_DISP_v_disp_height parameters

VGA-Display2 interface control register

VGA_HW_DEBUG - RW - 32 bits - [GpuF0MMReg:0x33C]			
Field Name	**Bits**	**Default**	**Description**
VGA_HW_DEBUG	31:0	0x0	

VGA_STATUS - RW - 32 bits - [GpuF0MMReg:0x340]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_ACCESS_STATUS (R)	0	0x0	Memory access status 0=No event 1=Event has occurred, interrupting if enabled
VGA_REG_ACCESS_STATUS (R)	1	0x0	Register access status 0=No event 1=Event has occurred, interrupting if enabled
VGA_DISPLAY_SWITCH_STATUS (R)	2	0x0	Display switch status 0=No event 1=Event has occurred, interrupting if enabled
VGA_MODE_AUTO_TRIGGER_STATUS (R)	3	0x0	VGA mode auto trigger status 0=No event 1=Event has occurred, interrupting if enabled

VGA status register

VGA_INTERRUPT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x344]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_ACCESS_INT_MASK	0	0x0	Enables the interrupt for the Memory access status 0=Disable the interrupt which is set when VGA memory is written or read 1=Enable the interrupt which is set when VGA memory is written or read
VGA_REG_ACCESS_INT_MASK	8	0x0	Enables the interrupt for the register access status 0=Disable the interrupt which is set when the standard VGA registers are written or read 1=Enable the interrupt which is set when the standard VGA registers are written or read
VGA_DISPLAY_SWITCH_INT_MASK	16	0x0	Enables the interrupt for the Display switch status 0=Disable the interrupt which is set when the VGA render switches display buffers 1=Enable the interrupt which is set when the VGA render switches display buffers
VGA_MODE_AUTO_TRIGGER_INT_MASK	24	0x0	Enables the interrupt for VGA mode auto trigger 0=Disable the interrupt which is set when VGA mode is auto-triggered 1=Enable the interrupt which is set when VGA mode is auto-triggered

VGA interrupt mask register

VGA_STATUS_CLEAR - RW - 32 bits - [GpuF0MMReg:0x348]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_ACCESS_INT_CLEAR (W)	0	0x0	Clears the Memory access interrupt 0=No effect 1=Clear status
VGA_REG_ACCESS_INT_CLEAR (W)	8	0x0	Clears the register access interrupt 0=No effect 1=Clear status
VGA_DISPLAY_SWITCH_INT_CLEAR (W)	16	0x0	Clears the display switch interrupt 0=No effect 1=Clear status
VGA_MODE_AUTO_TRIGGER_INT_CLEAR (W)	24	0x0	Clears the VGA mode auto trigger interrupt 0=No effect 1=Clear status

VGA interrupt clear register

VGA_INTERRUPT_STATUS - RW - 32 bits - [GpuF0MMReg:0x34C]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_ACCESS_INT_STATUS (R)	0	0x0	Memory access interrupt status 0=No event 1=Event has occurred
VGA_REG_ACCESS_INT_STATUS (R)	1	0x0	Register access interrupt status 0=No event 1=Event has occurred
VGA_DISPLAY_SWITCH_INT_STATUS (R)	2	0x0	Display switch interrupt status 0=No event 1=Event has occurred
VGA_MODE_AUTO_TRIGGER_INT_STATUS (R)	3	0x0	VGA mode auto trigger interrupt status 0=No event 1=Event has occurred

VGA Interrupt status register

VGA_MAIN_CONTROL - RW - 32 bits - [GpuF0MMReg:0x350]			
Field Name	**Bits**	**Default**	**Description**
VGA_CRTC_TIMEOUT	1:0	0x0	Controls whether and in what conditions the vga crtc calculations will be forced to start if the VBLANK from display takes too long to come 0=VGACRTC times out and is restarted after 1/50 sec without VBLANK 1=VGACRTC times out and is restarted after 1/10 sec without VBLANK 2=reserved 3=VGACRTC does not timeout
VGA_RENDER_TIMEOUT_COUNT	4:3	0x3	Controls whether and in how many display frames the vga render will be forced to finish or timeout 0=No timeout 1=2 frame 2=3 frames 3=4 frames

VGA_VIRTUAL_VERTICAL_RETRACE_DURATION	7:5	0x0	specifies the duration of the vga main state machine of the vga render virtual vertical retrace 0=256 us 1=512 us 2=768 us 3=1024 us 4=1280 us 5=1536 us 6=1792 us 7=2048 us
VGA_READBACK_VGA_VSTATUS_SOURCE_SELECT	9:8	0x0	selects the source for the VGA_VSTATUS readback register bit 0=Uses vga main render state machine virtual vertical retrace - a timer is used to make the duration equivalent as specified by VGA_VIRTUAL_VERTICAL_RETRACE_DURATION 1=reserved 2=Uses CRTC1 vblank signal 3=Uses CRTC2 vblank signal
VGA_READBACK_NO_DISPLAY_SOURCE_SELECT	17:16	0x0	selects the source for the NO_DISPLAY readback register bit 0=Uses vga main render state machine virtual vertical retrace - a timer is used to make the duration specified by VGA_VIRTUAL_VERTICAL_RETRACE_DURATION. Outside of the virtual vertical retrace we have a 31.25 KHz, 5/32 duty cycle pulse train generated independently by a timer asynchronous to the virtual vertical retrace, roughly equivalent to standard horizontal retrace times in standard VGA timings 1=Uses the time the vga render is not rendering. Outside of this time we have a 31.25 kHz pulse train of 5/32 duty cycle, asynchronous to the time the render is rendering and generated independently 2=Uses CRTC1 nodisplay signal 3=Uses CRTC2 nodisplay signal
VGA_READBACK_CRT_INTR_SOURCE_SELECT	25:24	0x0	selects the source for the CRT_INTR readback register bit and associated interrupt 0=Uses vga main render state machine virtual vertical retrace 1=reserved 2=Uses CRTC1 vblank signal 3=Uses CRTC2 vblank signal
VGA_READBACK_SENSE_SWITCH_SELECT	26	0x0	selects the source for the SENSE_SWITCH readback register bit 0=Uses CRTC1 sense_switch signal 1=Uses CRTC2 sense_switch signal
VGA_READ_URGENT_ENABLE	27	0x0	Urgent/stall bit for vga hdp and vga render reads 0=vga hdp and vga render reads not urgent 1=vga hdp and vga render reads urgent
VGA_WRITES_URGENT_ENABLE	28	0x0	0=vga hdp and vga render writes not urgent 1=vga hdp and vga render writes urgent

VGA_MAIN_TEST_VSTATUS_NO_DISPLAY_CRTC_TIMEOUT	31	0x0	For testing purposes, makes the virtual vertical retrace, the crtc timeout and the virtual no display horizontal pulses faster by using the engine clock frequency instead of 1MHz reference 0=VGACRTC timeout is as indicated by VGA_CRTC_TIMEOUT, virtual vertical retrace duration is as indicated by VGA_VIRTUAL_VERTICAL_RETRACE_DURATION, virtual no display horizontal pulses are 31.25 KHz if VGA_READBACK_NO_DISPLAY_SOURCE_SELECT is zero 1=VGACRTC timeout is one 400th of what is indicated by VGA_CRTC_TIMEOUT, virtual vertical retrace duration one 400th of what is indicated by VGA_VIRTUAL_VERTICAL_RETRACE_DURATION, virtual no display horizontal pulses are 400*31.25 KHz if VGA_READBACK_NO_DISPLAY_SOURCE_SELECT is zero

VGA Main control

VGA_TEST_CONTROL - RW - 32 bits - [GpuF0MMReg:0x354]			
Field Name	**Bits**	**Default**	**Description**
VGA_TEST_ENABLE	0	0x0	Controls wether the vga render looks at vertical blank signals from the displays to start rendering or will start through a register write 0=Render responds to status signals from DISP1, DISP2 1=Render responds to VGA_TEST_RENDER_START
VGA_TEST_RENDER_START	8	0x0	Starts the vga render 0=No event 1=Every time this is written with a high, if VGA_TEST_ENABLE is set, VGA Rendering starts
VGA_TEST_RENDER_DONE (R)	16	0x0	Signals when the vga render is done rendering 0=No event 1=If VGA_TEST_ENABLE is set, VGA Rendering is done
VGA_TEST_RENDER_DISPBUF_SELECT	24	0x0	Selects to which display buffer the render will render in test mode (VGA_TEST_ENABLE=1) 0=VGA Render will write into DISPBUF1 starting at VGA_DISPBUF1_SURFACE_ADDR 1=VGA Render will write into DISPBUF2 starting at VGA_DISPBUF2_SURFACE_ADDR

VGA test control register

VGA_DEBUG_READBACK_INDEX - RW - 32 bits - [GpuF0MMReg:0x358]			
Field Name	**Bits**	**Default**	**Description**
VGA_DEBUG_READBACK_INDEX	7:0	0x0	Index for the VGA debug readback

VGA debug readback index register

VGA_DEBUG_READBACK_DATA - RW - 32 bits - [GpuF0MMReg:0x35C]

Field Name	Bits	Default	Description
VGA_DEBUG_READBACK_DATA (R)	31:0	0x0	According to the value of VGA_DEBUG_READBACK_INDEX, VGA_DEBUG_READBACK_DATA will have this values: 0: VGAREG_DISP_h_total[10:0] 1: VGAREG_DISP_h_sync_end[10:0] 2: VGAREG_DISP_h_disp_start[10:0] 3: VGAREG_DISP_h_disp_width[10:0] 4: VGAREG_DISP_h_blank_start[10:0] 5: VGAREG_DISP_h_blank_end[10:0] 6: VGAREG_DISP_v_total[10:0] 7: VGAREG_DISP_v_sync_end[10:0] 8: VGAREG_DISP_v_disp_start[10:0] 9: VGAREG_DISP_v_disp_height[10:0] 10: VGAREG_DISP_v_blank_start[10:0] 11: VGAREG_DISP_v_blank_end[10:0] 12: VGAREG_DISP_overscan_colorR[5:0] 13: VGAREG_DISP_overscan_colorG[5:0] 14: VGAREG_DISP_overscan_colorB[5:0] 15: reserved 16 VGA_DISP_viewport_x_start 17 VGA_DISP_viewport_y_start

VGA debug readback data register

VGA_MEM_WRITE_PAGE_ADDR - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0x48]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_WRITE_PAGE0_ADDR	9:0	0x0	Write page 0 address
VGA_MEM_WRITE_PAGE1_ADDR	25:16	0x0	Write page 1 address

VGA write page register

VGA_MEM_READ_PAGE_ADDR - RW - 32 bits - [GpuF0MMReg,GpuIOReg:0x4C]			
Field Name	**Bits**	**Default**	**Description**
VGA_MEM_READ_PAGE0_ADDR	9:0	0x0	Read page 0 address
VGA_MEM_READ_PAGE1_ADDR	25:16	0x0	Read page 1 address

VGA read page register

2.7 Display Controller Registers

2.7.1 Primary Display Graphics Control Registers

D1GRPH_ENABLE - RW - 32 bits - [GpuF0MMReg:0x6100]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_ENABLE	0	0x1	Primary graphic enabled. 0=disable 1=enable

Primary graphic enabled.

D1GRPH_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6104]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_DEPTH	1:0	0x0	Primary graphic pixel depth. 0=8bpp 1=16bpp 2=32bpp 3=64bpp
D1GRPH_Z	5:4	0x0	Z[1:0] value for tiling
D1GRPH_FORMAT	10:8	0x0	Primary graphic pixel format. It is used together with D1GRPH_DEPTH to define the graphic pixel format. If (D1GRPH_DEPTH = 0x0)(8 bpp) 0x0 - indexed others - reserved else if (D1GRPH_DEPTH = 0x1)(16 bpp) 0x0 - ARGB 1555 0x1 - RGB 565 0x2 - ARGB 4444 0x3 - Alpha index 88 0x4 - monochrome 16 0x5 - BGRA 5551 others - reserved else if (D1GRPH_DEPTH = 0x2)(32 bpp) 0x0 - ARGB 8888 0x1 - ARGB 2101010 0x2 - 32bpp digital output 0x3 - 8-bit ARGB 2101010 0x4 - BGRA 1010102 0x5 - 8-bit BGRA 1010102 0x6 - RGB 111110 0x7 - BGR 101111 others - reserved else if (D1GRPH_DEPTH = 0x3)(64 bpp) 0x0 - ARGB 16161616 0x1 - 64bpp digital output ARGB[13:2] 0x2 - 64bpp digital output RGB[15:0] 0x3 - 64bpp digital output ARGB[11:0] 0x4 - 64bpp digital output BGR[15:0] others - reserved
D1GRPH_TILE_COMPACT_EN	12	0x0	Enables multichip tile compaction 0=Disable 1=Enable

D1GRPH_ADDRESS_TRANSLATION_ENABLE	16	0x0	Enables display 1 address translation 0=0=physical memory 1=1=virtual memory
D1GRPH_PRIVILEGED_ACCESS_ENABLE	17	0x0	Enables display 1 privileged page access 0=0=no priveledged access 1=1=priveledged access
D1GRPH_ARRAY_MODE	23:20	0x0	Defines the tiling mode 0=ARRAY_LINEAR_GENERAL: Unaligned linear array 1=ARRAY_LINEAR_ALIGNED: Aligned linear array 2=ARRAY_1D_TILED_THIN1: Uses 1D 8x8x1 tiles 3=ARRAY_1D_TILED_THICK: Uses 1D 8x8x4 tiles 4=ARRAY_2D_TILED_THIN1: Uses 8x8x1 macro-tiles 5=ARRAY_2D_TILED_THIN2: Macro-tiles are 2x high 6=ARRAY_2D_TILED_THIN4: Macro-tiles are 4x high 7=ARRAY_2D_TILED_THICK: Uses 8x8x4 macro-tiles 8=ARRAY_2B_TILED_THIN1: uses row bank swapping 9=ARRAY_2B_TILED_THIN2: uses row bank swapping 10=ARRAY_2B_TILED_THIN4: uses row bank swapping 11=ARRAY_2B_TILED_THICK: uses row bank swapping 12=ARRAY_3D_TILED_THIN1: Slices are pipe rotated 13=ARRAY_3D_TILED_THICK: Slices are pipe rotated 14=ARRAY_3B_TILED_THIN1: Slices are pipe rotated 15=ARRAY_3B_TILED_THICK: Slices are pipe rotated
D1GRPH_16BIT_ALPHA_MODE	25:24	0x0	This field is only used if 64 bpp graphics bit depth and graphics/overlay blend using per-pixel alpha from graphics channel. It is used for processing 16 bit alpha. The fixed point graphics alpha value in the frame buffer is always clamped to 0.0 - 1.0 data range. 0x0 - Floating point alpha (1 sign bit, 5 bit exponent, 10 bit mantissa) 0x1 - Fixed point alpha with normalization from 256/256 to 255/255 to represent 1.0 0x2 - Fixed point alpha with no normalization 0x3 - Fixed point alpha using lower 8 bits of frame buffer value, no normalization
D1GRPH_16BIT_FIXED_ALPHA_RANGE	30:28	0x0	This register field is only used if 64 bpp graphics bit depth and D1GRPH_16BIT_ALPHA_MODE = 01 or 10. Also only used if graphics/overlay blend using per-pixel alpha from graphics channel. Final alpha blend value is rounded to 8 bits after optional normalization step (see D1GRPH_16BIT_ALPHA_MODE). 0x0 - Use bits 15:0 of input alpha value for blend alpha 0x1 - Use bits 14:0 of input alpha value for blend alpha 0x2 - Use bits 13:0 of input alpha value for blend alpha 0x3 - Use bits 12:0 of input alpha value for blend alpha 0x4 - Use bits 11:0 of input alpha value for blend alpha 0x5 - Use bits 10:0 of input alpha value for blend alpha 0x6 - Use bits 9:0 of input alpha value for blend alpha 0x7 - Use bits 8:0 of input alpha value for blend alpha

Primary graphic pixel depth and format.

D1GRPH_LUT_SEL - RW - 32 bits - [GpuF0MMReg:0x6108]			
Field Name	**Bits**	**Default**	**Description**

D1GRPH_LUT_SEL	0	0x0	Primary graphic LUT selection. 0=select LUTA 1=select LUTB
D1GRPH_LUT_10BIT_BYPASS_EN	8	0x0	Enable bypass primary graphic LUT for 2101010 format 0=Use LUT 1=Bypass LUT when in 2101010 format. Ignored for other formats
D1GRPH_LUT_10BIT_BYPASS_DBL_BUF_EN	16	0x0	Enable double buffer D1GRPH_LUT_10BIT_BYPASS_EN 0=D1GRPH_LUT_10BIT_BYPASS_EN take effect right away 1=D1GRPH_LUT_10BIT_BYPASS_EN are double buffered

Primary graphic LUT selection.

D1GRPH_SWAP_CNTL - RW - 32 bits - [GpuF0MMReg:0x610C]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_ENDIAN_SWAP	1:0	0x0	MC endian swap select 0=0=none 1=1=8in16(0xaabb=>0xbbaa) 2=2=8in32(0xaabbccdd=>0xddccbbaa) 3=3=8in64(0xaabbccddeeff0011=>0x1100ffeeddccbbaa)
D1GRPH_RED_CROSSBAR	5:4	0x0	Red crossbar select 0=0=select from R 1=1=select from G 2=2=select from B 3=3=select from A
D1GRPH_GREEN_CROSSBAR	7:6	0x0	Green crossbar select 0=0=select from G 1=1=select from B 2=2=select from A 3=3=select from R
D1GRPH_BLUE_CROSSBAR	9:8	0x0	Blue crossbar select 0=0=select from B 1=1=select from A 2=2=select from R 3=3=select from G
D1GRPH_ALPHA_CROSSBAR	11:10	0x0	Alpha crossbar select 0=0=select from A 1=1=select from R 2=2=select from G 3=3=select from B

Endian swap and component reorder control

D1GRPH_PRIMARY_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6110]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_PRIMARY_DFQ_ENABLE	0	0x0	Primary surface address DFQ enable 0=0 = one deep queue mode 1=1 = DFQ mode
D1GRPH_PRIMARY_SURFACE_ADDRESS	31:8	0x0	Primary surface address for primary graphics in byte. It is 256 byte aligned.

Primary surface address for primary graphics in byte.

D1GRPH_SECONDARY_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6118]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_SECONDARY_DFQ_ENABLE *(mirror of D1GRPH_PRIMARY_SURFACE_ADDRESS:D1GRPH_PRIMARY_DFQ_ENABLE)*	0	0x0	Secondary surface address DFQ enable 0=0 = one deep queue mode 1=1 = DFQ mode
D1GRPH_SECONDARY_SURFACE_ADDRESS	31:8	0x0	Secondary surface address for primary graphics in byte. It is 256 byte aligned.

Secondary surface address for primary graphics in byte.

D1GRPH_PITCH - RW - 32 bits - [GpuF0MMReg:0x6120]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_PITCH	13:0	0x0	Primary graphic surface pitch in pixels. For Micro-tiled/Macro-tiled surface, it must be multiple of 64 pixeld in 8bpp mode. For Micro-linear/Macro-tiled surface, it must be multiple of 256 pixeld in 8bpp mode, multiple of 128 pixels in 16bpp mode and multiple of 64 pixels in 32bpp mode. For Micro-linear/Macro-linear surface, it must be multiple of 64 pixels in 8bpp mode. For other modes, it must be multiple of 32. NOTE: Bits 0:4 of this field are hardwired to ZERO.

Primary graphic surface pitch in pixels.

D1GRPH_SURFACE_OFFSET_X - RW - 32 bits - [GpuF0MMReg:0x6124]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_SURFACE_OFFSET_X	12:0	0x0	Primary graphic X surface offset. It is 256 pixels aligned. NOTE: Bits 0:7 of this field are hardwired to ZERO.

Primary graphic X surface offset.

D1GRPH_SURFACE_OFFSET_Y - RW - 32 bits - [GpuF0MMReg:0x6128]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_SURFACE_OFFSET_Y	12:0	0x0	Primary graphic Y surface offset. It must be even value NOTE: Bit 0 of this field is hardwired to ZERO.

Primary graphic Y surface offset.

D1GRPH_X_START - RW - 32 bits - [GpuF0MMReg:0x612C]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_X_START	12:0	0x0	Primary graphic X start coordinate relative to the desktop coordinates.

Primary graphic X start coordinate relative to the desktop coordinates.

D1GRPH_Y_START - RW - 32 bits - [GpuF0MMReg:0x6130]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_Y_START	12:0	0x0	Primary graphic Y start coordinate relative to the desktop coordinates.

Primary graphic Y start coordinate relative to the desktop coordinates.

D1GRPH_X_END - RW - 32 bits - [GpuF0MMReg:0x6134]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_X_END	13:0	0x0	Primary graphic X end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K

Primary graphic X end coordinate relative to the desktop coordinates.

D1GRPH_Y_END - RW - 32 bits - [GpuF0MMReg:0x6138]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_Y_END	13:0	0x0	Primary graphic Y end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K

Primary graphic Y end coordinate relative to the desktop coordinates.

D1GRPH_UPDATE - RW - 32 bits - [GpuF0MMReg:0x6144]			
Field Name	**Bits**	**Default**	**Description**

D1GRPH_MODE_UPDATE_PENDING (R)	0	0x0	Primary graphic mode register update pending control. It is set to 1 after a host write to graphics mode register. It is cleared after double buffering is done. This signal is only visible through register. The graphics surface register includes: D1GRPH_DEPTH D1GRPH_FORMAT D1GRPH_SWAP_RB D1GRPH_LUT_SEL D1GRPH_LUT_10BIT_BYPASS_EN D1GRPH_ENABLE D1GRPH_X_START D1GRPH_Y_START D1GRPH_X_END D1GRPH_Y_END The mode register double buffering can only occur at vertical retrace. The double buffering occurs when D1GRPH_MODE_UPDATE_PENDING = 1 and D1GRPH_UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC1 is disabled, the registers will be updated instantly. 0=No update pending 1=Update pending
D1GRPH_MODE_UPDATE_TAKEN (R)	1	0x0	Primary graphics update taken status for mode registers. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0.
D1GRPH_SURFACE_UPDATE_PENDING (R)	2	0x0	Primary graphic surface register update pending control. If it is set to 1 after a host write to graphics surface register. It is cleared after double buffering is done. It is cleared after double buffering is done. This signal also goes to both the RBBM wait_until and to the CP_RTS_discrete inputs. The graphics surface register includes: D1GRPH_PRIMARY_SURFACE_ADDRESS D1GRPH_SECONDARY_SURFACE_ADDRESS D1GRPH_PITCH D1GRPH_SURFACE_OFFSET_X D1GRPH_SURFACE_OFFSET_Y. If D1GRPH_SURFACE_UPDATE_H_RETRACE_EN = 0, the double buffering occurs in vertical retrace when D1GRPH_SURFACE_UPDATE_PENDING = 1 and D1GRPH_UPDATE_LOCK = 0 and V_UPDATE = 1. Otherwise the double buffering happens at horizontal retrace when D1GRPH_SURFACE_UPDATE_PENDING = 1 and D1GRPH_UPDATE_LOCK = 0 and Data request for last chunk of the line is sent from DCP to DMIF. If CRTC1 is disabled, the registers will be updated instantly
D1GRPH_SURFACE_UPDATE_TAKEN (R)	3	0x0	Primary graphics update taken status for surface registers. If D1GRPH_SURFACE_UPDATE_H_RETRACE_EN = 0, it is set to 1 when double buffering occurs and cleared when V_UPDATE = 0. Otherwise, it is active for one clock cycle when double buffering occurs at the horizontal retrace.
D1GRPH_UPDATE_LOCK	16	0x0	Primary graphic register update lock control. This lock bit control both surface and mode register double buffer 0=Unlocked 1=Locked

D1GRPH_MODE_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D1GRPH mode registers can be updated multiple times in one V_UPDATE period 1=D1GRPH mode registers can only be updated once in one V_UPDATE period
D1GRPH_SURFACE_DISABLE_MULTIPLE_UPDATE	28	0x0	0=D1GRPH surface registers can be updated multiple times in one V_UPDATE period 1=D1GRPH surface registers can only be updated once in one V_UPDATE period

Primary graphic update control

D1GRPH_FLIP_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6148]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_SURFACE_UPDATE_H_RETRACE_EN	0	0x0	Enable primary graphic surface register double buffer in horizontal retrace. 0=Vertical retrace flipping 1=Horizontal retrace flipping

Enable primary graphic surface register double buffer in horizontal retrace

D1GRPH_SURFACE_ADDRESS_INUSE - RW - 32 bits - [GpuF0MMReg:0x614C]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_SURFACE_ADDRESS_INUSE (R)	31:8	0x0	This register reads back snapshot of primary graphics surface address used for data request. The address is the signal sent to DMIF and is updated on SOF or horizontal surface update. The snapshot is triggered by writing 1 into field D1CRTC_SNAPSHOT_MANUAL_TRIGGER of CRTC register D1CRTC_SNAPSHOT_STATUS.

Snapshot of primary graphics surface address in use

2.7.2 Primary Display Video Overlay Control Registers

D1OVL_ENABLE - RW - 32 bits - [GpuF0MMReg:0x6180]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_ENABLE	0	0x0	Primary overlay enabled. 0=disable 1=enable

Primary overlay enabled.

D1OVL_CONTROL1 - RW - 32 bits - [GpuF0MMReg:0x6184]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_DEPTH	1:0	0x0	Primary overlay pixel depth 0=reserved 1=16bpp 2=32bpp 3=reserved
D1OVL_Z	5:4	0x0	Z[1:0] value for tiling
D1OVL_FORMAT	10:8	0x0	Primary overlay pixel format. It is used together with D1OVL_DEPTH to define the overlay format. If (D1OVL_DEPTH = 0x1)(16 bpp) 0x0- ARGB 1555 0x1 - RGB 565 0x2 - BGRA 5551 others - reserved else if (D1OVL_DEPTH = 0x2)(32 bpp) 0x0 - ACrYCb 8888 or ARGB 8888 0x1 - ACrYCb 2101010 or ARGB 2101010 0x2 - CbACrA or BGRA 1010102 others - reserved
D1OVL_TILE_COMPACT_EN	12	0x0	Enables multichip tile compaction 0=Disable 1=Enable
D1OVL_ADDRESS_TRANSLATION_ENABLE	16	0x0	Enables Overlay 1 address translation 0=0: physical memory 1=1: virtual memory
D1OVL_PRIVILEGED_ACCESS_ENABLE	17	0x0	Enables Overlay 1 privileged access 0=0: no privileged access 1=1: privileged access
D1OVL_ARRAY_MODE	23:20	0x0	Defines the tiling mode 0=ARRAY_LINEAR_GENERAL: Unaligned linear array 1=ARRAY_LINEAR_ALIGNED: Aligned linear array 2=ARRAY_1D_TILED_THIN1: Uses 1D 8x8x1 tiles 3=ARRAY_1D_TILED_THICK: Uses 1D 8x8x4 tiles 4=ARRAY_2D_TILED_THIN1: Uses 8x8x1 macro-tiles 5=ARRAY_2D_TILED_THIN2: Macro-tiles are 2x high 6=ARRAY_2D_TILED_THIN4: Macro-tiles are 4x high 7=ARRAY_2D_TILED_THICK: Uses 8x8x4 macro-tiles 8=ARRAY_2B_TILED_THIN1: uses row bank swapping 9=ARRAY_2B_TILED_THIN2: uses row bank swapping 10=ARRAY_2B_TILED_THIN4: uses row bank swapping 11=ARRAY_2B_TILED_THICK: uses row bank swapping 12=ARRAY_3D_TILED_THIN1: Slices are pipe rotated 13=ARRAY_3D_TILED_THICK: Slices are pipe rotated 14=ARRAY_3B_TILED_THIN1: Slices are pipe rotated 15=ARRAY_3B_TILED_THICK: Slices are pipe rotated
D1OVL_COLOR_EXPANSION_MODE	24	0x0	Primary overlay pixel format expansion mode. 0=dynamic expansion for RGB 1=zero expansion for YCbCr

Primary overlay pixel depth and format.

D1OVL_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x6188]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_HALF_RESOLUTION_ENABLE	0	0x0	Primary overlay half resolution control 0=disable 1=enable

Primary overlay half resolution control

D1OVL_SWAP_CNTL - RW - 32 bits - [GpuF0MMReg:0x618C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_ENDIAN_SWAP	1:0	0x0	MC endian swap select 0=0=none 1=1=8in16(0xaabb=>0xbbaa) 2=2=8in32(0xaabbccdd=>0xddccbbaa) 3=3=8in64(0xaabbccddeeff0011=>0x1100ffeeddccbbaa)
D1OVL_RED_CROSSBAR	5:4	0x0	Red crossbar select 0=0=select from R 1=1=select from G 2=2=select from B 3=3=select from A
D1OVL_GREEN_CROSSBAR	7:6	0x0	Green crossbar select 0=0=select from G 1=1=select from B 2=2=select from A 3=3=select from R
D1OVL_BLUE_CROSSBAR	9:8	0x0	Blue crossbar select 0=0=select from B 1=1=select from A 2=2=select from R 3=3=select from G
D1OVL_ALPHA_CROSSBAR	11:10	0x0	Alpha crossbar select 0=0=select from A 1=1=select from R 2=2=select from G 3=3=select from B

Endian swap and component reorder control

D1OVL_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6190]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_DFQ_ENABLE	0	0x0	Surface address DFQ enable
D1OVL_SURFACE_ADDRESS	31:8	0x0	Primary overlay surface base address in byte. It is 256 bytes aligned.

Primary overlay surface base address in byte.

D1OVL_PITCH - RW - 32 bits - [GpuF0MMReg:0x6198]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PITCH	13:0	0x0	Primary overlay surface pitch in pixels. For Micro-tiled/Macro-tiled surface, it must be multiple of 64 pixeld in 8bpp mode. For Micro-linear/Macro-tiled surface, it must be multiple of 256 pixeld in 8bpp mode, multiple of 128 pixels in 16bpp mode and multiple of 64 pixels in 32bpp mode. For Micro-linear/Macro-linear surface, it must be multiple of 64 pixels in 8bpp mode. For other modes, it must be multiple of 32. NOTE: Bits 0:4 of this field are hardwired to ZERO.

Primary overlay surface pitch in pixels.

D1OVL_SURFACE_OFFSET_X - RW - 32 bits - [GpuF0MMReg:0x619C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_SURFACE_OFFSET_X	12:0	0x0	Primary overlay X surface offset. It is 256 pixels aligned. NOTE: Bits 0:7 of this field are hardwired to ZERO.

Primary overlay X surface offset.

D1OVL_SURFACE_OFFSET_Y - RW - 32 bits - [GpuF0MMReg:0x61A0]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_SURFACE_OFFSET_Y	12:0	0x0	Primary overlay Y surface offset. It is even value. NOTE: Bit 0 of this field is hardwired to ZERO.

Primary overlay Y surface offset.

D1OVL_START - RW - 32 bits - [GpuF0MMReg:0x61A4]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_Y_START	12:0	0x0	Primary overlay Y start coordinate relative to the desktop coordinates.
D1OVL_X_START	28:16	0x0	Primary overlay X start coordinate relative to the desktop coordinates.

Primary overlay X, Y start coordinate relative to the desktop coordinates.

D1OVL_END - RW - 32 bits - [GpuF0MMReg:0x61A8]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_Y_END	13:0	0x0	Primary overlay Y end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K.
D1OVL_X_END	29:16	0x0	Primary overlay X end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K.

Primary overlay X, Y end coordinate relative to the desktop coordinates.

D1OVL_UPDATE - RW - 32 bits - [GpuF0MMReg:0x61AC]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_UPDATE_PENDING (R)	0	0x0	Primary overlay register update pending control. It is set to 1 after a host write to overlay double buffer register. It is cleared after double buffering is done. The double buffering occurs when UPDATE_PENDING = 1 and UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC1 is disabled, the registers will be updated instantly. D1OVL double buffer registers include: D1OVL_ENABLE D1OVL_DEPTH D1OVL_FORMAT D1OVL_SWAP_RB D1OVL_COLOR_EXPANSION_MODE D1OVL_HALF_RESOLUTION_ENABLE D1OVL_SURFACE_ADDRESS D1OVL_PITCH D1OVL_SURFACE_OFFSET_X D1OVL_SURFACE_OFFSET_Y D1OVL_START D1OVL_END 0=No update pending 1=Update pending
D1OVL_UPDATE_TAKEN (R)	1	0x0	Primary overlay update taken status. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0.
D1OVL_UPDATE_LOCK	16	0x0	Primary overlay register update lock control. 0=Unlocked 1=Locked
D1OVL_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D1OVL registers can be updated multiple times in one V_UPDATE period 1=D1OVL registers can only be updated once in one V_UPDATE period

Primary overlay register update

D1OVL_SURFACE_ADDRESS_INUSE - RW - 32 bits - [GpuF0MMReg:0x61B0]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_SURFACE_ADDRESS_INUSE (R)	31:8	0x0	This register reads back snapshot of primary overlay surface address used for data request. The address is the signal sent to DMIF and is updated on SOF or horizontal surface update. The snapshot is triggered by writing 1 into field D1CRTC_SNAPSHOT_MANUAL_TRIGGER of CRTC register D1CRTC_SNAPSHOT_STATUS.

Snapshot of primary overlay surface address in use

D1OVL_DFQ_CONTROL - RW - 32 bits - [GpuF0MMReg:0x61B4]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_DFQ_RESET	0	0x0	Reset the deep flip queue
D1OVL_DFQ_SIZE	6:4	0x0	Size of the deep flip queue: 0 = 1 deep queue, 1 = 2 deep queue,..., 7 = 8 deep queue
D1OVL_DFQ_MIN_FREE_ENTRIES	10:8	0x0	Minimum # of free entries before surface pending is asserted

Control of the deep flip queue for D1 overlay

D1OVL_DFQ_STATUS - RW - 32 bits - [GpuF0MMReg:0x61B8]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_DFQ_NUM_ENTRIES (R)	3:0	0x0	# of entries in deep flip queue. 0 = 1 entry, 1 = 2 entries, ... 7 = 8 entries
D1OVL_DFQ_RESET_FLAG (R)	8	0x0	Sticky bit: Deep flip queue in reset
D1OVL_DFQ_RESET_ACK (W)	9	0x0	Clear D1OVL_DFQ_RESET_FLAG

Status of the deep flip queue for D1 overlay

2.7.3 Primary Display Video Overlay Transform Registers

D1OVL_COLOR_MATRIX_TRANSFORMATION_CNTL - RW - 32 bits - [GpuF0MMReg:0x6140]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	2:0	0x0	Matrix transformation control for primary display overlay pixels. It is used when PIX_TYPE is 0. 0=No color space adjustment on display output of overlay pixels 1=Apply display x color spcae control on the overlay pixels based on DxCOLOR_MATRIX_COEF register settings 2=Convert overlay pixel to standard definition YCbCr(601) color space 3=Convert overlay pixels to high definition YCbCR(709) color space 4=Convert overlay pixels to high definition TVRGB color space

Matrix transformation control for primary display overlay pixels.

D1OVL_MATRIX_TRANSFORM_EN - RW - 32 bits - [GpuF0MMReg:0x6200]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_TRANSFORM_EN	0	0x0	Primary overlay matrix conversion enable 0=disable 1=enable

Primary overlay matrix conversion enable.

D1OVL_MATRIX_COEF_1_1 - RW - 32 bits - [GpuF0MMReg:0x6204]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_MATRIX_COEF_1_1	18:0	0x198a0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_1_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_1_2 - RW - 32 bits - [GpuF0MMReg:0x6208]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_1_2	18:0	0x12a20	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_1_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_1_3 - RW - 32 bits - [GpuF0MMReg:0x620C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_1_3	18:0	0x0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_1_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_1_4 - RW - 32 bits - [GpuF0MMReg:0x6210]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_1_4	26:8	0x48700	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S11.1. NOTE: Bits 0:6 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_1_4	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_2_1 - RW - 32 bits - [GpuF0MMReg:0x6214]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_MATRIX_COEF_2_1	18:0	0x72fe0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_2_1	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_2_2 - RW - 32 bits - [GpuF0MMReg:0x6218]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_2_2	18:0	0x12a20	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_2_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_2_3 - RW - 32 bits - [GpuF0MMReg:0x621C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_2_3	18:0	0x79bc0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_2_3	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_2_4 - RW - 32 bits - [GpuF0MMReg:0x6220]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_2_4	26:8	0x22100	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S11.1. NOTE: Bits 0:6 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_2_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_3_1 - RW - 32 bits - [GpuF0MMReg:0x6224]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_MATRIX_COEF_3_1	18:0	0x0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_3_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_3_2 - RW - 32 bits - [GpuF0MMReg:0x6228]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_3_2	18:0	0x12a20	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_3_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_3_3 - RW - 32 bits - [GpuF0MMReg:0x622C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_3_3	18:0	0x20460	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_3_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

D1OVL_MATRIX_COEF_3_4 - RW - 32 bits - [GpuF0MMReg:0x6230]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_MATRIX_COEF_3_4	26:8	0x3af80	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay. Format fix-point S11.1. NOTE: Bits 0:6 of this field are hardwired to ZERO.
D1OVL_MATRIX_SIGN_3_4	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for primary overlay.

2.7.4 Primary Display Video Overlay Gamma Correction Registers

D1OVL_PWL_TRANSFORM_EN - RW - 32 bits - [GpuF0MMReg:0x6280]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_PWL_TRANSFORM_EN	0	0x0	Primary overlay gamma correction enable. 0=disable 1=enable

Primary overlay gamma correction enable.

D1OVL_PWL_0TOF - RW - 32 bits - [GpuF0MMReg:0x6284]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_0TOF_OFFSET	8:0	0x0	Primary overlay gamma correction non-linear offset for input 0x0-0xF. Format fix-point 8.1 (0.0 to +255.5).
D1OVL_PWL_0TOF_SLOPE	26:16	0x100	Primary overlay gamma correction non-linear slope for input 0x0-0xF. Format fix-point 3.8 (0.00 to +7.99).

Primary overlay gamma correction non-linear offset and slope for input 0x0-0xF

D1OVL_PWL_10TO1F - RW - 32 bits - [GpuF0MMReg:0x6288]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_10TO1F_OFFSET	8:0	0x20	Primary overlay gamma correction non-linear offset for input 0x10-0x1F. Format fix-point 8.1 (0.0 to +255.5).
D1OVL_PWL_10TO1F_SLOPE	26:16	0x100	Primary overlay gamma correction non-linear slope for input 0x10-0x1F. Format fix-point 3.8 (0.00 to +7.99).

Primary overlay gamma correction non-linear offset and slope for input 0x10-0x1F

D1OVL_PWL_20TO3F - RW - 32 bits - [GpuF0MMReg:0x628C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_20TO3F_OFFSET	9:0	0x40	Primary overlay gamma correction non-linear offset for input 0x20-0x3F. Format fix-point 9.1 (0.0 to +511.5).
D1OVL_PWL_20TO3F_SLOPE	25:16	0x100	Primary overlay gamma correction non-linear slope for input 0x20-0x3F. Format fix-point 2.8 (0.00 to +3.99).

Primary overlay gamma correction non-linear offset and slope for input 0x20-0x3F

D1OVL_PWL_40TO7F - RW - 32 bits - [GpuF0MMReg:0x6290]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_40TO7F_OFFSET	9:0	0x80	Primary overlay gamma correction non-linear offset for input 40-7F. Format fix-point 9.1 (0.0 to +511.5).
D1OVL_PWL_40TO7F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 40-7F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 40-7F.

D1OVL_PWL_80TOBF - RW - 32 bits - [GpuF0MMReg:0x6294]

Field Name	Bits	Default	Description
D1OVL_PWL_80TOBF_OFFSET	10:0	0x100	Primary overlay gamma correction non-linear offset for input 80-BF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_80TOBF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 80-BF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 80-BF.

D1OVL_PWL_C0TOFF - RW - 32 bits - [GpuF0MMReg:0x6298]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_C0TOFF_OFFSET	10:0	0x180	Primary overlay gamma correction non-linear offset for input C0-FF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_C0TOFF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input C0-FF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input C0-FF.

D1OVL_PWL_100TO13F - RW - 32 bits - [GpuF0MMReg:0x629C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_100TO13F_OFFSET	10:0	0x200	Primary overlay gamma correction non-linear offset for input 100-13F. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_100TO13F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 100-13F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 100-13F.

D1OVL_PWL_140TO17F - RW - 32 bits - [GpuF0MMReg:0x62A0]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_140TO17F_OFFSET	10:0	0x280	Primary overlay gamma correction non-linear offset for input 140-17F. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_140TO17F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 140-17F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 140-17F.

D1OVL_PWL_180TO1BF - RW - 32 bits - [GpuF0MMReg:0x62A4]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_180TO1BF_OFFSET	10:0	0x300	Primary overlay gamma correction non-linear offset for input 180-1BF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_180TO1BF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 180-1BF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 180-1BF.

D1OVL_PWL_1C0TO1FF - RW - 32 bits - [GpuF0MMReg:0x62A8]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_1C0TO1FF_OFFSET	10:0	0x380	Primary overlay gamma correction non-linear offset for input 1C0-1FF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_1C0TO1FF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 1C0-1FF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 1C0-1FF.

D1OVL_PWL_200TO23F - RW - 32 bits - [GpuF0MMReg:0x62AC]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_200TO23F_OFFSET	10:0	0x400	Primary overlay gamma correction non-linear offset for input 200-23F. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_200TO23F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 200-23F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 200-23F.

D1OVL_PWL_240TO27F - RW - 32 bits - [GpuF0MMReg:0x62B0]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_240TO27F_OFFSET	10:0	0x480	Primary overlay gamma correction non-linear offset for input 240-27F. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_240TO27F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 240-27F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 240-27F.

D1OVL_PWL_280TO2BF - RW - 32 bits - [GpuF0MMReg:0x62B4]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_280TO2BF_OFFSET	10:0	0x500	Primary overlay gamma correction non-linear offset for input 280-2BF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_280TO2BF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 280-2BF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 280-2BF.

D1OVL_PWL_2C0TO2FF - RW - 32 bits - [GpuF0MMReg:0x62B8]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_2C0TO2FF_OFFSET	10:0	0x580	Primary overlay gamma correction non-linear offset for input 2C0-2FF. Format fix-point 10.1(0.0 to +1023.5).
D1OVL_PWL_2C0TO2FF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 2C0-2FF. Format fix-point 1.8(0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 2C0-2FF.

D1OVL_PWL_300TO33F - RW - 32 bits - [GpuF0MMReg:0x62BC]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_300TO33F_OFFSET	10:0	0x600	Primary overlay gamma correction non-linear offset for input 300-33F. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_300TO33F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 300-33F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 300-33F.

D1OVL_PWL_340TO37F - RW - 32 bits - [GpuF0MMReg:0x62C0]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_340TO37F_OFFSET	10:0	0x680	Primary overlay gamma correction non-linear offset for input 340-37F. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_340TO37F_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 340-37F. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 340-37F.

D1OVL_PWL_380TO3BF - RW - 32 bits - [GpuF0MMReg:0x62C4]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_380TO3BF_OFFSET	10:0	0x700	Primary overlay gamma correction non-linear offset for input 380-3BF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_380TO3BF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 380-3BF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 380-3BF.

D1OVL_PWL_3C0TO3FF - RW - 32 bits - [GpuF0MMReg:0x62C8]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_PWL_3C0TO3FF_OFFSET	10:0	0x780	Primary overlay gamma correction non-linear offset for input 3C0-3FF. Format fix-point 10.1 (0.0 to +1023.5).
D1OVL_PWL_3C0TO3FF_SLOPE	24:16	0x100	Primary overlay gamma correction non-linear slope for input 3C0-3FF. Format fix-point 1.8 (0.00 to +1.99).

Primary overlay gamma correction non-linear offset and slope for input 3C0-3FF.

2.7.5 Primary Display Graphics and Overlay Blending Registers

D1OVL_KEY_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6300]			
Field Name	**Bits**	**Default**	**Description**

D1GRPH_KEY_FUNCTION	1:0	0x0	Selects graphic keyer result equation for primary display. 0=GRPH1_KEY = FALSE = 0 1=GRPH1_KEY = TRUE = 1 2=GPPH1_KEY = (GRPH1_RED in range) AND (GRPH1_GREEN in range) AND (GRPH1_BLUE in range) AND(GRPH1_ALPHA in range) 3=GRPH1_KEY = not [(GRPH1_RED in range) AND (GRPH1_GREEN in range) AND (GRPH1_BLUE in range) AND(GRPH1_ALPHA in range)]
D1OVL_KEY_FUNCTION	9:8	0x0	Selects overlay keyer result equation for primary display. 0=OVL1_KEY = FALSE = 0 1=OVL1_KEY = TRUE = 1 2=OVL1_KEY = (OVL1_Cr_RED in range) AND (OVL1_Y_GREEN in range) AND (OVL1_Cb_BLUE in range) AND (OVL1_ALPHA in range) 3=OVL1_KEY = not [(OVL1_Cr_RED in range) AND (OVL1_Y_GREEN in range) AND (OVL1_Cb_BLUE in range) AND (OVL1_ALPHA in range)]
D1OVL_KEY_COMPARE_MIX	16	0x0	Selects final mix of graphics and overlay keys for primary display. 0=GRPH_OVL_KEY = GRPH_KEY or OVL_KEY 1=GRPH_OVL_KEY = GRPH_KEY and OVL_KEY

Primary display key control

D1GRPH_ALPHA - RW - 32 bits - [GpuF0MMReg:0x6304]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_ALPHA	7:0	0xff	Global graphic alpha for use in key mode and global alpha modes. See D1OVL_ALPHA_MODE register filed for more details

Global graphic alpha for use in key mode and global alpha modes.

D1OVL_ALPHA - RW - 32 bits - [GpuF0MMReg:0x6308]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_ALPHA	7:0	0xff	Global overlay alpha for use in key mode and global alpha modes. See D1OVL_ALPHA_MODE register filed for more details

Global overlay alpha for use in key mode and global alpha modes.

D1OVL_ALPHA_CONTROL - RW - 32 bits - [GpuF0MMReg:0x630C]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_ALPHA_MODE	1:0	0x0	Graphics/overlay alpha blending mode for primary controller. In any case, if there is only graphics, the input OVL_DATA is forced to blank. If there is only overlay, the input GRPH_DATA is forced to blank. 0=Keyer mode, select graphic or overlay keyer to mix graphics and overlay 1=Per pixel graphic alpha mode.Alpha blend graphic and overlay layer. The alpha from graphic pixel may be inverted according to register field 2=Global alpha mode 3=Per pixel overlay alpha mode
D1OVL_ALPHA_PREMULT	8	0x0	For use with per pixel alpha blend mode. Selects whether pre-multiplied alpha or non-multiplied alpha. 0=0x0 - When DxOVL_ALPHA_MODE = 0x1, then Pixel = PIX_ALPHA * graphics pixel + (1-PIX_ALPHA) * overlay pixel.When DxOVL_ALPHA_MODE = 0x3, then Pixel = PIX_ALPHA * overlay pixel + (1-PIX_ALPHA) * graphic pixel 1=0x1 - When DxOVL_ALPHA_MODE = 0x1, then Pixel = graphic pixel + (1-PIX_ALPHA) * overlay pixel.When DxOVL_ALPHA_MODE = 0x3, then Pixel = overlay pixel + (1-PIX_ALPHA) * graphic pixel
D1OVL_ALPHA_INV	16	0x0	For use with pixel blend mode. Apply optional inversion to the alpha value extracted form the graphics or overlay surface data. 0=PIX_ALPHA = alpha from graphics or overlay 1=PIX_ALPHA = 1 - alpha from graphics or overlay

Primary display graphics/overlay alpha blending control

D1GRPH_KEY_RANGE_RED - RW - 32 bits - [GpuF0MMReg:0x6310]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_KEY_RED_LOW	15:0	0x0	Primary graphics keyer red component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D1GRPH_KEY_RED_HIGH	31:16	0x0	Primary graphics keyer red component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Primary graphics keyer red component range

D1GRPH_KEY_RANGE_GREEN - RW - 32 bits - [GpuF0MMReg:0x6314]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_KEY_GREEN_LOW	15:0	0x0	Primary graphics keyer green component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D1GRPH_KEY_GREEN_HIGH	31:16	0x0	Primary graphics keyer green component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Primary graphics keyer green component range

D1GRPH_KEY_RANGE_BLUE - RW - 32 bits - [GpuF0MMReg:0x6318]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_KEY_BLUE_LOW	15:0	0x0	Primary graphics keyer blue component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D1GRPH_KEY_BLUE_HIGH	31:16	0x0	Primary graphics keyer blue component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Primary graphics keyer blue component range

D1GRPH_KEY_RANGE_ALPHA - RW - 32 bits - [GpuF0MMReg:0x631C]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_KEY_ALPHA_LOW	15:0	0x0	Primary graphics keyer alpha component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D1GRPH_KEY_ALPHA_HIGH	31:16	0x0	Primary graphics keyer alpha component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Primary graphics keyer alpha component range

D1OVL_KEY_RANGE_RED_CR - RW - 32 bits - [GpuF0MMReg:0x6320]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_KEY_RED_CR_LOW	9:0	0x0	Primary overlay keyer red component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D1OVL_KEY_RED_CR_HIGH	25:16	0x0	Primary overlay keyer red component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Primary overlay keyer red component range

D1OVL_KEY_RANGE_GREEN_Y - RW - 32 bits - [GpuF0MMReg:0x6324]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_KEY_GREEN_Y_LOW	9:0	0x0	Primary overlay keyer green component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D1OVL_KEY_GREEN_Y_HIGH	25:16	0x0	Primary overlay keyer green component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Primary overlay keyer green component range

D1OVL_KEY_RANGE_BLUE_CB - RW - 32 bits - [GpuF0MMReg:0x6328]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_KEY_BLUE_CB_LOW	9:0	0x0	Primary overlay keyer blue component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D1OVL_KEY_BLUE_CB_HIGH	25:16	0x0	Primary overlay keyer blue component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Primary overlay keyer blue component range

D1OVL_KEY_ALPHA - RW - 32 bits - [GpuF0MMReg:0x632C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_KEY_ALPHA_LOW	7:0	0x0	Primary overlay keyer alpha component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D1OVL_KEY_ALPHA_HIGH	23:16	0x0	Primary overlay keyer alpha component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Primary overlay keyer alpha component range

2.7.6 Primary Display Color Matrix Transform Registers

D1GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL - RW - 32 bits - [GpuF0MMReg:0x6380]			
Field Name	**Bits**	**Default**	**Description**
D1GRPH_COLOR_MATRIX_TRANSFORMATION_EN	0	0x0	Matrix transformation control for primary display graphics and cursor pixel. It is used when PIX_TYPE is 1. 0=disable 1=enable

Matrix transformation control for primary display graphics and cursor pixel.

D1COLOR_MATRIX_COEF_1_1 - RW - 32 bits - [GpuF0MMReg:0x6384]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_1_1	16:0	0x0	Combined matrix constant C11 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S1.11(-2.00 to +1.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_1_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C11 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_1_2 - RW - 32 bits - [GpuF0MMReg:0x6388]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_1_2	15:0	0x0	Combined matrix constant C12 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S0.11(-1.00 to + 0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_1_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C12 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_1_3 - RW - 32 bits - [GpuF0MMReg:0x638C]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_1_3	15:0	0x0	Combined matrix constant C13 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S0.11(-1.0 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_1_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C13 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_1_4 - RW - 32 bits - [GpuF0MMReg:0x6390]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_1_4	26:8	0x0	Combined matrix constant C14 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S11.1(-2048.5 to +2047.5). It includes subtraction of 512 offset NOTE: Bits 0:6 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_1_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C14 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_2_1 - RW - 32 bits - [GpuF0MMReg:0x6394]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_2_1	15:0	0x0	Combined matrix constant C21 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_2_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C21 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_2_2 - RW - 32 bits - [GpuF0MMReg:0x6398]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_2_2	16:0	0x0	Combined matrix constant C22 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S1.11(-2.00 to +1.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_2_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C22 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_2_3 - RW - 32 bits - [GpuF0MMReg:0x639C]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_2_3	15:0	0x0	Combined matrix constant C23 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_2_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C23 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_2_4 - RW - 32 bits - [GpuF0MMReg:0x63A0]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_2_4	26:8	0x0	Combined matrix constant C24 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S11.1(-2048.5 to +2047.5). It includes subtraction of 512 offset NOTE: Bits 0:6 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_2_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C24 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_3_1 - RW - 32 bits - [GpuF0MMReg:0x63A4]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_3_1	15:0	0x0	Combined matrix constant C31 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_3_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C31 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_3_2 - RW - 32 bits - [GpuF0MMReg:0x63A8]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_3_2	15:0	0x0	Combined matrix constant C32 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_3_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C32 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_3_3 - RW - 32 bits - [GpuF0MMReg:0x63AC]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_3_3	16:0	0x0	Combined matrix constant C33 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S1.11(-2.00 to +1.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_3_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C33 of RGB->YCbCr, contrast and brightness adjustment for primary display.

D1COLOR_MATRIX_COEF_3_4 - RW - 32 bits - [GpuF0MMReg:0x63B0]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_MATRIX_COEF_3_4	26:8	0x0	Combined matrix constant C34 of RGB->YCbCr, contrast and brightness adjustment for primary display. Format fix-point S11.1(-2048.5 to +2047.5). It includes subtraction of 512 offset NOTE: Bits 0:6 of this field are hardwired to ZERO.
D1COLOR_MATRIX_SIGN_3_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C34 of RGB->YCbCr, contrast and brightness adjustment for primary display.

2.7.7 Primary Display Subsampling Registers

D1COLOR_SPACE_CONVERT - RW - 32 bits - [GpuF0MMReg:0x613C]			
Field Name	**Bits**	**Default**	**Description**
D1COLOR_SUBSAMPLE_CRCB_MODE	1:0	0x0	Sub-sampling control for primary display 0=do not subsample CrCb(RB) 1=subsample CrCb (RB) by using 2 tap average method 2=subsample CrCb (RB) by using 1 tap on even pixel 3=subsample CrCb (RB) by using 1 tap on odd pixel

Sub-sampling control for primary display.

2.7.8 Primary Display Realtime Overlay Registers

D1OVL_RT_SKEWCOMMAND - RW - 32 bits - [GpuF0MMReg:0x6500]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_RT_CLEAR_GOBBLE_COUNT (W)	0	0x0	writing 1 to this bit clear the gobbleCount this bit has higher priority than inc_gobblecount
D1OVL_RT_INC_GOBBLE_COUNT (W)	4	0x0	writing 1 to this bit increments the gobbleCount
D1OVL_RT_CLEAR_SUBMIT_COUNT (W)	8	0x0	writing 1 to this bit clear the submitCount this bit has higher priority than inc_submitcount
D1OVL_RT_INC_SUBMIT_COUNT (W)	12	0x0	writing 1 to this bit increments the submitCount
D1OVL_RT_GOBBLE_COUNT (R)	18:16	0x0	read only register gobble count value which increments with each inc_gobble_count and reset with clear_gobble_count commands. it wraps around on overflow during increment.
D1OVL_RT_SUBMIT_COUNT (R)	26:24	0x0	read only register submit count value which increments with each inc_submit_count and reset with clear_submit_count commands. it wraps around on overflow during increment.

reset or increment submit and gobble count

D1OVL_RT_SKEWCONTROL - RW - 32 bits - [GpuF0MMReg:0x6504]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_RT_CAPS	2:0	0x0	max value in submitCount and gobbleCount this is the number of contents buffer - 1 should reset counters before programming this field
D1OVL_RT_SKEW_MAX	6:4	0x0	max skew allowed between gobbleCount and submitCount

controls for submit and gobble counts

D1OVL_RT_BAND_POSITION - RW - 32 bits - [GpuF0MMReg:0x6508]			
Field Name	**Bits**	**Default**	**Description**

D1OVL_RT_TOP_SCAN	13:0	0x0	define the top scan line for the next RT (inclusive)
D1OVL_RT_BTM_SCAN	29:16	0x0	define the bottom scan line for next RT (exclusive)

the position of the top and bottom scan line for next RT

D1OVL_RT_PROCEED_COND - RW - 32 bits - [GpuF0MMReg:0x650C]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_RT_REDUCE_DELAY	0	0x0	0 selects delay optimized scheme 1 selects basic render behind delay scan scheme
D1OVL_RT_RT_FLIP	4	0x0	0 selects bandSync to be exposed to CP 1 selects frameSync to be exposed to CP
D1OVL_RT_PROCEED_ON_EOF_DISABLE	8	0x0	0 enables unfinished bands to pass bandSync on EOF (valid only in basic scheme) 1 disables this feature
D1OVL_RT_WITH_HELD_ON_SOF	12	0x0	0 disables proceedOnEOF on next frameSync 1 disables proceedOnEOF on next SOF
D1OVL_RT_CLEAR_GOBBLE_GO (W)	14	0x0	This bit clear gobbleGo disable another frame submit before next flip (ignored in basic scheme)
D1OVL_RT_TEAR_PROOF_HEIGHT	29:16	0x0	define the number of scan lines above topscan. if display starts reading from there, RT should wait

select RT flip proceed condition

D1OVL_RT_STAT - RW - 32 bits - [GpuF0MMReg:0x6510]			
Field Name	**Bits**	**Default**	**Description**
D1OVL_RT_FIP_PROCEED_ACK (W)	0	0x0	The sticky bit clears the FIP_PROCEED FLAG flag when written
D1OVL_RT_FRAME_SYNC_ACK (W)	1	0x0	The sticky bit clears the RT_FRAME_SYNC flag when written
D1OVL_RT_OVL_START_ACK (W)	2	0x0	The sticky bit clears the OVL_START FLAG flag when written
D1OVL_RT_BAND_INVISIBLE (R)	8	0x0	Debug bit indicating that overlay scanning in invisble region
D1OVL_RT_BAND_SYNC (R)	9	0x0	Debug bit indicating that overlay bottom scan is less the line counter
D1OVL_RT_EOF_PRPCEED (R)	10	0x0	Debug bit indicating that overlay is ended. Set at eof and reset at overlay start
D1OVL_RT_FIP_PROCEED (R)	11	0x0	Sticky debug bit that set when RT_FLIP_PROCEED signal asserted.
D1OVL_RT_FRAME_SYNC (R)	12	0x0	Sticky debug bit indicating that overlay start set and a new submission occured
D1OVL_RT_GOBBLE_GO (R)	13	0x0	Debug bit that set on frame_sync and clear at gobbleclr
D1OVL_RT_NEW_SUBMIT (R)	14	0x0	Debug bit indicating a new submission occurred
D1OVL_RT_OVL_START (R)	15	0x0	Debug bit indicating that line buffer detects start of overlay being accessed
D1OVL_RT_OVL_ENDED (R)	16	0x0	Debug bit indicating that line buffer detects that the end of overlay being accessed
D1OVL_RT_SAFE_ZONE (R)	17	0x0	Debug bit indicating that overlay is scaning in safe zone

D1OVL_RT_SWITCH_REGIONS (R)	18	0x0	Debug bit showing the postion of scan region relative to display
D1OVL_SKEW_MAX_REACHED (R)	19	0x0	Debug bit indicating that line buffer detected maximum skew reached
D1OVL_LINE_COUNTER (R)	31:20	0x0	debug bit showing display line counter value

Status Bits

2.7.9 Primary Display Hardware Cursor Registers

D1CUR_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6400]			
Field Name	**Bits**	**Default**	**Description**
D1CURSOR_EN	0	0x0	Primary display hardware cursor enabled. 0=disable 1=enable
D1CURSOR_MODE	9:8	0x0	Primary display hardware cursor mode. For 2bpp mode, each line of cursor data is stored in memory as 16 bits of AND data followed by 16 bits XOR data. For color AND/XOR mode, each pixel is stored sequentially in memory as 32bits each in aRGB8888 format with bit 31 of each DWord being the AND bit. For the color alpha modes the format is also 32bpp aRGB8888 with all 8 bits of the alpha being used.All HW cursor lines must be 64 pixels wide and all lines must be stored sequentially in memory. 0=Mono (2bpp) 1=Color 24bpp + 1 bit AND (32bpp) 2=Color 24bpp + 8 bit alpha (32bpp) premultiplied alpha 3=Color 24bpp + 8 bit alpha (32bpp)unmultiplied alpha
D1CURSOR_2X_MAGNIFY	16	0x0	Primary display hardware cursor 2x2 magnification. 0=no 2x2 magnification 1=2x2 magnification in horizontal and vertical direction
D1CURSOR_FORCE_MC_ON	20	0x0	When set, if the incoming data is in D1 cursor region, DCP_LB_cursor1_allow_stutter is set. This field in this double bufferred register is not double buffered

Primary display hardware control

D1CUR_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6408]			
Field Name	**Bits**	**Default**	**Description**
D1CURSOR_SURFACE_ADDRESS	31:0	0x0	Primary display hardware cursor surface base address in byte. It is 4K byte aligned. NOTE: Bits 0:11 of this field are hardwired to ZERO.

Primary display hardware cursor surface base address.

D1CUR_SIZE - RW - 32 bits - [GpuF0MMReg:0x6410]			
Field Name	**Bits**	**Default**	**Description**
D1CURSOR_HEIGHT	5:0	0x0	Primary display hardware cursor height minus 1.
D1CURSOR_WIDTH	21:16	0x0	Primary display hardware cursor width minus 1.

Primary display hardware size

D1CUR_POSITION - RW - 32 bits - [GpuF0MMReg:0x6414]			
Field Name	**Bits**	**Default**	**Description**
D1CURSOR_Y_POSITION	12:0	0x0	Primary display hardware cursor X coordinate at the hot spot relative to the desktop coordinates.
D1CURSOR_X_POSITION	28:16	0x0	Primary display hardware cursor X coordinate at the hot spot relative to the desktop coordinates.

Primary display hardware cursor position

D1CUR_HOT_SPOT - RW - 32 bits - [GpuF0MMReg:0x6418]			
Field Name	**Bits**	**Default**	**Description**
D1CURSOR_HOT_SPOT_Y	5:0	0x0	Primary display hardware cursor hot spot X length relative to the top left corner.
D1CURSOR_HOT_SPOT_X	21:16	0x0	Primary display hardware cursor hot spot Y length relative to the top left corner.

Primary display hardware cursor hot spot position

D1CUR_COLOR1 - RW - 32 bits - [GpuF0MMReg:0x641C]			
Field Name	**Bits**	**Default**	**Description**
D1CUR_COLOR1_BLUE	7:0	0x0	Primary display hardware cursor blue component of color 1.
D1CUR_COLOR1_GREEN	15:8	0x0	Primary display hardware cursor green component of color 1.
D1CUR_COLOR1_RED	23:16	0x0	Primary display hardware cursor red component of color 1.

Primary display hardware cursor color 1.

D1CUR_COLOR2 - RW - 32 bits - [GpuF0MMReg:0x6420]			
Field Name	**Bits**	**Default**	**Description**
D1CUR_COLOR2_BLUE	7:0	0x0	Primary display hardware cursor blue component of color 2.
D1CUR_COLOR2_GREEN	15:8	0x0	Primary display hardware cursor green component of color 2.
D1CUR_COLOR2_RED	23:16	0x0	Primary display hardware cursor red component of color 2.

Primary display hardware cursor color 2.

D1CUR_UPDATE - RW - 32 bits - [GpuF0MMReg:0x6424]			
Field Name	**Bits**	**Default**	**Description**

D1CURSOR_UPDATE_PENDING (R)	0	0x0	Primary display hardware cursor update pending status. It is set to 1 after a host write to cursor double buffer register. It is cleared after double buffering is done. The double buffering occurs when D1CURSOR_UPDATE_PENDING = 1 and D1CURSOR_UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC1 is disabled, the registers will be updated instantly. The D1CUR double buffer registers are: D1CURSOR_EN D1CURSOR_MODE D1CURSOR_2X_MAGNIFY D1CURSOR_SURFACE_ADDRESS D1CURSOR_HEIGHT D1CURSOR_WIDTH D1CURSOR_X_POSITION D1CURSOR_Y_POSITION D1CURSOR_HOT_SPOT_X D1CURSOR_HOT_SPOT_Y 0=No update pending 1=Update pending
D1CURSOR_UPDATE_TAKEN (R)	1	0x0	Primary display hardware cursor update taken status. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0
D1CURSOR_UPDATE_LOCK	16	0x0	Primary display hardware cursor update lock control. 0=Unlocked 1=Locked
D1CURSOR_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D1CURSOR registers can be updated multiple times in one V_UPDATE period 1=D1CURSOR registers can only be updated once in one V_UPDATE period

2.7.10 Primary Display Hardware Icon Registers

D1ICON_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6440]			
Field Name	**Bits**	**Default**	**Description**
D1ICON_ENABLE	0	0x0	Primary display hardware icon enable. 0=disable 1=enable
D1ICON_2X_MAGNIFY	16	0x0	Primary display hardware icon 2x2 magnification. 0=no 2x2 magnification 1=2x2 magnification in horizontal and vertical direction
D1ICON_FORCE_MC_ON	20	0x0	When set, if the incoming data is in D1 icon region, DCP_LB_icon1_allow_stutter is set. This field in this double bufferred register is not double buffered.

Primary display hardware icon control.

D1ICON_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6448]			
Field Name	**Bits**	**Default**	**Description**

D1ICON_SURFACE_ADDRESS	31:0	0x0	Primary display hardware icon surface base address in byte. It is 4K byte aligned. NOTE: Bits 0:11 of this field are hardwired to ZERO.

Primary display hardware icon surface base address.

D1ICON_SIZE - RW - 32 bits - [GpuF0MMReg:0x6450]			
Field Name	**Bits**	**Default**	**Description**
D1ICON_HEIGHT	6:0	0x0	Primary display hardware icon height minus 1.
D1ICON_WIDTH	22:16	0x0	Primary display hardware icon width minus 1.

Primary display hardware icon size.

D1ICON_START_POSITION - RW - 32 bits - [GpuF0MMReg:0x6454]			
Field Name	**Bits**	**Default**	**Description**
D1ICON_Y_POSITION	12:0	0x0	Primary display hardware icon Y start coordinate related to the desktop coordinates. Note: Icon can not be off the top and off the left edge of the display surface. But can be off the bottom and off the right edge of the display.
D1ICON_X_POSITION	28:16	0x0	Primary display hardware icon X start coordinate relative to the desktop coordinates. Note: Icon can not be off the top and off the left edge of the display surface. But can be off the bottom and off the right edge of the display.

Primary display hardware icon position

D1ICON_COLOR1 - RW - 32 bits - [GpuF0MMReg:0x6458]			
Field Name	**Bits**	**Default**	**Description**
D1ICON_COLOR1_BLUE	7:0	0x0	Primary display hardware icon blue component of color 1.
D1ICON_COLOR1_GREEN	15:8	0x0	Primary display hardware icon green component of color 1.
D1ICON_COLOR1_RED	23:16	0x0	Primary display hardware icon red component of color 1.

Primary display hardware icon color 1.

D1ICON_COLOR2 - RW - 32 bits - [GpuF0MMReg:0x645C]			
Field Name	**Bits**	**Default**	**Description**
D1ICON_COLOR2_BLUE	7:0	0x0	Primary display hardware icon blue component of color 2.
D1ICON_COLOR2_GREEN	15:8	0x0	Primary display hardware icon green component of color 2.
D1ICON_COLOR2_RED	23:16	0x0	Primary display hardware icon red component of color 2.

Primary display hardware icon color 2.

D1ICON_UPDATE - RW - 32 bits - [GpuF0MMReg:0x6460]			
Field Name	**Bits**	**Default**	**Description**
D1ICON_UPDATE_PENDING (R)	0	0x0	Primary display hardware icon update Pending status. It is set to 1 after a host write to icon double buffer register. It is cleared after double buffering is done. The double buffering occurs when D1ICON_UPDATE_PENDING = 1 and D1ICON_UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC1 is disabled, the registers will be updated instantly. D1IOCN double buffer registers include : D1ICON_ENABLE D1ICON_2X_MAGNIFY D1ICON_SURFACE_ADDRESS D1ICON_HEIGHT D1ICON_WIDTH D1ICON_Y_POSITION D1ICON_X_POSITION 0=No update pending 1=Update pending
D1ICON_UPDATE_TAKEN (R)	1	0x0	Primary display hardware icon update Taken status. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0
D1ICON_UPDATE_LOCK	16	0x0	Primary display hardware icon update lock control. 0=Unlocked 1=Locked
D1ICON_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D1ICON registers can be updated multiple times in one V_UPDATE period 1=D1ICON registers can only be updated once in one V_UPDATE period

Primary display hardware icon update control

2.7.11 Primary Display Multi-VPU Control Registers

D1CRTC_MVP_CONTROL1 - RW - 32 bits - [GpuF0MMReg:0x6038]			
Field Name	**Bits**	**Default**	**Description**
MVP_EN	0	0x0	Enable MVP feature
MVP_MIXER_MODE	6:4	0x0	000 - Split mode/super-tile mode; 001 - AFR manual (driver control); 010 - AFR (switching); 011 - AFR manual switch (set inband control character through register); 100 - SuperAA with gamma and degamma enabled; 101 - SuperAA with only gamma enabled
MVP_MIXER_SLAVE_SEL	8	0x0	0 - in AFR manual (drive control) mode, use master inputs in the next frame; '1' - use the slave input
MVP_MIXER_SLAVE_SEL_DELAY_UNTIL_END_OF_BLANK	9	0x0	0 - MVP_MIXER_SLAVE_SEL takes effect immediately; '1' - MVP_MIXER_SLAVE_SEL does not take effect until end of horizontal or vertical blank region
MVP_ARBITRATION_MODE_FOR_AFR_MANUAL_SWITCH_MODE	10	0x0	Arbitration scheme used when both master and slave GPU switch AFR flip queue status 0 = pixel source comes from the GPU which last make the switch 1 = pixel source changes to the GPU which is not currently displayed
MVP_RATE_CONTROL	12	0x0	0 - DDR; 1 - SDR

MVP_CHANNEL_CONTROL	16	0x0	0 - single channel; 1 - dual channel
MVP_GPU_CHAIN_LOCATION	21:20	0x0	The location of the GPU in a chain: 00 - Master GPU, 01 - middle GPU, 10 - head slave GPU (or slave GPU in dual-GPU system
MVP_DISABLE_MSB_EXPAND	24	0x0	How to expand each color component of pixel data from slave GPU from 8 to 10 bits: 0 - dynamic expansion, 1 - pad 0s
MVP_30BPP_EN	28	0x0	Enable 30bpp operation
MVP_TERMINATION_CNTL_A	30	0x0	Controls DVP termination resistors
MVP_TERMINATION_CNTL_B	31	0x0	Controls DVP termination resistors

MVP Control 1

D1CRTC_MVP_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x603C]			
Field Name	**Bits**	**Default**	**Description**
MVP_MUX_DE_DVOCNTL0_SEL	0	0x0	0 - selects DVOCNT2; 1 - selects DVOCNT0
MVP_MUX_DE_DVOCNTL2_SEL	4	0x0	0 - selects DVOCNT2; 1 - selects DVOCNT0
MVP_MUXA_CLK_SEL	8	0x0	0 - selects CLKA; 1 - selects CLKB
MVP_MUXB_CLK_SEL	12	0x0	0 - selects CLKA; 1 - selects CLKB
MVP_DVOCNTL_MUX	16	0x0	0 - DVOCNTL[2:0] = DVO_DE, DVO_HSYNC, DVO_VSYNC; 1 - DVOCNTL[2:0] = DVO_DE, MVP_DVOCLK_C, DVO_DE
MVP_FLOW_CONTROL_OUT_EN	20	0x0	Enable flow_control_out
MVP_SWAP_LOCK_OUT_EN	24	0x0	0 - Swap_lock_out is not enabled; 1 - enable swap_lock out in GPIO
MVP_SWAP_AB_IN_DC_DDR	28	0x1	1 - swap in A & B data in dual channel DDR mode. This is the default

MVP Control 2

D1CRTC_MVP_FIFO_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6040]			
Field Name	**Bits**	**Default**	**Description**
MVP_STOP_SLAVE_WM	7:0	0x8	At the period after the start of DE from slave GPU, if MVP FIFO level exceeds this watermark, flow control is asserted
MVP_PAUSE_SLAVE_WM	15:8	0x8	In the middle of receiving a raster line from the slave GPU, if MVP FIFO level falls below this watermark, flow control signal is asserted for MVP_PAUSE_SLAVE_CNT cycles
MVP_PAUSE_SLAVE_CNT	23:16	0x4	In the middle of receiving a raster line from the slave GPU, if MVP FIFO level falls below this watermark, flow control signal is asserted for MVP_PAUSE_SLAVE_CNT cycles

MVP FIFO Control

D1CRTC_MVP_FIFO_STATUS - RW - 32 bits - [GpuF0MMReg:0x6044]			
Field Name	**Bits**	**Default**	**Description**
MVP_FIFO_LEVEL (R)	7:0	0x0	MVP FIFO level, in # of pixels
MVP_FIFO_OVERFLOW (R)	8	0x0	MVP FIFO overflows
MVP_FIFO_OVERFLOW_OCCURRED (R)	12	0x0	Sticky bit - MVP FIFO overflow has occurred
MVP_FIFO_OVERFLOW_ACK	16	0x0	Resets MVP_FIFO_OVERFLOW_OCCURRED
MVP_FIFO_UNDERFLOW (R)	20	0x0	MVP FIFO underflows

MVP_FIFO_UNDERFLOW_OCCURRED (R)	24	0x0	Sticky bit - MVP FIFO underflows occurred
MVP_FIFO_UNDERFLOW_ACK	28	0x0	Resets MVP_FIFO_UNDERFLOW_OCCURRED
MVP_FIFO_ERROR_MASK	30	0x0	Set to 1 to enable interrupt on mvp fifo overflow or underflow event
MVP_FIFO_ERROR_INT_STATUS (R)	31	0x0	Fifo error status flag (masked OR of fifo over/uderflow)

MVP FIFO Status

D1CRTC_MVP_SLAVE_STATUS - RW - 32 bits - [GpuF0MMReg:0x6048]			
Field Name	**Bits**	**Default**	**Description**
MVP_SLAVE_PIXELS_PER_LINE_RCVED (R)	12:0	0x0	The number of active pixels per line received from the slave GPU
MVP_SLAVE_LINES_PER_FRAME_RCVED (R)	28:16	0x0	The number of active lines per frame received from the slave GPU

MVP Slave Status

D1CRTC_MVP_INBAND_CNTL_CAP - RW - 32 bits - [GpuF0MMReg:0x604C]			
Field Name	**Bits**	**Default**	**Description**
MVP_IGNOR_INBAND_CNTL	0	0x1	Master GPU ignors the inband control signal
MVP_PASSING_INBAND_CNTL_EN	4	0x0	Slave GPU passes upstream slave GPU to downstream slave GPU/master GPU
MVP_INBAND_CNTL_CHAR_CAP (R)	31:8	0x0	Inband control signal received from slave GPU

MVP Capture Inband Control

D1CRTC_MVP_INBAND_CNTL_INSERT - RW - 32 bits - [GpuF0MMReg:0x6050]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_MVP_INBAND_OUT_MODE	1:0	0x0	00 - disable inband insertion; 01 - used for debug only: insert register MVP_INBAND_CNTL_CHAR_INSERT; 10 - normal mode: insert the character generated by MVP_mixer
D1CRTC_MVP_INBAND_CNTL_CHAR_INSERT	31:8	0x0	Used for debug only: 24-bit control character for insertion

MVP Insert Inband Control

D1CRTC_MVP_INBAND_CNTL_INSERT_TIMER - RW - 32 bits - [GpuF0MMReg:0x6054]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_MVP_INBAND_CNTL_CHAR_INSERT_TIMER	7:0	0x8	The number of clock cycles the character insertion trigger from the line buffer needs to be ahead of end of lines for CRTC to insert the in-band control character

MVP Insert Inband Control Timer

D1CRTC_MVP_BLACK_KEYER - RW - 32 bits - [GpuF0MMReg:0x6058]			
Field Name	**Bits**	**Default**	**Description**
MVP_BLACK_KEYER_R	9:0	0x0	Black keyer value, for red pixel
MVP_BLACK_KEYER_G	19:10	0x0	Black keyer value, for green pixel
MVP_BLACK_KEYER_B	29:20	0x0	Black keyer value, for blue pixel

MVP Black keyer for smoothing out pixels after black keyer in LB in SFR mode

D1CRTC_MVP_STATUS - RW - 32 bits - [GpuF0MMReg:0x605C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_FLIP_NOW_OCCURRED (R)	0	0x0	Reports whether flip_now has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D1CRTC_AFR_HSYNC_SWITCH_DONE_OCCURRED (R)	4	0x0	Reports whether afr_hsync_switch_done has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D1CRTC_FLIP_NOW_CLEAR (W)	16	0x0	Clears the sticky bit D1CRTC_FLIP_NOW_OCCURRED when written with '1'
D1CRTC_AFR_HSYNC_SWITCH_DONE_CLEAR (W)	20	0x0	Clears the sticky bit D1CRTC_AFR_HSYNC_SWITCH_DONE_OCCURRED when written with '1'

Reports status for MVP flipping in CRTC1

D2CRTC_MVP_INBAND_CNTL_INSERT - RW - 32 bits - [GpuF0MMReg:0x6838]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_MVP_INBAND_OUT_MODE	1:0	0x0	00 - disable inband insertion; 01 - used for debug only: insert register MVP_INBAND_CNTL_CHAR_INSERT; 10 - normal mode: insert the character generated by MVP_mixer
D2CRTC_MVP_INBAND_CNTL_CHAR_INSERT	31:8	0x0	Used for debug only: 24-bit control character for insertion

MVP Insert Inband Control for CRTC2

D2CRTC_MVP_INBAND_CNTL_INSERT_TIMER - RW - 32 bits - [GpuF0MMReg:0x683C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_MVP_INBAND_CNTL_CHAR_INSERT_TIMER	7:0	0x8	The number of clock cycles the character insertion trigger from the line buffer needs to be ahead of end of lines for CRTC to insert the in-band control character

MVP Insert Inband Control Timer for CRTC2

D1CRTC_MVP_CRC_CNTL - RW - 32 bits - [GpuF0MMReg:0x6840]			
Field Name	**Bits**	**Default**	**Description**
MVP_CRC_BLUE_MASK	7:0	0xff	mask bit for blue component

MVP_CRC_GREEN_MASK	15:8	0xff	mask bit for green component
MVP_CRC_RED_MASK	23:16	0xff	mask bit for red component
MVP_CRC_EN	28	0x0	0 - CRC disabled; 1 - CRC enabled

CRC control register for MVP

D1CRTC_MVP_CRC_RESULT - RW - 32 bits - [GpuF0MMReg:0x6844]			
Field Name	**Bits**	**Default**	**Description**
MVP_CRC_BLUE_RESULT (R)	7:0	0x0	CRC result for each frame (DE region only) - Blue component
MVP_CRC_GREEN_RESULT (R)	15:8	0x0	CRC result for each frame (DE region only) - Green component
MVP_CRC_RED_RESULT (R)	23:16	0x0	CRC result for each frame (DE region only) - Red component

CRC result for each frame

D1CRTC_MVP_CRC2_CNTL - RW - 32 bits - [GpuF0MMReg:0x6848]			
Field Name	**Bits**	**Default**	**Description**
MVP_CRC2_BLUE_MASK	7:0	0xff	mask bit for blue component
MVP_CRC2_GREEN_MASK	15:8	0xff	mask bit for green component
MVP_CRC2_RED_MASK	23:16	0xff	mask bit for red component
MVP_CRC2_EN	28	0x0	0 - CRC2 disabled; 1 - CRC2 enabled

CRC2 control register for MVP

D1CRTC_MVP_CRC2_RESULT - RW - 32 bits - [GpuF0MMReg:0x684C]			
Field Name	**Bits**	**Default**	**Description**
MVP_CRC2_BLUE_RESULT (R)	7:0	0x0	CRC2 result for each frame (DE region only) - Blue component
MVP_CRC2_GREEN_RESULT (R)	15:8	0x0	CRC2 result for each frame (DE region only) - Green component
MVP_CRC2_RED_RESULT (R)	23:16	0x0	CRC2 result for each frame (DE region only) - Red component

CRC2 result for each frame

D1CRTC_MVP_CONTROL3 - RW - 32 bits - [GpuF0MMReg:0x6850]			
Field Name	**Bits**	**Default**	**Description**
MVP_RESET_IN_BETWEEN_FRAMES	0	0x1	1 - reset pointers, state machines of the MVP receiving logic between frames
MVP_DDR_SC_AB_SEL	4	0x0	0 - select bundle A in DDR single channel mode, 1 - select bundle B
MVP_DDR_SC_B_START_MODE	8	0x0	0 - Assuming the read & write clocks for meso-FIFO B is meso-chronous; 1 - assuming they are synchronous
MVP_FLOW_CONTROL_OUT_FORCE_ONE	12	0x0	1 - force flow_control_out to 1
MVP_FLOW_CONTROL_OUT_FORCE_ZERO	16	0x0	1 - force flow_control_out to 0

MVP_FLOW_CONTROL_CASCADE_EN	20	0x0	1 - cascade flow control in multi-GPU
MVP_SWAP_48BIT_EN	24	0x0	1 - swap the least & most signficant 24 bits of the data as they read out of the FIFO
MVP_FLOW_CONTROL_IN_CAP (R)	28	0x0	capture flow_control_in, used for diagnostics

MVP Control Register 3

D1CRTC_MVP_RECEIVE_CNT_CNTL1 - RW - 32 bits - [GpuF0MMReg:0x6854]			
Field Name	**Bits**	**Default**	**Description**
MVP_SLAVE_PIXEL_ERROR_CNT (R)	12:0	0x0	Count # of pixels in a line that is wrong, reset by active edge of hsync
MVP_SLAVE_LINE_ERROR_CNT (R)	28:16	0x0	Count # of lines in a frame that is wrong, reset by frame start
MVP_SLAVE_DATA_CHK_EN	31	0x1	Enable line & pixel counter, should be enabled a couple of frames after master is enabled

MVP Receive Counter Control 1

D1CRTC_MVP_RECEIVE_CNT_CNTL2 - RW - 32 bits - [GpuF0MMReg:0x6858]			
Field Name	**Bits**	**Default**	**Description**
MVP_SLAVE_FRAME_ERROR_CNT (R)	12:0	0x0	Count # of frames that is wrong
MVP_SLAVE_FRAME_ERROR_CNT_RESET	31	0x0	Reset MVP_SLAVE_FRAME_ERROR_CNT

MVP Receiver Counter Control 2

D1_MVP_AFR_FLIP_MODE - RW - 32 bits - [GpuF0MMReg:0x6514]			
Field Name	**Bits**	**Default**	**Description**
D1_MVP_AFR_FLIP_MODE	1:0	0x0	10 - real flip; 11 - dummy flip

S/W writes to this register in AFR mode for display 1 page flip

D1_MVP_AFR_FLIP_FIFO_CNTL - RW - 32 bits - [GpuF0MMReg:0x6518]			
Field Name	**Bits**	**Default**	**Description**
D1_MVP_AFR_FLIP_FIFO_NUM_ENTRIES (R)	3:0	0x0	number of valid entries in the AFR flip FIFO
D1_MVP_AFR_FLIP_FIFO_RESET	4	0x0	reset the AFR flip FIFO
D1_MVP_AFR_FLIP_FIFO_RESET_FLAG (R)	8	0x0	sticky bit of the AFR flip fifo reset status
D1_MVP_AFR_FLIP_FIFO_RESET_ACK	12	0x0	clear the DC_LB_MVP_AFR_FLIP_RESET_FLAG register bit

This register controls AFR Flip FIFO in display 1

D1_MVP_FLIP_LINE_NUM_INSERT - RW - 32 bits - [GpuF0MMReg:0x651C]			
Field Name	**Bits**	**Default**	**Description**
D1_MVP_FLIP_LINE_NUM_INSERT_MODE	1:0	0x2	00 - no insertion, 0 is appended; 01 - debug: insert D1_MVP_FLIP_LINE_NUM_INSERT register value; 10 - normal Hsync mode, insert the sum of LB line number + DC_LB_MVP_FLIP_LINE_NUM_OFFSET
D1_MVP_FLIP_LINE_NUM_INSERT	21:8	0x0	used for debug purpose, this is what will be the line number carried to downstream GPUs if D1_MVP_FLIP_LINE_NUM_INSERT_EN is set
D1_MVP_FLIP_LINE_NUM_OFFSET	29:24	0x0	used in normal HSYNC flipping operation. this is the number added to the current LB (desktop) line number for carrying to the downstream GPUs
D1_MVP_FLIP_AUTO_ENABLE	30	0x0	Enabling automatic AFR/SFR flipping for display 1

This register controls line number insertion for the Hsync flipping mode in display 1

2.7.12 Secondary Display Graphics Control Registers

D2GRPH_ENABLE - RW - 32 bits - [GpuF0MMReg:0x6900]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_ENABLE	0	0x1	Secondary graphic enabled. 0=disable 1=enable

Secondary graphic enabled.

D2GRPH_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6904]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_DEPTH	1:0	0x0	Secondary graphic pixel depth. 0=8bpp 1=16bpp 2=32bpp 3=64bpp
D2GRPH_Z	5:4	0x0	Z[1:0] value for tiling

D2GRPH_FORMAT	10:8	0x0	Secondary graphic pixel format. It is used together with D1GRPH_DEPTH to define the graphic pixel format. If (D1GRPH_DEPTH = 0x0)(8 bpp) 0x0 - indexed others - reserved else if (D1GRPH_DEPTH = 0x1)(16 bpp) 0x0 - ARGB 1555 0x1 - RGB 565 0x2 - ARGB 4444 0x3 - Alpha index 88 0x4 - monochrome 16 0x5 - BGRA 5551 others - reserved else if (D1GRPH_DEPTH = 0x2)(32 bpp) 0x0 - ARGB 8888 0x1 - ARGB 2101010 0x2 - 32bpp digital output 0x3 - 8-bit ARGB 2101010 0x4 - BGRA 1010102 0x5 - 8-bit BGRA 1010102 0x6 - RGB 111110 0x7 - BGR 101111 others - reserved else if (D1GRPH_DEPTH = 0x3)(64 bpp) 0x0 - ARGB 16161616 0x1 - 64bpp digital output ARGB[13:2] 0x2 - 64bpp digital output RGB[15:0] 0x3 - 64bpp digital output ARGB[11:0] 0x4 - 64bpp digital output BGR[15:0] others - reserved
D2GRPH_TILE_COMPACT_EN	12	0x0	Enables multichip tile compaction 0=Disable 1=Enable
D2GRPH_ADDRESS_TRANSLATION_ENABLE	16	0x0	Enables display 2 address translation 0=0=physical memory 1=1=virtual memory
D2GRPH_PRIVILEGED_ACCESS_ENABLE	17	0x0	Enables display 2 privileged page access 0=0=no priveledged access 1=1=priveledged access
D2GRPH_ARRAY_MODE	23:20	0x0	Defines the tiling mode 0=ARRAY_LINEAR_GENERAL: Unaligned linear array 1=ARRAY_LINEAR_ALIGNED: Aligned linear array 2=ARRAY_1D_TILED_THIN1: Uses 1D 8x8x1 tiles 3=ARRAY_1D_TILED_THICK: Uses 1D 8x8x4 tiles 4=ARRAY_2D_TILED_THIN1: Uses 8x8x1 macro-tiles 5=ARRAY_2D_TILED_THIN2: Macro-tiles are 2x high 6=ARRAY_2D_TILED_THIN4: Macro-tiles are 4x high 7=ARRAY_2D_TILED_THICK: Uses 8x8x4 macro-tiles 8=ARRAY_2B_TILED_THIN1: uses row bank swapping 9=ARRAY_2B_TILED_THIN2: uses row bank swapping 10=ARRAY_2B_TILED_THIN4: uses row bank swapping 11=ARRAY_2B_TILED_THICK: uses row bank swapping 12=ARRAY_3D_TILED_THIN1: Slices are pipe rotated 13=ARRAY_3D_TILED_THICK: Slices are pipe rotated 14=ARRAY_3B_TILED_THIN1: Slices are pipe rotated 15=ARRAY_3B_TILED_THICK: Slices are pipe rotated

D2GRPH_16BIT_ALPHA_MODE	25:24	0x0	This field is only used if 64 bpp graphics bit depth and graphics/overlay blend using per-pixel alpha from graphics channel. It is used for processing 16 bit alpha. The fixed point graphics alpha value in the frame buffer is always clamped to 0.0 - 1.0 data range. 0x0 - Floating point alpha (1 sign bit, 5 bit exponent, 10 bit mantissa) 0x1 - Fixed point alpha with normalization from 256/256 to 255/255 to represent 1.0 0x2 - Fixed point alpha with no normalization 0x3 - Fixed point alpha using lower 8 bits of frame buffer value, no normalization
D2GRPH_16BIT_FIXED_ALPHA_RANGE	30:28	0x0	This register field is only used if 64 bpp graphics bit depth and D2GRPH_16BIT_ALPHA_MODE = 01 or 10. Also only used if graphics/overlay blend using per-pixel alpha from graphics channel. Final alpha blend value is rounded to 8 bits after optional normalization step (see D2GRPH_16BIT_ALPHA_MODE). 0x0 - Use bits 15:0 of input alpha value for blend alpha 0x1 - Use bits 14:0 of input alpha value for blend alpha 0x2 - Use bits 13:0 of input alpha value for blend alpha 0x3 - Use bits 12:0 of input alpha value for blend alpha 0x4 - Use bits 11:0 of input alpha value for blend alpha 0x5 - Use bits 10:0 of input alpha value for blend alpha 0x6 - Use bits 9:0 of input alpha value for blend alpha 0x7 - Use bits 8:0 of input alpha value for blend alpha

Secondary graphic pixel depth and format.

D2GRPH_LUT_SEL - RW - 32 bits - [GpuF0MMReg:0x6908]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_LUT_SEL	0	0x0	Secondary graphic LUT selection. 0=select LUTA 1=select LUTB
D2GRPH_LUT_10BIT_BYPASS_EN	8	0x0	Enable bypass secondary graphic LUT for 2101010 format 0=Use LUT 1=Bypass LUT when in 2101010 format. Ignored for other formats
D2GRPH_LUT_10BIT_BYPASS_DBL_BUF_EN	16	0x0	Enable double buffer D2GRPH_LUT_10BIT_BYPASS_EN 0=D1GRPH_LUT_10BIT_BYPASS_EN take effect right away 1=D1GRPH_LUT_10BIT_BYPASS_EN are double buffered

Secondary graphic LUT selection.

D2GRPH_SWAP_CNTL - RW - 32 bits - [GpuF0MMReg:0x690C]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_ENDIAN_SWAP	1:0	0x0	MC endian swap select 0=0=none 1=1=8in16(0xaabb=>0xbbaa) 2=2=8in32(0xaabbccdd=>0xddccbbaa) 3=3=8in64(0xaabbccddeeff0011=>0x1100ffeeddccbbaa)

D2GRPH_RED_CROSSBAR	5:4	0x0	Red crossbar select 0=0=select from R 1=1=select from G 2=2=select from B 3=3=select from A
D2GRPH_GREEN_CROSSBAR	7:6	0x0	Green crossbar select 0=0=select from G 1=1=select from B 2=2=select from A 3=3=select from R
D2GRPH_BLUE_CROSSBAR	9:8	0x0	Blue crossbar select 0=0=select from B 1=1=select from A 2=2=select from R 3=3=select from G
D2GRPH_ALPHA_CROSSBAR	11:10	0x0	Alpha crossbar select 0=0=select from A 1=1=select from R 2=2=select from G 3=3=select from B

Endian swap and component reorder control

D2GRPH_PRIMARY_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6910]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_PRIMARY_DFQ_ENABLE	0	0x0	Primary surface address DFQ enable 0=0 = one deep queue mode 1=1 = DFQ mode
D2GRPH_PRIMARY_SURFACE_ADDRESS	31:8	0x0	Secondary surface address for secondary graphics in byte. It is 256 byte aligned.

Secondary surface address for secondary graphics in byte.

D2GRPH_SECONDARY_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6918]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_SECONDARY_DFQ_ENABLE *(mirror of D2GRPH_PRIMARY_SURFACE_ADDRESS:D2GRPH_PRIMARY_DFQ_ENABLE)*	0	0x0	Secondary surface address DFQ enable 0=0 = one deep queue mode 1=1 = DFQ mode
D2GRPH_SECONDARY_SURFACE_ADDRESS	31:8	0x0	Secondary surface address for secondary graphics in byte. It is 256 byte aligned.

Secondary surface address for secondary graphics in byte.

D2GRPH_PITCH - RW - 32 bits - [GpuF0MMReg:0x6920]			
Field Name	**Bits**	**Default**	**Description**

D2GRPH_PITCH	13:0	0x0	Secondary graphic surface pitch in pixels. For Micro-tiled/Macro-tiled surface, it must be multiple of 64 pixeld in 8bpp mode. For Micro-linear/Macro-tiled surface, it must be multiple of 256 pixeld in 8bpp mode, multiple of 128 pixels in 16bpp mode and multiple of 64 pixels in 32bpp mode. For Micro-linear/Macro-linear surface, it must be multiple of 64 pixels in 8bpp mode. For other modes, it must be multiple of 32. NOTE: Bits 0:4 of this field are hardwired to ZERO.

Secondary graphic surface pitch in pixels.

D2GRPH_SURFACE_OFFSET_X - RW - 32 bits - [GpuF0MMReg:0x6924]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_SURFACE_OFFSET_X	12:0	0x0	Secondary graphic X surface offset. It is 256 pixels aligned. NOTE: Bits 0:7 of this field are hardwired to ZERO.

Secondary graphic X surface offset.

D2GRPH_SURFACE_OFFSET_Y - RW - 32 bits - [GpuF0MMReg:0x6928]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_SURFACE_OFFSET_Y	12:0	0x0	Secondary graphic Y surface offset. It must be even value NOTE: Bit 0 of this field is hardwired to ZERO.

Secondary graphic Y surface offset.

D2GRPH_X_START - RW - 32 bits - [GpuF0MMReg:0x692C]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_X_START	12:0	0x0	Secondary graphic X start coordinate relative to the desktop coordinates.

Secondary graphic X start coordinate relative to the desktop coordinates.

D2GRPH_Y_START - RW - 32 bits - [GpuF0MMReg:0x6930]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_Y_START	12:0	0x0	Secondary graphic Y start coordinate relative to the desktop coordinates.

Secondary graphic Y start coordinate relative to the desktop coordinates.

D2GRPH_X_END - RW - 32 bits - [GpuF0MMReg:0x6934]

Field Name	Bits	Default	Description
D2GRPH_X_END	13:0	0x0	Secondary graphic X end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K

Secondary graphic X end coordinate relative to the desktop coordinates.

D2GRPH_Y_END - RW - 32 bits - [GpuF0MMReg:0x6938]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_Y_END	13:0	0x0	Secondary graphic Y end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K

Secondary graphic Y end coordinate relative to the desktop coordinates.

D2GRPH_UPDATE - RW - 32 bits - [GpuF0MMReg:0x6944]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_MODE_UPDATE_PENDING (R)	0	0x0	Secondary graphic mode register update pending control. It is set to 1 after a host write to graphics mode register. It is cleared after double buffering is done. This signal is only visible through register. The graphics surface register includes: D2GRPH_DEPTH D2GRPH_FORMAT D2GRPH_SWAP_RB D2GRPH_LUT_SEL D2GRPH_LUT_10BIT_BYPASS_EN D2GRPH_ENABLE D2GRPH_X_START D2GRPH_Y_START D2GRPH_X_END D2GRPH_Y_END The mode register double buffering can only occur at vertical retrace. The double buffering occurs when D2GRPH_MODE_UPDATE_PENDING = 1 and D2GRPH_UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC2 is disabled, the registers will be updated instantly. 0=No update pending 1=Update pending
D2GRPH_MODE_UPDATE_TAKEN (R)	1	0x0	Secondary graphics update taken status for mode registers. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0.

D2GRPH_SURFACE_UPDATE_PENDING (R)	2	0x0	Secondary graphic surface register update pending control. If it is set to 1 after a host write to graphics surface register. It is cleared after double buffering is done. It is cleared after double buffering is done. This signal also goes to both the RBBM wait_until and to the CP_RTS_discrete inputs. The graphics surface register includes: D2GRPH_PRIMARY_SURFACE_ADDRESS D2GRPH_SECONDARY_SURFACE_ADDRESS D2GRPH_PITCH D2GRPH_SURFACE_OFFSET_X D2GRPH_SURFACE_OFFSET_Y. If D2GRPH_SURFACE_UPDATE_H_RETRACE_EN = 0, the double buffering occurs in vertical retrace when D2GRPH_SURFACE_UPDATE_PENDING = 1 and D2GRPH_UPDATE_LOCK = 0 and V_UPDATE = 1. Otherwise the double buffering happens at horizontal retrace when D2GRPH_SURFACE_UPDATE_PENDING = 1 and D2GRPH_UPDATE_LOCK = 0 and Data request for last chunk of the line is sent from DCP to DMIF. If CRTC2 is disabled, the registers will be updated instantly.
D2GRPH_SURFACE_UPDATE_TAKEN (R)	3	0x0	Secondary graphics update taken status for surface registers. If D2GRPH_SURFACE_UPDATE_H_RETRACE_EN = 0, it is set to 1 when double buffering occurs and cleared when V_UPDATE = 0. Otherwise, it is active for one clock cycle when double buffering occurs at the horizontal retrace.
D2GRPH_UPDATE_LOCK	16	0x0	Secondary graphic register update lock control. This lock bit control both surface and mode register double buffer 0=Unlocked 1=Locked
D2GRPH_MODE_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D2GRPH mode registers can be updated multiple times in one V_UPDATE period 1=D2GRPH mode registers can only be updated once in one V_UPDATE period
D2GRPH_SURFACE_DISABLE_MULTIPLE_UPDATE	28	0x0	0=D2GRPH surface registers can be updated multiple times in one V_UPDATE period 1=D2GRPH surface registers can only be updated once in one V_UPDATE period

Secondary graphic update control

D2GRPH_FLIP_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6948]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_SURFACE_UPDATE_H_RETRACE_EN	0	0x0	Enable secondary graphic surface register double buffer in horizontal retrace. 0=Vertical retrace flipping 1=Horizontal retrace flipping

Enable secondary graphic surface register double buffer in horizontal retrace

D2GRPH_SURFACE_ADDRESS_INUSE - RW - 32 bits - [GpuF0MMReg:0x694C]			
Field Name	**Bits**	**Default**	**Description**

D2GRPH_SURFACE_ADDRESS_INUSE (R)	31:8	0x0	This register reads back snapshot of secondary graphics surface address used for data request. The address is the signal sent to DMIF and is updated on SOF or horizontal surface update. The snapshot is triggered by writing 1 into field D1CRTC_SNAPSHOT_MANUAL_TRIGGER of CRTC register D1CRTC_SNAPSHOT_STATUS.

Snapshot of secondary graphics surface address in use

2.7.13 Secondary Display Video Overlay Control Registers

D2OVL_ENABLE - RW - 32 bits - [GpuF0MMReg:0x6980]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_ENABLE	0	0x0	Secondary overlay enabled. 0=disable 1=enable

Secondary overlay enabled.

D2OVL_CONTROL1 - RW - 32 bits - [GpuF0MMReg:0x6984]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_DEPTH	1:0	0x0	Secondary overlay pixel depth 0=reserved 1=16bpp 2=32bpp 3=reserved
D2OVL_Z	5:4	0x0	Z[1:0] value for tiling
D2OVL_FORMAT	10:8	0x0	Secondary overlay pixel format. It is used together with D1OVL_DEPTH to define the overlay format. If (D1OVL_DEPTH = 0x1)(16 bpp) 0x0- ARGB 1555 0x1 - RGB 565 0x2 - BGRA 5551 others - reserved else if (D1OVL_DEPTH = 0x2)(32 bpp) 0x0 - ACrYCb 8888 or ARGB 8888 0x1 - ACrYCb 2101010 or ARGB 2101010 0x2 - CbACrA or BGRA 1010102 others - reserved
D2OVL_TILE_COMPACT_EN	12	0x0	Enables multichip tile compaction 0=Enables multichip tile compaction
D2OVL_ADDRESS_TRANSLATION_ENABLE	16	0x0	enables Overlay 2 address translation 0=0: physical memory 1=1: virtual memory
D2OVL_PRIVILEGED_ACCESS_ENABLE	17	0x0	Enables overlay 2 privileged page access 0=0: no privileged access 1=1: privileged access

D2OVL_ARRAY_MODE	23:20	0x0	Defines the tiling mode 0=ARRAY_LINEAR_GENERAL: Unaligned linear array 1=ARRAY_LINEAR_ALIGNED: Aligned linear array 2=ARRAY_1D_TILED_THIN1: Uses 1D 8x8x1 tiles 3=ARRAY_1D_TILED_THICK: Uses 1D 8x8x4 tiles 4=ARRAY_2D_TILED_THIN1: Uses 8x8x1 macro-tiles 5=ARRAY_2D_TILED_THIN2: Macro-tiles are 2x high 6=ARRAY_2D_TILED_THIN4: Macro-tiles are 4x high 7=ARRAY_2D_TILED_THICK: Uses 8x8x4 macro-tiles 8=ARRAY_2B_TILED_THIN1: uses row bank swapping 9=ARRAY_2B_TILED_THIN2: uses row bank swapping 10=ARRAY_2B_TILED_THIN4: uses row bank swapping 11=ARRAY_2B_TILED_THICK: uses row bank swapping 12=ARRAY_3D_TILED_THIN1: Slices are pipe rotated 13=ARRAY_3D_TILED_THICK: Slices are pipe rotated 14=ARRAY_3B_TILED_THIN1: Slices are pipe rotated 15=ARRAY_3B_TILED_THICK: Slices are pipe rotated
D2OVL_COLOR_EXPANSION_MODE	24	0x0	Secondary overlay pixel format expansion mode. 0=dynamic expansion for RGB 1=zero expansion for YCbCr

Secondary overlay pixel depth and format.

D2OVL_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x6988]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_HALF_RESOLUTION_ENABLE	0	0x0	Secondary overlay half resolution control 0=disable 1=enable

Secondary overlay half resolution control

D2OVL_SWAP_CNTL - RW - 32 bits - [GpuF0MMReg:0x698C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_ENDIAN_SWAP	1:0	0x0	MC endian swap select 0=0=none 1=1=8in16(0xaabb=>0xbbaa) 2=2=8in32(0xaabbccdd=>0xddccbbaa) 3=3=8in64(0xaabbccddeeff0011=>0x1100ffeeddccbbaa)
D2OVL_RED_CROSSBAR	5:4	0x0	Red Crossbar select 0=0=select from R 1=1=select from G 2=2=select from B 3=3=select from A
D2OVL_GREEN_CROSSBAR	7:6	0x0	Green Crossbar select 0=0=select from G 1=1=select from B 2=2=select from A 3=3=select from R
D2OVL_BLUE_CROSSBAR	9:8	0x0	Blue Crossbar select 0=0=select from B 1=1=select from A 2=2=select from R 3=3=select from G

D2OVL_ALPHA_CROSSBAR	11:10	0x0	Alpha Crossbar select 0=0=select from A 1=1=select from R 2=2=select from G 3=3=select from B

Endian swap and component reorder control

D2OVL_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6990]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_DFQ_ENABLE	0	0x0	Secondary overlay surface address DFQ enable
D2OVL_SURFACE_ADDRESS	31:8	0x0	Secondary overlay surface base address in byte. It is 256 bytes aligned.

Secondary overlay surface base address in byte.

D2OVL_PITCH - RW - 32 bits - [GpuF0MMReg:0x6998]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PITCH	13:0	0x0	Secondary overlay surface pitch in pixels. For Micro-tiled/Macro-tiled surface, it must be multiple of 64 pixeld in 8bpp mode. For Micro-linear/Macro-tiled surface, it must be multiple of 256 pixeld in 8bpp mode, multiple of 128 pixels in 16bpp mode and multiple of 64 pixels in 32bpp mode. For Micro-linear/Macro-linear surface, it must be multiple of 64 pixels in 8bpp mode. For other modes, it must be multiple of 32. NOTE: Bits 0:4 of this field are hardwired to ZERO.

Secondary overlay surface pitch in pixels.

D2OVL_SURFACE_OFFSET_X - RW - 32 bits - [GpuF0MMReg:0x699C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_SURFACE_OFFSET_X	12:0	0x0	Secondary overlay X surface offset. It is 256 pixels aligned. NOTE: Bits 0:7 of this field are hardwired to ZERO.

Secondary overlay X surface offset.

D2OVL_SURFACE_OFFSET_Y - RW - 32 bits - [GpuF0MMReg:0x69A0]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_SURFACE_OFFSET_Y	12:0	0x0	Secondary overlay Y surface offset. It is even value. NOTE: Bit 0 of this field is hardwired to ZERO.

Secondary overlay Y surface offset.

D2OVL_START - RW - 32 bits - [GpuF0MMReg:0x69A4]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_Y_START	12:0	0x0	Secondary overlay Y start coordinate relative to the desktop coordinates.
D2OVL_X_START	28:16	0x0	Secondary overlay X start coordinate relative to the desktop coordinates.

Secondary overlay X, Y start coordinate relative to the desktop coordinates.

D2OVL_END - RW - 32 bits - [GpuF0MMReg:0x69A8]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_Y_END	13:0	0x0	Secondary overlay Y end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K
D2OVL_X_END	29:16	0x0	Secondary overlay X end coordinate relative to the desktop coordinates. It is exclusive and the maximum value is 8K

Secondary overlay X, Y end coordinate relative to the desktop coordinates.

D2OVL_UPDATE - RW - 32 bits - [GpuF0MMReg:0x69AC]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_UPDATE_PENDING (R)	0	0x0	Secondary overlay register update pending control. It is set to 1 after a host write to overlay double buffer register. It is cleared after double buffering is done. The double buffering occurs when UPDATE_PENDING = 1 and UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC2 is disabled, the registers will be updated instantly. D2OVL double buffer registers include: D2OVL_ENABLE D2OVL_DEPTH D2OVL_FORMAT D2OVL_SWAP_RB D2OVL_COLOR_EXPANSION_MODE D2OVL_HALF_RESOLUTION_ENABLE D2OVL_SURFACE_ADDRESS D2OVL_PITCH D2OVL_SURFACE_OFFSET_X D2OVL_SURFACE_OFFSET_Y D2OVL_START D2OVL_END 0=No update pending 1=Update pending
D2OVL_UPDATE_TAKEN (R)	1	0x0	Secondary overlay update taken status. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0.
D2OVL_UPDATE_LOCK	16	0x0	Secondary overlay register update lock control. 0=Unlocked 1=Locked
D2OVL_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D2OVL registers can be updated multiple times in one V_UPDATE period 1=D2OVL registers can only be updated once in one V_UPDATE period

Secondary overlay register update

D2OVL_SURFACE_ADDRESS_INUSE - RW - 32 bits - [GpuF0MMReg:0x69B0]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_SURFACE_ADDRESS_INUSE (R)	31:8	0x0	This register reads back snapshot of secondary overlay surface address used for data request. The address is the signal sent to DMIF and is updated on SOF or horizontal surface update. The snapshot is triggered by writing 1 into field D1CRTC_SNAPSHOT_MANUAL_TRIGGER of CRTC register D1CRTC_SNAPSHOT_STATUS.

Snapshot of secondary overlay surface address in use

D2OVL_DFQ_CONTROL - RW - 32 bits - [GpuF0MMReg:0x69B4]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_DFQ_RESET	0	0x0	Reset the deep flip queue
D2OVL_DFQ_SIZE	6:4	0x0	Size of the deep flip queue: 0 = 1 deep queue, 1 = 2 deep queue,..., 7 = 8 deep queue
D2OVL_DFQ_MIN_FREE_ENTRIES	10:8	0x0	Minimum # of free entries before surface pending is asserted

Control of the deep flip queue for D2 overlay

D2OVL_DFQ_STATUS - RW - 32 bits - [GpuF0MMReg:0x69B8]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_DFQ_NUM_ENTRIES (R)	3:0	0x0	# of entries in deep flip queue. 0 = 1 entry, 1 = 2 entries, ... 7 = 8 entries
D2OVL_DFQ_RESET_FLAG (R)	8	0x0	Sticky bit: Deep flip queue in reset
D2OVL_DFQ_RESET_ACK (W)	9	0x0	Clear D2OVL_DFQ_RESET_FLAG

Status of the deep flip queue for D2 overlay

2.7.14 Secondary Display Video Overlay Transform Registers

D2OVL_MATRIX_TRANSFORM_EN - RW - 32 bits - [GpuF0MMReg:0x6A00]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_TRANSFORM_EN	0	0x0	Secondary overlay matrix conversion enable 0=disable 1=enable

Secondary overlay matrix conversion enable.

D2OVL_MATRIX_COEF_1_1 - RW - 32 bits - [GpuF0MMReg:0x6A04]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_1_1	18:0	0x198a0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_1_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_1_2 - RW - 32 bits - [GpuF0MMReg:0x6A08]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_1_2	18:0	0x12a20	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_1_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_1_3 - RW - 32 bits - [GpuF0MMReg:0x6A0C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_1_3	18:0	0x0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_1_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_1_4 - RW - 32 bits - [GpuF0MMReg:0x6A10]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_1_4	26:8	0x48700	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S11.1. NOTE: Bits 0:6 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_1_4	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_2_1 - RW - 32 bits - [GpuF0MMReg:0x6A14]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_2_1	18:0	0x72fe0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_2_1	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_2_2 - RW - 32 bits - [GpuF0MMReg:0x6A18]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_2_2	18:0	0x12a20	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_2_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_2_3 - RW - 32 bits - [GpuF0MMReg:0x6A1C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_2_3	18:0	0x79bc0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_2_3	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_2_4 - RW - 32 bits - [GpuF0MMReg:0x6A20]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_2_4	26:8	0x22100	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S11.1. NOTE: Bits 0:6 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_2_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_3_1 - RW - 32 bits - [GpuF0MMReg:0x6A24]

Field Name	Bits	Default	Description
D2OVL_MATRIX_COEF_3_1	18:0	0x0	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_3_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_3_2 - RW - 32 bits - [GpuF0MMReg:0x6A28]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_3_2	18:0	0x12a20	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_3_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_3_3 - RW - 32 bits - [GpuF0MMReg:0x6A2C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_3_3	18:0	0x20460	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S3.11. NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_3_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_MATRIX_COEF_3_4 - RW - 32 bits - [GpuF0MMReg:0x6A30]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_MATRIX_COEF_3_4	26:8	0x3af80	Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay. Format fix-point S11.1. NOTE: Bits 0:6 of this field are hardwired to ZERO.
D2OVL_MATRIX_SIGN_3_4	31	0x1	Sign bit of combined matrix constant

Combined matrix constant of YCbCr->RGB, contrast and brightness adjustment for secondary overlay.

D2OVL_COLOR_MATRIX_TRANSFORMATION_CNTL - RW - 32 bits - [GpuF0MMReg:0x6940]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	2:0	0x0	Matrix transformation control for secondary display overlay pixels. It is used when PIX_TYPE is 0.

Matrix transformation control for secondary display overlay pixels.

2.7.15 Secondary Display Video Overlay Gamma Correction Registers

D2OVL_PWL_TRANSFORM_EN - RW - 32 bits - [GpuF0MMReg:0x6A80]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_TRANSFORM_EN	0	0x0	Secondary overlay gamma correction enable. 0=disable 1=enable

Secondary overlay gamma correction enable.

D2OVL_PWL_0TOF - RW - 32 bits - [GpuF0MMReg:0x6A84]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_0TOF_OFFSET	8:0	0x0	Secondary overlay gamma correction non-linear offset for input 0x0-0xF. Format fix-point 8.1 (0.0 to +255.5).
D2OVL_PWL_0TOF_SLOPE	26:16	0x100	Secondary overlay gamma correction non-linear slope for input 0x0-0xF. Format fix-point 3.8 (0.00 to +7.99).

Secondary overlay gamma correction non-linear offset and slope for input 0x0-0xF

D2OVL_PWL_10TO1F - RW - 32 bits - [GpuF0MMReg:0x6A88]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_10TO1F_OFFSET	8:0	0x20	Secondary overlay gamma correction non-linear offset for input 0x10-0x1F. Format fix-point 8.1 (0.0 to +255.5).
D2OVL_PWL_10TO1F_SLOPE	26:16	0x100	Secondary overlay gamma correction non-linear slope for input 0x10-0x1F. Format fix-point 3.8 (0.00 to +7.99).

Secondary overlay gamma correction non-linear offset and slope for input 0x10-0x1F

D2OVL_PWL_20TO3F - RW - 32 bits - [GpuF0MMReg:0x6A8C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_20TO3F_OFFSET	9:0	0x40	Secondary overlay gamma correction non-linear offset for input 0x20-0x3F. Format fix-point 9.1 (0.0 to +511.5).
D2OVL_PWL_20TO3F_SLOPE	25:16	0x100	Secondary overlay gamma correction non-linear slope for input 0x20-0x3F. Format fix-point 2.8 (0.00 to +3.99).

Secondary overlay gamma correction non-linear offset and slope for input 0x20-0x3F

D2OVL_PWL_40TO7F - RW - 32 bits - [GpuF0MMReg:0x6A90]			
Field Name	**Bits**	**Default**	**Description**

D2OVL_PWL_40TO7F_OFFSET	9:0	0x80	Secondary overlay gamma correction non-linear offset for input 40-7F. Format fix-point 9.1 (0.0 to +511.5).
D2OVL_PWL_40TO7F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 40-7F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 40-7F.

D2OVL_PWL_80TOBF - RW - 32 bits - [GpuF0MMReg:0x6A94]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_80TOBF_OFFSET	10:0	0x100	Secondary overlay gamma correction non-linear offset for input 80-BF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_80TOBF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 80-BF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 80-BF.

D2OVL_PWL_C0TOFF - RW - 32 bits - [GpuF0MMReg:0x6A98]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_C0TOFF_OFFSET	10:0	0x180	Secondary overlay gamma correction non-linear offset for input C0-FF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_C0TOFF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input C0-FF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input C0-FF.

D2OVL_PWL_100TO13F - RW - 32 bits - [GpuF0MMReg:0x6A9C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_100TO13F_OFFSET	10:0	0x200	Secondary overlay gamma correction non-linear offset for input 100-13F. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_100TO13F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 100-13F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 100-13F.

D2OVL_PWL_140TO17F - RW - 32 bits - [GpuF0MMReg:0x6AA0]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_140TO17F_OFFSET	10:0	0x280	Secondary overlay gamma correction non-linear offset for input 140-17F. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_140TO17F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 140-17F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 140-17F.

D2OVL_PWL_180TO1BF - RW - 32 bits - [GpuF0MMReg:0x6AA4]

Field Name	Bits	Default	Description
D2OVL_PWL_180TO1BF_OFFSET	10:0	0x300	Secondary overlay gamma correction non-linear offset for input 180-1BF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_180TO1BF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 180-1BF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 180-1BF.

D2OVL_PWL_1C0TO1FF - RW - 32 bits - [GpuF0MMReg:0x6AA8]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_1C0TO1FF_OFFSET	10:0	0x380	Secondary overlay gamma correction non-linear offset for input 1C0-1FF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_1C0TO1FF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 1C0-1FF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 1C0-1FF.

D2OVL_PWL_200TO23F - RW - 32 bits - [GpuF0MMReg:0x6AAC]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_200TO23F_OFFSET	10:0	0x400	Secondary overlay gamma correction non-linear offset for input 200-23F. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_200TO23F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 200-23F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 200-23F.

D2OVL_PWL_240TO27F - RW - 32 bits - [GpuF0MMReg:0x6AB0]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_240TO27F_OFFSET	10:0	0x480	Secondary overlay gamma correction non-linear offset for input 240-27F. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_240TO27F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 240-27F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 240-27F.

D2OVL_PWL_280TO2BF - RW - 32 bits - [GpuF0MMReg:0x6AB4]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_280TO2BF_OFFSET	10:0	0x500	Secondary overlay gamma correction non-linear offset for input 280-2BF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_280TO2BF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 280-2BF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 280-2BF.

D2OVL_PWL_2C0TO2FF - RW - 32 bits - [GpuF0MMReg:0x6AB8]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_2C0TO2FF_OFFSET	10:0	0x580	Secondary overlay gamma correction non-linear offset for input 2C0-2FF. Format fix-point 10.1(0.0 to +1023.5).
D2OVL_PWL_2C0TO2FF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 2C0-2FF. Format fix-point 1.8(0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 2C0-2FF.

D2OVL_PWL_300TO33F - RW - 32 bits - [GpuF0MMReg:0x6ABC]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_300TO33F_OFFSET	10:0	0x600	Secondary overlay gamma correction non-linear offset for input 300-33F. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_300TO33F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 300-33F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 300-33F.

D2OVL_PWL_340TO37F - RW - 32 bits - [GpuF0MMReg:0x6AC0]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_340TO37F_OFFSET	10:0	0x680	Secondary overlay gamma correction non-linear offset for input 340-37F. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_340TO37F_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 340-37F. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 340-37F.

D2OVL_PWL_380TO3BF - RW - 32 bits - [GpuF0MMReg:0x6AC4]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_380TO3BF_OFFSET	10:0	0x700	Secondary overlay gamma correction non-linear offset for input 380-3BF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_380TO3BF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 380-3BF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 380-3BF.

D2OVL_PWL_3C0TO3FF - RW - 32 bits - [GpuF0MMReg:0x6AC8]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_PWL_3C0TO3FF_OFFSET	10:0	0x780	Secondary overlay gamma correction non-linear offset for input 3C0-3FF. Format fix-point 10.1 (0.0 to +1023.5).
D2OVL_PWL_3C0TO3FF_SLOPE	24:16	0x100	Secondary overlay gamma correction non-linear slope for input 3C0-3FF. Format fix-point 1.8 (0.00 to +1.99).

Secondary overlay gamma correction non-linear offset and slope for input 3C0-3FF.

D2OVL_KEY_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6B00]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_KEY_FUNCTION	1:0	0x0	Selects graphic keyer result equation for secondary display. 0=GRPH2_KEY = FALSE = 0 1=GRPH2_KEY = TRUE = 1 2=GPPH2_KEY = (GRPH2_RED in range) AND (GRPH2_GREEN in range) AND (GRPH2_BLUE in range) AND (GRPH2_ALPHA in range) 3=GRPH2_KEY = not [(GRPH2_RED in range) AND (GRPH2_GREEN in range) AND (GRPH2_BLUE in range) AND (GRPH2_ALPHA in range)]
D2OVL_KEY_FUNCTION	9:8	0x0	Selects overlay keyer result equation for secondary display. 0=OVL2_KEY = FALSE = 0 1=OVL2_KEY = TRUE = 1 2=OVL2_KEY = (OVL2_Cr_RED in range) AND (OVL2_Y_GREEN in range) AND (OVL2_Cb_BLUE in range) AND (OVL2_ALPHA in range) 3=OVL2_KEY = not [(OVL2_Cr_RED in range) AND (OVL2_Y_GREEN in range) AND (OVL2_Cb_BLUE in range) AND (OVL2_ALPHA in range)]
D2OVL_KEY_COMPARE_MIX	16	0x0	Selects final mix of graphics and overlay keys for secondary display. 0=GRPH_OVL_KEY = GRPH_KEY or OVL_KEY 1=GRPH_OVL_KEY = GRPH_KEY and OVL_KEY

Secondary display key control

2.7.16 Secondary Display Graphics and Overlay Blending Registers

D2GRPH_ALPHA - RW - 32 bits - [GpuF0MMReg:0x6B04]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_ALPHA	7:0	0xff	Global graphic alpha for use in key mode and global alpha modes. See D2OVL_ALPHA_MODE register filed for more details

Global graphic alpha for use in key mode and global alpha modes.

D2OVL_ALPHA - RW - 32 bits - [GpuF0MMReg:0x6B08]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_ALPHA	7:0	0xff	Global overlay alpha for use in key mode and global alpha modes. See D2OVL_ALPHA_MODE register filed for more details

Global overlay alpha for use in key mode and global alpha modes.

D2OVL_ALPHA_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6B0C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_ALPHA_MODE	1:0	0x0	Graphics/overlay alpha blending mode for secondary controller. In any case, if there is only graphics, the input OVL_DATA is forced to blank. If there is only overlay, the input GRPH_DATA is forced to blank. 0=Keyer mode, select graphic or overlay keyer to mix graphics and overlay 1=Per pixel graphic alpha mode.Alpha blend graphic and overlay layer. The alpha from graphic pixel may be inverted according to register field 2=Global alpha mode 3=Per pixel overlay alpha mode
D2OVL_ALPHA_PREMULT	8	0x0	For use with per pixel alpha blend mode. Selects whether pre-multiplied alpha or non-multiplied alpha. 0=0x0 - When DxOVL_ALPHA_MODE = 0x1, then Pixel = PIX_ALPHA * graphics pixel + (1-PIX_ALPHA) * overlay pixel.When DxOVL_ALPHA_MODE = 0x3, then Pixel = PIX_ALPHA * overlay pixel + (1-PIX_ALPHA) * graphic pixel 1=0x1 - When DxOVL_ALPHA_MODE = 0x1, then Pixel = graphic pixel + (1-PIX_ALPHA) * overlay pixel.When DxOVL_ALPHA_MODE = 0x3, then Pixel = overlay pixel + (1-PIX_ALPHA) * graphic pixel
D2OVL_ALPHA_INV	16	0x0	For use with pixel blend mode. Apply optional inversion to the alpha value extracted form the graphics or overlay surface data. 0=PIX_ALPHA = alpha from graphics or overlay 1=PIX_ALPHA = 1 - alpha from graphics or overlay

Secondary display graphics/overlay alpha blending control

D2GRPH_KEY_RANGE_RED - RW - 32 bits - [GpuF0MMReg:0x6B10]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_KEY_RED_LOW	15:0	0x0	Secondary graphics keyer red component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D2GRPH_KEY_RED_HIGH	31:16	0x0	Secondary graphics keyer red component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Secondary graphics keyer red component range

D2GRPH_KEY_RANGE_GREEN - RW - 32 bits - [GpuF0MMReg:0x6B14]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_KEY_GREEN_LOW	15:0	0x0	Secondary graphics keyer green component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

D2GRPH_KEY_GREEN_HIGH	31:16	0x0	Secondary graphics keyer green component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Secondary graphics keyer green component range

D2GRPH_KEY_RANGE_BLUE - RW - 32 bits - [GpuF0MMReg:0x6B18]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_KEY_BLUE_LOW	15:0	0x0	Secondary graphics keyer blue component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D2GRPH_KEY_BLUE_HIGH	31:16	0x0	Secondary graphics keyer blue component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Secondary graphics keyer blue component range

D2GRPH_KEY_RANGE_ALPHA - RW - 32 bits - [GpuF0MMReg:0x6B1C]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_KEY_ALPHA_LOW	15:0	0x0	Secondary graphics keyer alpha component lower limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.
D2GRPH_KEY_ALPHA_HIGH	31:16	0x0	Secondary graphics keyer alpha component upper limit. Note: If the graphic component is less than 16 bit, msbs are all zeros.

Secondary graphics keyer alpha component range

D2OVL_KEY_RANGE_RED_CR - RW - 32 bits - [GpuF0MMReg:0x6B20]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_KEY_RED_CR_LOW	9:0	0x0	Secondary overlay keyer red component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D2OVL_KEY_RED_CR_HIGH	25:16	0x0	Secondary overlay keyer red component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Secondary overlay keyer red component range

D2OVL_KEY_RANGE_GREEN_Y - RW - 32 bits - [GpuF0MMReg:0x6B24]

Field Name	Bits	Default	Description
D2OVL_KEY_GREEN_Y_LOW	9:0	0x0	Secondary overlay keyer green component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D2OVL_KEY_GREEN_Y_HIGH	25:16	0x0	Secondary overlay keyer green component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Secondary overlay keyer green component range

D2OVL_KEY_RANGE_BLUE_CB - RW - 32 bits - [GpuF0MMReg:0x6B28]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_KEY_BLUE_CB_LOW	9:0	0x0	Secondary overlay keyer blue component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D2OVL_KEY_BLUE_CB_HIGH	25:16	0x0	Secondary overlay keyer blue component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Secondary overlay keyer blue component range

D2OVL_KEY_ALPHA - RW - 32 bits - [GpuF0MMReg:0x6B2C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_KEY_ALPHA_LOW	7:0	0x0	Secondary overlay keyer alpha component lower limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.
D2OVL_KEY_ALPHA_HIGH	23:16	0x0	Secondary overlay keyer alpha component upper limit. Note: If the overlay component is less than 16 bit, msbs are all zeros.

Secondary overlay keyer alpha component range

2.7.17 Secondary Display Color Matrix Transform Registers

D2GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL - RW - 32 bits - [GpuF0MMReg:0x6B80]			
Field Name	**Bits**	**Default**	**Description**
D2GRPH_COLOR_MATRIX_TRANSFORMATION_EN	0	0x0	Matrix transformation control for secondary display graphics and cursor pixel. It is used when PIX_TYPE is 1. 0=disable 1=enable

Matrix transformation control for secondary display graphics and cursor pixel.

D2COLOR_MATRIX_COEF_1_1 - RW - 32 bits - [GpuF0MMReg:0x6B84]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_1_1	16:0	0x0	Combined matrix constant C11 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S1.11(-2.00 to +1.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_1_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C11 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_1_2 - RW - 32 bits - [GpuF0MMReg:0x6B88]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_1_2	15:0	0x0	Combined matrix constant C12 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S0.11(-1.00 to + 0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_1_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C12 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_1_3 - RW - 32 bits - [GpuF0MMReg:0x6B8C]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_1_3	15:0	0x0	Combined matrix constant C13 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S0.11(-1.0 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_1_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C13 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_1_4 - RW - 32 bits - [GpuF0MMReg:0x6B90]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_1_4	26:8	0x0	Combined matrix constant C14 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S11.1(-2048.5 to +2047.5). It includes subtraction of 512 offset NOTE: Bits 0:6 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_1_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C14 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_2_1 - RW - 32 bits - [GpuF0MMReg:0x6B94]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_2_1	15:0	0x0	Combined matrix constant C21 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_2_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C21 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_2_2 - RW - 32 bits - [GpuF0MMReg:0x6B98]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_2_2	16:0	0x0	Combined matrix constant C22 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S1.11(-2.00 to +1.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_2_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C22 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_2_3 - RW - 32 bits - [GpuF0MMReg:0x6B9C]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_2_3	15:0	0x0	Combined matrix constant C23 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_2_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C23 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_2_4 - RW - 32 bits - [GpuF0MMReg:0x6BA0]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_2_4	26:8	0x0	Combined matrix constant C24 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S11.1(-2048.5 to +2047.5). It includes subtraction of 512 offset NOTE: Bits 0:6 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_2_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C24 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_3_1 - RW - 32 bits - [GpuF0MMReg:0x6BA4]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_3_1	15:0	0x0	Combined matrix constant C31 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_3_1	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C31 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_3_2 - RW - 32 bits - [GpuF0MMReg:0x6BA8]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_3_2	15:0	0x0	Combined matrix constant C32 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S0.11(-1.00 to +0.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_3_2	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C32 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_3_3 - RW - 32 bits - [GpuF0MMReg:0x6BAC]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_3_3	16:0	0x0	Combined matrix constant C33 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S1.11(-2.00 to +1.99). NOTE: Bits 0:4 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_3_3	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C33 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

D2COLOR_MATRIX_COEF_3_4 - RW - 32 bits - [GpuF0MMReg:0x6BB0]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_MATRIX_COEF_3_4	26:8	0x0	Combined matrix constant C34 of RGB->YCbCr, contrast and brightness adjustment for secondary display. Format fix-point S11.1(-2048.5 to +2047.5). It includes subtraction of 512 offset NOTE: Bits 0:6 of this field are hardwired to ZERO.
D2COLOR_MATRIX_SIGN_3_4	31	0x0	Sign bit of combined matrix constant

Combined matrix constant C34 of RGB->YCbCr, contrast and brightness adjustment for secondary display.

2.7.18 Secondary Display Subsampling Registers

D2COLOR_SPACE_CONVERT - RW - 32 bits - [GpuF0MMReg:0x693C]			
Field Name	**Bits**	**Default**	**Description**
D2COLOR_SUBSAMPLE_CRCB_MODE	1:0	0x0	Sub-sampling control for secondary display 0=do not subsample CrCb(RB) 1=subsample CrCb (RB) by using 2 tap average method 2=subsample CrCb (RB) by using 1 tap on even pixel 3=subsample CrCb (RB) by using 1 tap on odd pixel

Sub-sampling control for secondary display.

2.7.19 Secondary Display Realtime Overlay Registers

D2OVL_RT_SKEWCOMMAND - RW - 32 bits - [GpuF0MMReg:0x6D00]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_RT_CLEAR_GOBBLE_COUNT (W)	0	0x0	writing 1 to this bit clear the gobbleCount this bit has higher priority than inc_gobblecount
D2OVL_RT_INC_GOBBLE_COUNT (W)	4	0x0	writing 1 to this bit increments the gobbleCount
D2OVL_RT_CLEAR_SUBMIT_COUNT (W)	8	0x0	writing 1 to this bit clear the submitCount this bit has higher priority than inc_submitcount
D2OVL_RT_INC_SUBMIT_COUNT (W)	12	0x0	writing 1 to this bit increments the submitCount
D2OVL_RT_GOBBLE_COUNT (R)	18:16	0x0	read only register gobble count value which increments with each inc_gobble_count and reset with clear_gobble_count commands. it wraps around on overflow during increment.
D2OVL_RT_SUBMIT_COUNT (R)	22:20	0x0	read only register submit count value which increments with each inc_submit_count and reset with clear_submit_count commands. it wraps around on overflow during increment.

reset or increment submit and gobble count

D2OVL_RT_SKEWCONTROL - RW - 32 bits - [GpuF0MMReg:0x6D04]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_RT_CAPS	2:0	0x0	max value in submitCount and gobbleCount this is the number of contents buffer - 1 should reset counters before programming this field
D2OVL_RT_SKEW_MAX	6:4	0x0	max skew allowed between gobbleCount and submitCount

controls for submit and gobble counts

D2OVL_RT_BAND_POSITION - RW - 32 bits - [GpuF0MMReg:0x6D08]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_RT_TOP_SCAN	13:0	0x0	define the top scan line for the next RT (inclusive)
D2OVL_RT_BTM_SCAN	29:16	0x0	define the bottom scan line for next RT (exclusive)

The position of the top and bottom scan line for next RT

D2OVL_RT_PROCEED_COND - RW - 32 bits - [GpuF0MMReg:0x6D0C]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_RT_REDUCE_DELAY	0	0x0	0 selects delay optimized scheme 1 selects basic render behind delay scan scheme

D2OVL_RT_RT_FLIP	4	0x0	0 selects bandSync to be exposed to CP 1 selects frameSync to be exposed to CP
D2OVL_RT_PROCEED_ON_EOF_DISABLE	8	0x0	0 enables unfinished bands to pass bandSync on EOF (valid only in basic scheme) 1 disables this feature
D2OVL_RT_WITH_HELD_ON_SOF	12	0x0	0 disables proceedOnEOF on next frameSync 1 disables proceedOnEOF on next SOF
D2OVL_RT_CLEAR_GOBBLE_GO (W)	14	0x0	This bit clear gobbleGo disable another frame submit before next flip (ignored in basic scheme)
D2OVL_RT_TEAR_PROOF_HEIGHT	29:16	0x0	define the number of scan lines above topscan. if display starts reading from there, RT should wait

select RT flip proceed condition

D2OVL_RT_STAT - RW - 32 bits - [GpuF0MMReg:0x6D10]			
Field Name	**Bits**	**Default**	**Description**
D2OVL_RT_FIP_PROCEED_ACK (W)	0	0x0	The sticky bit clears the FIP_PROCEED FLAG flag when written
D2OVL_RT_FRAME_SYNC_ACK (W)	1	0x0	The sticky bit clears the RT_FRAME_SYNC flag when written
D2OVL_RT_OVL_START_ACK (W)	2	0x0	The sticky bit clears the OVL_START FLAG flag when written
D2OVL_RT_BAND_INVISIBLE (R)	8	0x0	Debug bit indicating that overlay scanning in invisble region
D2OVL_RT_BAND_SYNC (R)	9	0x0	Debug bit indicating that overlay bottom scan is less the line counter
D2OVL_RT_EOF_PRPCEED (R)	10	0x0	Debug bit indicating that overlay is ended. Set at eof and reset at overlay start
D2OVL_RT_FIP_PROCEED (R)	11	0x0	Sticky debug bit that set when RT_FLIP_PROCEED signal asserted.
D2OVL_RT_FRAME_SYNC (R)	12	0x0	Sticky debug bit indicating that overlay start set and a new submission occured
D2OVL_RT_GOBBLE_GO (R)	13	0x0	Debug bit that set on frame_sync and clear at gobbleclr
D2OVL_RT_NEW_SUBMIT (R)	14	0x0	Debug bit indicating a new submission occurred
D2OVL_RT_OVL_START (R)	15	0x0	Debug bit indicating that line buffer detects start of overlay being accessed
D2OVL_RT_OVL_ENDED (R)	16	0x0	Debug bit indicating that line buffer detects that the end of overlay being accessed
D2OVL_RT_SAFE_ZONE (R)	17	0x0	Debug bit indicating that overlay is scaning in safe zone
D2OVL_RT_SWITCH_REGIONS (R)	18	0x0	Debug bit showing the postion of scan region relative to display
D2OVL_SKEW_MAX_REACHED (R)	19	0x0	Debug bit indicating that line buffer detected maximum skew reached
D2OVL_LINE_COUNTER (R)	31:20	0x0	debug bit showing display line counter value

Status Bits

2.7.20 Secondary Display Hardware Cursor Registers

D2CUR_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6C00]			
Field Name	**Bits**	**Default**	**Description**

D2CURSOR_EN	0	0x0	Secondary display hardware cursor enabled. 0=disable 1=enable
D2CURSOR_MODE	9:8	0x0	Secondary display hardware cursor mode. For 2bpp mode, each line of cursor data is stored in memory as 16 bits of AND data followed by 16 bits XOR data. For color AND/XOR mode, each pixel is stored sequentially in memory as 32bits each in aRGB8888 format with bit 31 of each DWord being the AND bit. For the color alpha modes the format is also 32bpp aRGB8888 with all 8 bits of the alpha being used.All HW cursor lines must be 64 pixels wide and all lines must be stored sequentially in memory. 0=Mono (2bpp) 1=Color 24bpp + 1 bit AND (32bpp) 2=Color 24bpp + 8 bit alpha (32bpp) premultiplied alpha 3=Color 24bpp + 8 bit alpha (32bpp)unmultiplied alpha
D2CURSOR_2X_MAGNIFY	16	0x0	Secondary display hardware cursor 2x2 magnification. 0=no 2x2 magnification 1=2x2 magnification in horizontal and vertical direction
D2CURSOR_FORCE_MC_ON	20	0x0	When set, if the incoming data is in D1 cursor region, DCP_LB_cursor1_allow_stutter is set. This field in this double bufferred register is not double buffered.

Secondary display hardware control

D2CUR_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6C08]			
Field Name	**Bits**	**Default**	**Description**
D2CURSOR_SURFACE_ADDRESS	31:0	0x0	Secondary display hardware cursor surface base address in byte. It is 4K byte aligned. NOTE: Bits 0:11 of this field are hardwired to ZERO.

Secondary display hardware cursor surface base address.

D2CUR_SIZE - RW - 32 bits - [GpuF0MMReg:0x6C10]			
Field Name	**Bits**	**Default**	**Description**
D2CURSOR_HEIGHT	5:0	0x0	Secondary display hardware cursor height minus 1.
D2CURSOR_WIDTH	21:16	0x0	Secondary display hardware cursor width minus 1.

Secondary display hardware size

D2CUR_POSITION - RW - 32 bits - [GpuF0MMReg:0x6C14]			
Field Name	**Bits**	**Default**	**Description**
D2CURSOR_Y_POSITION	12:0	0x0	Secondary display hardware cursor X coordinate at the hot spot relative to the desktop coordinates.
D2CURSOR_X_POSITION	28:16	0x0	Secondary display hardware cursor X coordinate at the hot spot relative to the desktop coordinates.

Secondary display hardware cursor position

D2CUR_HOT_SPOT - RW - 32 bits - [GpuF0MMReg:0x6C18]			
Field Name	**Bits**	**Default**	**Description**
D2CURSOR_HOT_SPOT_Y	5:0	0x0	Secondary display hardware cursor hot spot X length relative to the top left corner.
D2CURSOR_HOT_SPOT_X	21:16	0x0	Secondary display hardware cursor hot spot Y length relative to the top left corner.

Secondary display hardware cursor hot spot position

D2CUR_COLOR1 - RW - 32 bits - [GpuF0MMReg:0x6C1C]			
Field Name	**Bits**	**Default**	**Description**
D2CUR_COLOR1_BLUE	7:0	0x0	Secondary display hardware cursor blue component of color 1.
D2CUR_COLOR1_GREEN	15:8	0x0	Secondary display hardware cursor green component of color 1.
D2CUR_COLOR1_RED	23:16	0x0	Secondary display hardware cursor red component of color 1.

Secondary display hardware cursor color 1.

D2CUR_COLOR2 - RW - 32 bits - [GpuF0MMReg:0x6C20]			
Field Name	**Bits**	**Default**	**Description**
D2CUR_COLOR2_BLUE	7:0	0x0	Secondary display hardware cursor blue component of color 2.
D2CUR_COLOR2_GREEN	15:8	0x0	Secondary display hardware cursor green component of color 2.
D2CUR_COLOR2_RED	23:16	0x0	Secondary display hardware cursor red component of color 2.

Secondary display hardware cursor color 2.

D2CUR_UPDATE - RW - 32 bits - [GpuF0MMReg:0x6C24]			
Field Name	**Bits**	**Default**	**Description**

D2CURSOR_UPDATE_PENDING (R)	0	0x0	Secondary display hardware cursor update pending status. It is set to 1 after a host write to cursor double buffer register. It is cleared after double buffering is done. The double buffering occurs when D2CURSOR_UPDATE_PENDING = 1 and D2CURSOR_UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC2 is disabled, the registers will be updated instantly. The D2CUR double buffer registers are: D2CURSOR_EN D2CURSOR_MODE D2CURSOR_2X_MAGNIFY D2CURSOR_SURFACE_ADDRESS D2CURSOR_HEIGHT D2CURSOR_WIDTH D2CURSOR_X_POSITION D2CURSOR_Y_POSITION D2CURSOR_HOT_SPOT_X D2CURSOR_HOT_SPOT_Y 0=No update pending 1=Update pending
D2CURSOR_UPDATE_TAKEN (R)	1	0x0	Secondary display hardware cursor update taken status. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0
D2CURSOR_UPDATE_LOCK	16	0x0	Secondary display hardware cursor update lock control. 0=Unlocked 1=Locked
D2CURSOR_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D2CURSOR registers can be updated multiple times in one V_UPDATE period 1=D2CURSOR registers can only be updated once in one V_UPDATE period

2.7.21 Secondary Display Hardware Icon Registers

D2ICON_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6C40]			
Field Name	**Bits**	**Default**	**Description**
D2ICON_ENABLE	0	0x0	Secondary display hardware icon enable. 0=disable 1=enable
D2ICON_2X_MAGNIFY	16	0x0	Secondary display hardware icon 2x2 magnification. 0=no 2x2 magnification 1=2x2 magnification in horizontal and vertical direction
D2ICON_FORCE_MC_ON	20	0x0	When set, if the incoming data is in D1 icon region, DCP_LB_icon2_allow_stutter is set. This field in this double bufferred register is not double buffered.

Secondary display hardware icon control.

D2ICON_SURFACE_ADDRESS - RW - 32 bits - [GpuF0MMReg:0x6C48]			
Field Name	**Bits**	**Default**	**Description**

D2ICON_SURFACE_ADDRESS	31:0	0x0	Secondary display hardware icon surface base address in byte. It is 4K byte aligned. NOTE: Bits 0:11 of this field are hardwired to ZERO.

Secondary display hardware icon surface base address.

D2ICON_SIZE - RW - 32 bits - [GpuF0MMReg:0x6C50]			
Field Name	**Bits**	**Default**	**Description**
D2ICON_HEIGHT	6:0	0x0	Secondary display hardware icon height minus 1.
D2ICON_WIDTH	22:16	0x0	Secondary display hardware icon width minus 1.

Secondary display hardware icon size.

D2ICON_START_POSITION - RW - 32 bits - [GpuF0MMReg:0x6C54]			
Field Name	**Bits**	**Default**	**Description**
D2ICON_Y_POSITION	12:0	0x0	Secondary display hardware icon Y start coordinate related to the desktop coordinates. Note: Icon can not be off the top and off the left edge of the display surface. But can be off the bottom and off the right edge of the display.
D2ICON_X_POSITION	28:16	0x0	Secondary display hardware icon X start coordinate relative to the desktop coordinates. Note: Icon can not be off the top and off the left edge of the display surface. But can be off the bottom and off the right edge of the display.

Secondary display hardware icon position

D2ICON_COLOR1 - RW - 32 bits - [GpuF0MMReg:0x6C58]			
Field Name	**Bits**	**Default**	**Description**
D2ICON_COLOR1_BLUE	7:0	0x0	Secondary display hardware icon blue component of color 1.
D2ICON_COLOR1_GREEN	15:8	0x0	Secondary display hardware icon green component of color 1.
D2ICON_COLOR1_RED	23:16	0x0	Secondary display hardware icon red component of color 1.

Secondary display hardware icon color 1.

D2ICON_COLOR2 - RW - 32 bits - [GpuF0MMReg:0x6C5C]			
Field Name	**Bits**	**Default**	**Description**
D2ICON_COLOR2_BLUE	7:0	0x0	Secondary display hardware icon blue component of color 2.
D2ICON_COLOR2_GREEN	15:8	0x0	Secondary display hardware icon green component of color 2.

D2ICON_COLOR2_RED	23:16	0x0	Secondary display hardware icon red component of color 2.

Secondary display hardware icon color 2.

D2ICON_UPDATE - RW - 32 bits - [GpuF0MMReg:0x6C60]			
Field Name	**Bits**	**Default**	**Description**
D2ICON_UPDATE_PENDING (R)	0	0x0	Secondary display hardware icon update Pending status. It is set to 1 after a host write to icon double buffer register. It is cleared after double buffering is done. The double buffering occurs when D2ICON_UPDATE_PENDING = 1 and D2ICON_UPDATE_LOCK = 0 and V_UPDATE = 1. If CRTC2 is disabled, the registers will be updated instantly. D2IOCN double buffer registers include : D2ICON_ENABLE D2ICON_2X_MAGNIFY D2ICON_SURFACE_ADDRESS D2ICON_HEIGHT D2ICON_WIDTH D2ICON_Y_POSITION D2ICON_X_POSITION 0=No update pending 1=Update pending
D2ICON_UPDATE_TAKEN (R)	1	0x0	Secondary display hardware icon update Taken status. It is set to 1 when double buffering occurs and cleared when V_UPDATE = 0
D2ICON_UPDATE_LOCK	16	0x0	Secondary display hardware icon update lock control. 0=Unlocked 1=Locked
D2ICON_DISABLE_MULTIPLE_UPDATE	24	0x0	0=D2ICON registers can be updated multiple times in one V_UPDATE period 1=D2ICON registers can only be updated once in one V_UPDATE period

Secondary display hardware icon update control

2.7.22 Secondary Display Multi-VPU Control Registers

D2CRTC_MVP_STATUS - RW - 32 bits - [GpuF0MMReg:0x685C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_FLIP_NOW_OCCURRED (R)	0	0x0	Reports whether flip_now has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D2CRTC_FLIP_NOW_CLEAR (W)	16	0x0	Clears the sticky bit D2CRTC_FLIP_NOW_OCCURRED when written with '1'

Reports status for MVP flipping in CRTC2

D2_MVP_AFR_FLIP_MODE - RW - 32 bits - [GpuF0MMReg:0x65E8]			
Field Name	**Bits**	**Default**	**Description**
D2_MVP_AFR_FLIP_MODE	1:0	0x0	10 - real flip; 11 - dummy flip

S/W writes to this register in AFR mode for display 2 page flip

D2_MVP_AFR_FLIP_FIFO_CNTL - RW - 32 bits - [GpuF0MMReg:0x65EC]			
Field Name	**Bits**	**Default**	**Description**
D2_MVP_AFR_FLIP_FIFO_NUM_ENTRIES (R)	3:0	0x0	number of valid entries in the AFR flip FIFO
D2_MVP_AFR_FLIP_FIFO_RESET	4	0x0	reset the AFR flip FIFO
D2_MVP_AFR_FLIP_FIFO_RESET_FLAG (R)	8	0x0	sticky bit of the AFR flip fifo reset status
D2_MVP_AFR_FLIP_FIFO_RESET_ACK	12	0x0	clear the DC_LB_MVP_AFR_FLIP_RESET_FLAG register bit

This register controls AFR Flip FIFO in display 2

D2_MVP_FLIP_LINE_NUM_INSERT - RW - 32 bits - [GpuF0MMReg:0x65F0]			
Field Name	**Bits**	**Default**	**Description**
D2_MVP_FLIP_LINE_NUM_INSERT_MODE	1:0	0x2	00 - no insertion, 0 is appended; 01 - debug: insert D2_MVP_FLIP_LINE_NUM_INSERT regiser value; 10 - normal Hsync mode, insert the sum of LB line number + DC_LB_MVP_FLIP_LINE_NUM_OFFSET
D2_MVP_FLIP_LINE_NUM_INSERT	21:8	0x0	used for debug purpose, this is what will be the line number carried to downstream GPUs if D2_MVP_FLIP_LINE_NUM_INSERT_EN is set
D2_MVP_FLIP_LINE_NUM_OFFSET	29:24	0x0	used in normal HSYNC flipping operation. this is the number added to the current LB (desktop) line number for carrying to the downstream GPUs
D2_MVP_FLIP_AUTO_ENABLE	30	0x0	Enabling automatic AFR/SFR flipping for display 2

This register controls line number insertion for the Hsync flipping mode in display 2

2.7.23 Display Look Up Table Control Registers

DC_LUT_RW_SELECT - RW - 32 bits - [GpuF0MMReg:0x6480]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_RW_SELECT	0	0x0	LUT host Read/write selection. 0=Host reads/writes to the LUT access the lower half of the LUT 1=Host reads/writes to the LUT access the upper half of the LUT

LUT host Read/write selection.

DC_LUT_RW_MODE - RW - 32 bits - [GpuF0MMReg:0x6484]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_RW_MODE	0	0x0	LUT host read/write mode. 0=Host reads/writes to the LUT in 256-entry table mode 1=Host reads/writes to the LUT in piece wise linear (PWL) mode

LUT host read/write mode.

DC_LUT_RW_INDEX - RW - 32 bits - [GpuF0MMReg:0x6488]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_RW_INDEX	7:0	0x0	LUT index for host read/write. In 256-entry table mode: LUT_ADDR[6:0] = INDEX[7:1]. INDEX[0] is used to select LUT lower or upper 10 bits. In piece wise linear (PWL) mode: LUT_ADDR[6:0] = INDEX[6:0]. INDEX[7] is not used

LUT index for host read/write.

DC_LUT_SEQ_COLOR - RW - 32 bits - [GpuF0MMReg:0x648C]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_SEQ_COLOR	15:0	0x0	Sequential 10-bit R,G,B host read/write for LUT 256-entry table mode. After reset or writing DC_LUT_RW_INDEX register, first DC_LUT_SEQ_COLOR access is for red component, the second one is for green component and the third one is for blue component. Always access this register three times for one LUT entry in LUT 256-entry table mode. The LUT index is increased by 1 when LUT blue data is accessed. This allow you to access the next LUT entry without programming DC_LUT_RW_INDEX again. NOTE: Bits 0:5 of this field are hardwired to ZERO.

Sequential 10-bit R,G,B host read/write for LUT 256-entry table mode.

DC_LUT_PWL_DATA - RW - 32 bits - [GpuF0MMReg:0x6490]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_BASE	15:0	0x0	Linear interpolation of base value for host read/write. NOTE: Bits 0:5 of this field are hardwired to ZERO.
DC_LUT_DELTA	31:16	0x0	Linear interpolation of delta value for host read/write. The LUT index is increased by 1 when register DC_LUT_PWL_DATA is accessed. NOTE: Bits 0:5 of this field are hardwired to ZERO.

Linear interpolation of base and delta host read/write for LUT PWL mode

DC_LUT_30_COLOR - RW - 32 bits - [GpuF0MMReg:0x6494]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_COLOR_10_BLUE	9:0	0x0	10-bit blue value for host read/write. The LUT index is increased by 1 when register DC_LUT_30_COLOR is accessed.
DC_LUT_COLOR_10_GREEN	19:10	0x0	10-bit green value for host read/write.
DC_LUT_COLOR_10_RED	29:20	0x0	10-bit red value for host read/write.

Host read/write LUT R,G,B value for LUT 256-entry table mode

DC_LUT_READ_PIPE_SELECT - RW - 32 bits - [GpuF0MMReg:0x6498]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_READ_PIPE_SELECT	0	0x0	LUT pipe selection for host read. 0=Host read select pipe 0 1=Host read select pipe 1

LUT pipe selection for host read.

DC_LUT_WRITE_EN_MASK - RW - 32 bits - [GpuF0MMReg:0x649C]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_WRITE_EN_MASK	5:0	0x3f	Look-up table macro write enable mask for host write. For each bit 0 - host write disable 1 - host write enable Bit[0] - For pipe 1, B macro Bit[1] - For pipe 1, G macro Bit[2] - For pipe 1, R macro Bit[3] - For pipe 0, B macro Bit[4] - For pipe 0, G macro Bit[5] - For pipe 0, R macro

Look-up table macro write enable mask for host write.

DC_LUT_AUTOFILL - RW - 32 bits - [GpuF0MMReg:0x64A0]			
Field Name	**Bits**	**Default**	**Description**
DC_LUT_AUTOFILL (W)	0	0x0	Enable LUT autofill when 1 is written into this field 0=No effect 1=Start LUT autofill
DC_LUT_AUTOFILL_DONE (R)	1	0x0	LUT autofill is done 0=LUT autofill is not completed 1=LUT autofill is done

LUT autofill control

2.7.24 Display Controller Look Up Table A Registers

DC_LUTA_CONTROL - RW - 32 bits - [GpuF0MMReg:0x64C0]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_INC_B	3:0	0x0	Exponent of Power-of-two of blue data increment of LUTA palette. If INC = 0, LUT 256-entry table mode is enabled. LUT_INDEX = PIX_DATA[7:0]. Output = LUT_DATA[LUT_INDEX]. If INC > 0, LUT PWL mode is enabled with 128 entries of base and delta values. LUT_INDEX = PIX_DATA[INC+6:INC]. Mult = PIX_DATA[INC-1:0]. Base = LUT_BASE[LUT_INDEX]. Delta = LUT_DELTA[LUT_INDEX]. Output = Base + (Mult * Delta) / increment 0=Blue data increment = N/A 1=Blue data increment = 2 2=Blue data increment = 4 3=Blue data increment = 8 4=Blue data increment = 16 5=Blue data increment = 32 6=Blue data increment = 64 7=Blue data increment = 128 8=Blue data increment = 256 9=Blue data increment = 512
DC_LUTA_DATA_B_SIGNED_EN	4	0x0	Frame buffer blue data signed enable for look-up table A. 0=Blue data is unsigned 1=Blue data is signed
DC_LUTA_DATA_B_FLOAT_POINT_EN	5	0x0	Frame buffer blue data float point enable for look-up table A. 0=Blue data is fix point 1=Blue data is float point

DC_LUTA_INC_G	11:8	0x0	Exponent of Power-of-two of green data increment of LUTA palette. If INC = 0, LUT 256-entry table mode is enabled. LUT_INDEX = PIX_DATA[7:0]. Output = LUT_DATA[LUT_INDEX]. If INC > 0, LUT PWL mode is enabled with 128 entries of base and delta values. LUT_INDEX = PIX_DATA[INC+6:INC]. Mult = PIX_DATA[INC-1:0]. Base = LUT_BASE[LUT_INDEX]. Delta = LUT_DELTA[LUT_INDEX]. Output = Base + (Mult * Delta) / increment 0=Green data increment = N/A 1=Green data increment = 2 2=Green data increment = 4 3=Green data increment = 8 4=Green data increment = 16 5=Green data increment = 32 6=Green data increment = 64 7=Green data increment = 128 8=Green data increment = 256 9=Green data increment = 512
DC_LUTA_DATA_G_SIGNED_EN	12	0x0	Frame buffer green data signed enable for look-up table A. 0=Green data is unsigned 1=Green data is signed
DC_LUTA_DATA_G_FLOAT_POINT_EN	13	0x0	Frame buffer green data float point enable for look-up table A. 0=Green data is fix point 1=Green data is float point
DC_LUTA_INC_R	19:16	0x0	Exponent of Power-of-two of red data increment of LUTA palette. If INC = 0, LUT 256-entry table mode is enabled. LUT_INDEX = PIX_DATA[7:0]. Output = LUT_DATA[LUT_INDEX]. If INC > 0, LUT PWL mode is enabled with 128 entries of base and delta values. LUT_INDEX = PIX_DATA[INC+6:INC]. Mult = PIX_DATA[INC-1:0]. Base = LUT_BASE[LUT_INDEX]. Delta = LUT_DELTA[LUT_INDEX]. Output = Base + (Mult * Delta) / increment 0=Red data increment = N/A 1=Red data increment = 2 2=Red data increment = 4 3=Red data increment = 8 4=Red data increment = 16 5=Red data increment = 32 6=Red data increment = 64 7=Red data increment = 128 8=Red data increment = 256 9=Red data increment = 512
DC_LUTA_DATA_R_SIGNED_EN	20	0x0	Frame buffer red data signed enable for look-up table A. 0=Red data is unsigned 1=Red data is signed
DC_LUTA_DATA_R_FLOAT_POINT_EN	21	0x0	Frame buffer red data float point enable for look-up table A. 0=Red data is fix point 1=Red data is float point

LUTA mode control

DC_LUTA_BLACK_OFFSET_BLUE - RW - 32 bits - [GpuF0MMReg:0x64C4]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_BLACK_OFFSET_BLUE	15:0	0x0	Black value offset of blue component for LUTA.

Black value offset of blue component for LUTA.

DC_LUTA_BLACK_OFFSET_GREEN - RW - 32 bits - [GpuF0MMReg:0x64C8]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_BLACK_OFFSET_GREEN	15:0	0x0	Black value offset of green component for LUTA.

Black value offset of green component for LUTA.

DC_LUTA_BLACK_OFFSET_RED - RW - 32 bits - [GpuF0MMReg:0x64CC]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_BLACK_OFFSET_RED	15:0	0x0	Black value offset of red component for LUTA.

Black value offset of red component for LUTA.

DC_LUTA_WHITE_OFFSET_BLUE - RW - 32 bits - [GpuF0MMReg:0x64D0]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_WHITE_OFFSET_BLUE	15:0	0xffff	White value offset of blue component for LUTA

White value offset of blue component for LUTA.

DC_LUTA_WHITE_OFFSET_GREEN - RW - 32 bits - [GpuF0MMReg:0x64D4]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_WHITE_OFFSET_GREEN	15:0	0xffff	White value offset of green component for LUTA

White value offset of green component for LUTA

DC_LUTA_WHITE_OFFSET_RED - RW - 32 bits - [GpuF0MMReg:0x64D8]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTA_WHITE_OFFSET_RED	15:0	0xffff	White value offset of red component for LUTA

White value offset of red component for LUTA

2.7.25 Display Controller Look Up Table B Registers

DC_LUTB_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6CC0]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_INC_B	3:0	0x0	Exponent of Power-of-two of blue data increment of LUTB palette. If INC = 0, LUT 256-entry table mode is enabled. LUT_INDEX = PIX_DATA[7:0]. Output = LUT_DATA[LUT_INDEX]. If INC > 0, LUT PWL mode is enabled with 128 entries of base and delta values. LUT_INDEX = PIX_DATA[INC+6:INC]. Mult = PIX_DATA[INC-1:0]. Base = LUT_BASE[LUT_INDEX]. Delta = LUT_DELTA[LUT_INDEX]. Output = Base + (Mult * Delta) / increment 0=Blue data increment = N/A 1=Blue data increment = 2 2=Blue data increment = 4 3=Blue data increment = 8 4=Blue data increment = 16 5=Blue data increment = 32 6=Blue data increment = 64 7=Blue data increment = 128 8=Blue data increment = 256 9=Blue data increment = 512
DC_LUTB_DATA_B_SIGNED_EN	4	0x0	Frame buffer blue data signed enable for look-up table A. 0=Blue data is unsigned 1=Blue data is signed
DC_LUTB_DATA_B_FLOAT_POINT_EN	5	0x0	Frame buffer blue data float point enable for look-up table A. 0=Blue data is fix point 1=Blue data is float point

DC_LUTB_INC_G	11:8	0x0	Exponent of Power-of-two of green data increment of LUTB palette. If INC = 0, LUT 256-entry table mode is enabled. LUT_INDEX = PIX_DATA[7:0]. Output = LUT_DATA[LUT_INDEX]. If INC > 0, LUT PWL mode is enabled with 128 entries of base and delta values. LUT_INDEX = PIX_DATA[INC+6:INC]. Mult = PIX_DATA[INC-1:0]. Base = LUT_BASE[LUT_INDEX]. Delta = LUT_DELTA[LUT_INDEX]. Output = Base + (Mult * Delta) / increment 0=Green data increment = N/A 1=Green data increment = 2 2=Green data increment = 4 3=Green data increment = 8 4=Green data increment = 16 5=Green data increment = 32 6=Green data increment = 64 7=Green data increment = 128 8=Green data increment = 256 9=Green data increment = 512
DC_LUTB_DATA_G_SIGNED_EN	12	0x0	Frame buffer green data signed enable for look-up table A. 0=Green data is unsigned 1=Green data is signed
DC_LUTB_DATA_G_FLOAT_POINT_EN	13	0x0	Frame buffer green data float point enable for look-up table A. 0=Green data is fix point 1=Green data is float point
DC_LUTB_INC_R	19:16	0x0	Exponent of Power-of-two of red data increment of LUTB palette. If INC = 0, LUT 256-entry table mode is enabled. LUT_INDEX = PIX_DATA[7:0]. Output = LUT_DATA[LUT_INDEX]. If INC > 0, LUT PWL mode is enabled with 128 entries of base and delta values. LUT_INDEX = PIX_DATA[INC+6:INC]. Mult = PIX_DATA[INC-1:0]. Base = LUT_BASE[LUT_INDEX]. Delta = LUT_DELTA[LUT_INDEX]. Output = Base + (Mult * Delta) / increment 0=Red data increment = N/A 1=Red data increment = 2 2=Red data increment = 4 3=Red data increment = 8 4=Red data increment = 16 5=Red data increment = 32 6=Red data increment = 64 7=Red data increment = 128 8=Red data increment = 256 9=Red data increment = 512
DC_LUTB_DATA_R_SIGNED_EN	20	0x0	Frame buffer red data signed enable for look-up table A. 0=Red data is unsigned 1=Red data is signed
DC_LUTB_DATA_R_FLOAT_POINT_EN	21	0x0	Frame buffer red data float point enable for look-up table A. 0=Red data is fix point 1=Red data is float point

LUTB mode control

DC_LUTB_BLACK_OFFSET_BLUE - RW - 32 bits - [GpuF0MMReg:0x6CC4]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_BLACK_OFFSET_BLUE	15:0	0x0	Black value offset of blue component for LUTB.

Black value offset of blue component for LUTB.

DC_LUTB_BLACK_OFFSET_GREEN - RW - 32 bits - [GpuF0MMReg:0x6CC8]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_BLACK_OFFSET_GREEN	15:0	0x0	Black value offset of green component for LUTB.

Black value offset of green component for LUTB.

DC_LUTB_BLACK_OFFSET_RED - RW - 32 bits - [GpuF0MMReg:0x6CCC]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_BLACK_OFFSET_RED	15:0	0x0	Black value offset of red component for LUTB.

Black value offset of red component for LUTB.

DC_LUTB_WHITE_OFFSET_BLUE - RW - 32 bits - [GpuF0MMReg:0x6CD0]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_WHITE_OFFSET_BLUE	15:0	0xffff	White value offset of blue component for LUTB

White value offset of blue component for LUTB.

DC_LUTB_WHITE_OFFSET_GREEN - RW - 32 bits - [GpuF0MMReg:0x6CD4]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_WHITE_OFFSET_GREEN	15:0	0xffff	White value offset of green component for LUTB

White value offset of green component for LUTB

DC_LUTB_WHITE_OFFSET_RED - RW - 32 bits - [GpuF0MMReg:0x6CD8]			
Field Name	**Bits**	**Default**	**Description**
DC_LUTB_WHITE_OFFSET_RED	15:0	0xffff	White value offset of red component for LUTB

White value offset of red component for LUTB

2.7.26 Display Controller CRC Registers

DCP_CRC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6C80]			
Field Name	**Bits**	**Default**	**Description**
DCP_CRC_ENABLE	0	0x0	Enable DCP CRC.
DCP_CRC_DISPLAY_SEL	1	0x0	Select display number for DCP CRC. 0= from display 1 1= from display 2
DCP_CRC_SOURCE_SEL	4:2	0x0	Select data source for DCP CRC. 0=DCP to LB pixel data 1=Lower 32 bits of graphics input data to DCP from DMIF 2=Upper 32 bits of graphics input data to DCP from DMIF 3=Overlay input data to DCP from DMIF 4=DCP to LB control signals TAG[2:0] and end of chunk

DCP CRC control

DCP_CRC_MASK - RW - 32 bits - [GpuF0MMReg:0x6C84]			
Field Name	**Bits**	**Default**	**Description**
DCP_CRC_MASK	31:0	0x0	Mask bits to apply to DCP CRC function. Allows CRC of only specific color and/or specific bits if wanted. Igore those bits with mask bits to be 0

Mask bits to apply to DCP CRC function.

DCP_CRC_P0_CURRENT - RW - 32 bits - [GpuF0MMReg:0x6C88]			
Field Name	**Bits**	**Default**	**Description**
DCP_CRC_P0_CURRENT (R)	31:0	0x0	Current value of CRC for current frame pipe 0.

Current value of CRC for current frame pipe 0.

DCP_CRC_P1_CURRENT - RW - 32 bits - [GpuF0MMReg:0x6C8C]			
Field Name	**Bits**	**Default**	**Description**
DCP_CRC_P1_CURRENT (R)	31:0	0x0	Current value of CRC for current frame pipe 1.

Current value of CRC for current frame pipe 1.

DCP_CRC_P0_LAST - RW - 32 bits - [GpuF0MMReg:0x6C90]			
Field Name	**Bits**	**Default**	**Description**
DCP_CRC_P0_LAST (R)	31:0	0x0	Final value of CRC for previous frame pipe 0.

Final value of CRC for previous frame pipe 0.

DCP_CRC_P1_LAST - RW - 32 bits - [GpuF0MMReg:0x6C94]			
Field Name	**Bits**	**Default**	**Description**
DCP_CRC_P1_LAST (R) Final value of CRC for previous frame pipe 1.	31:0	0x0	Final value of CRC for previous frame pipe 1.

2.7.27 Display/Memory Interface Control and Status Registers

DCP_TILING_CONFIG - RW - 32 bits - [GpuF0MMReg:0x6CA0]			
Field Name	**Bits**	**Default**	**Description**
PIPE_TILING	3:1	0x3	This specifies the number of logical rendering pipes to use in the tiling pattern. Typically this should match the number of memory channels. 0=CONFIG_1_PIPE: 1 logical rendering pipe 1=CONFIG_2_PIPE: 2 logical rendering pipes 2=CONFIG_4_PIPE: 4 logical rendering pipes 3=CONFIG_8_PIPE: 8 logical rendering pipes
BANK_TILING	5:4	0x0	This specifies the number of logical banks to use in the tiling pattern. Typically this should match the number of physical banks in the DRAMs, though it can be smaller (e.g. for DRAMs that have more banks than the tiling supports) or larger (e.g. if rank selection is treated as a logical bank bit). 0=CONFIG_4_BANK: 4 logical DRAM banks 1=CONFIG_8_BANK: 8 logical DRAM banks
GROUP_SIZE	7:6	0x0	This specifies the memory interleave group size. All surfaces must be aligned to start at a group interleave boundary. Sequential reads or writes in device address space access this many bytes from each memory channel in turn. Therefore this value determines the maximum DRAM burst size for sequential accesses. 0=CONFIG_256B_GROUP: 256B memory interleve groups 1=CONFIG_512B_GROUP: 512B memory interleve groups
ROW_TILING	10:8	0x2	This specifies a DRAM row size for use in tiling, within a given bank of a given memory channel. This may be smaller than the actual DRAM row size, but should not be larger. The tiling pattern switches banks at these row boundaries and clients may also use this field to determine whether two accesses might be in the same row. These strategies are not effective for scattered virtual memorty mappings. 0=CONFIG_1KB_ROW: Treat 1KB as DRAM row boundary 1=CONFIG_2KB_ROW: Treat 2KB as DRAM row boundary 2=CONFIG_4KB_ROW: Treat 4KB as DRAM row boundary 3=CONFIG_8KB_ROW: Treat 8KB as DRAM row boundary

BANK_SWAPS	13:11	0x1	When performing display reads, this specifies the maximum number of bytes accessed per memory channel within each bank before switching banks. This affects the DRAM burst length for display accesses. The actual burst length may be less, depending on the row size above and on whether the display access starts in the middle of a bank swap sequence. This also ensures that crossing a DRAM row boundary switches banks, provided that the virtual page mapping is aligned properly. 0=CONFIG_128B_SWAPS: Perform bank swap after 128B 1=CONFIG_256B_SWAPS: Perform bank swap after 256B 2=CONFIG_512B_SWAPS: Perform bank swap after 512B 3=CONFIG_1KB_SWAPS: Perform bank swap after 1KB
SAMPLE_SPLIT	15:14	0x3	This controls the number of bytes per tile that may be used to store multiple samples of fragments. If multi-sample data requires more bytes than this per tile, it is split into multiple slices. 0=CONFIG_1KB_SPLIT: Split multi-sample tiles over 1KB 1=CONFIG_2KB_SPLIT: Split multi-sample tiles over 2KB 2=CONFIG_4KB_SPLIT: Split multi-sample tiles over 4KB 3=CONFIG_8KB_SPLIT: Split multi-sample tiles over 8KB

This register is a copy of PDMA_TILING_CONFIG and may ONLY be written when the chip is idle, and MUST be matched by a write to GB_TILING_CONFIG, PDMA_TILING_CONFIG and all copies of *TILING_CONFIG. It affects the 2D tiling modes, so writing to it invalidates all 2D tiled surfaces.

DCP_MULTI_CHIP_CNTL - RW - 32 bits - [GpuF0MMReg:0x6CA4]			
Field Name	**Bits**	**Default**	**Description**
LOG2_NUM_CHIPS	2:0	0x0	Log2 of the number of chips in the multi-chip configuration.
MULTI_CHIP_TILE_SIZE	4:3	0x0	Size of the tile per chip within each super-tile. 0=16 x 16 pixel tile per chip. 1=32 x 32 pixel tile per chip. 2=64 x 64 pixel tile per chip. 3=128x128 pixel tile per chip.

Should be programmed with the same value as PA_SC_MULTI_CHIP_CNTL. Controls the Screen Divisioning for Multi-Chip Configurations

DMIF_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6CB0]			
Field Name	**Bits**	**Default**	**Description**
DMIF_BUFF_SIZE	1:0	0x0	DMIF memory size. 0x0 - full memory size, 384x256bits. 0x1 - 2/3 memory size. 0x2 - 1/3 memory size. 0x3 - reserved

DMIF_D1_REQ_BURST_SIZE	10:8	0x2	DMIF request burst size for display 1. 0x0 - 1 request. 0x1 - 2 requests. 0x2 - 4 requests. 0x3 - 8 requests. 0x4 - 16 requests.
DMIF_D2_REQ_BURST_SIZE	18:16	0x2	DMIF request burst size for display 2. 0x0 - 1 request. 0x1 - 2 requests. 0x2 - 4 requests. 0x3 - 8 requests. 0x4 - 16 requests.

DMIF control register

DMIF_STATUS - RW - 32 bits - [GpuF0MMReg:0x6CB4]			
Field Name	**Bits**	**Default**	**Description**
DMIF_MC_SEND_ON_IDLE (R)	0	0x0	This register bit is set to 1 if MH returns data to DMIF when there is no pending request. It is sticky bit. Once this bit is set to high, it will stay high until it is cleared by writing 1 to register DMIF_CLEAR_MH_DATA_ON_IDLE 0=MC does not send data to DMIF when there is no data request pending 1=MH sends data to DMIF when there is no data pending request.
DMIF_CLEAR_MC_SEND_ON_IDLE (W)	1	0x0	This register bit is used to clear register DMIF_MH_SEND_ON_IDLE 0=No effect 1=Clear register bit DMIF_MH_SEND_ON_IDLE
DMIF_MC_LATENCY_COUNTER_ENABLE	8	0x0	0=Disable MC latency counter 1=Enable MC latency counter

This is a debug register. DMIF status.

2.7.28 MCIF Control Registers

MCIF_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6CB8]			
Field Name	**Bits**	**Default**	**Description**
MCIF_BUFF_SIZE	1:0	0x0	MCIF memory size. 0x0 - full memory size, 16x143bits. 0x1 - 3/4 memory size. 0x2 - 1/2 memory size. 0x3 - 1/4 memory size.
ADDRESS_TRANSLATION_ENABLE	4	0x0	Enables address translation for vga, cursor and icon memory controller requests 0=disable 1=enable
PRIVILEGED_ACCESS_ENABLE	8	0x0	Enables privileged page access for vga, cursor and icon memory controller requests 0=disable 1=enable

LOW_READ_URG_LEVEL	23:16	0x0	This is the urgency level for vga, cursor, icon and vip reads when they are all in low priority
MC_CLEAN_DEASSERT_LATENCY	29:24	0x10	This is the number of cycles mcif will wait after a write is transfered to the memory controller and before looking at the clean signal from the memory controller

MCIF control register

2.7.29 Display Controller to Line Buffer Control Registers

DCP_LB_DATA_GAP_BETWEEN_CHUNK - RW - 32 bits - [GpuF0MMReg:0x6CBC]			
Field Name	**Bits**	**Default**	**Description**
DCP_LB_GAP_BETWEEN_CHUNK_20BPP	3:0	0x5	This register is used to control gap between data chunks sent from DCP to LB when the next LB data chunk is in 20bpp mode. The gap between current chunk and next chunk will be register value plus 1. The default value is 5. If any display has 32bpp digital output enabled, this valus should be set to 6.
DCP_LB_GAP_BETWEEN_CHUNK_30BPP	7:4	0x1	This register is used to control gap between data chunks sent from DCP to LB when the next LB data chunk is in 30bpp mode. The gap between current chunk and next chunk will be register value plus 1. The default value is 1. If any display has 32bpp digital output enabled, this valus should be set to 4

DCP LB chunk gap control

2.7.30 Multi VPU Control Registers

DC_MVP_LB_CONTROL - RW - 32 bits - [GpuF0MMReg:0x65F4]			
Field Name	**Bits**	**Default**	**Description**
D1_MVP_SWAP_LOCK_IN_MODE	1:0	0x1	01 - force input to 1, used for master GPU; 10 - use swap_lock_in, used for slave GPU or middle GPU; 01 is the default
D2_MVP_SWAP_LOCK_IN_MODE	5:4	0x2	01 - force input to 1, used for master GPU; 10 - use swap_lock_in, used for slave GPU or middle GPU; 10 is the default
DC_MVP_SWAP_LOCK_OUT_SEL	8	0x0	0 - use D1 swap out output, 1 - use D2 swap out output; default is D1 swap out
DC_MVP_SWAP_LOCK_OUT_FORCE_ONE	12	0x0	Force Swap_lock to be one
DC_MVP_SWAP_LOCK_OUT_FORCE_ZERO	16	0x0	Force Swap_lock to be zero
DC_MVP_D1_DFQ_EN	18	0x0	Enable DFQ in multi-GPU mode to select update_pending from DFQ engine
DC_MVP_D2_DFQ_EN	19	0x0	Enable DFQ in multi-GPU mode to select update_pending from DFQ engine
DC_MVP_D1_SWAP_LOCK_STATUS (R)	20	0x0	D1 swap_lock status
DC_MVP_D2_SWAP_LOCK_STATUS (R)	24	0x0	D2 swap_lock status
DC_MVP_SWAP_LOCK_IN_CAP (R)	28	0x0	Capture swap_lock_in, used in diagnostic mode
DC_MVP_SPARE_FLOPS (R)	31	0x0	USED for keeping spare flops (ECO)

DC MVP LB control register

2.8 CRTC Control Registers

DC_CRTC_MASTER_EN - RW - 32 bits - [GpuF0MMReg:0x60F8]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_MASTER_EN *(mirror of D1CRTC_CONTROL:D1CRTC_MASTER_EN)*	0	0x0	Mirror of D1CRTC_MASTER_EN field in D1CRTC_CONTROL register
D2CRTC_MASTER_EN *(mirror of D2CRTC_CONTROL:D2CRTC_MASTER_EN)*	1	0x0	Mirror of D2CRTC_MASTER_EN field in D1CRTC_CONTROL register

Contains mirror of DxCRTC_MASTER_EN register field in DxCRTC_CONTROL registers

DC_CRTC_TV_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60FC]			
Field Name	**Bits**	**Default**	**Description**
CRTC_TV_DATA_SOURCE	0	0x0	Determines source of pixel data and control signals to TV encoder 0 = CRTC1 1 = CRTC2

Controls source of pixel data and control signals to TV encoder

2.8.1 Primary Display CRTC Control Registers

D1CRTC_H_TOTAL - RW - 32 bits - [GpuF0MMReg:0x6000]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_H_TOTAL	12:0	0x0	Horizontal total minus one. Sum of display width, overscan left and right, front and back porch and H sync width. E.g. for 800 pixels set to 799 = 0x31F Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Defines horizontal dimension of the display timing for CRTC1

D1CRTC_H_BLANK_START_END - RW - 32 bits - [GpuF0MMReg:0x6004]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_H_BLANK_START	12:0	0x0	Start of the horizontal blank. The location of the first pixel of horizontal blank, relative to pixel zero. If right overscan border, then blank starts after border ends. Double-buffered with D1MODE_MASTER_UPDATE_LOCK
D1CRTC_H_BLANK_END	28:16	0x0	End of the horizontal blank. The location of the next pixel after the last pixel of horizontal blank, relative to pixel zero. Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Defines horizontal blank region of the display timing for CRTC1

D1CRTC_H_SYNC_A - RW - 32 bits - [GpuF0MMReg:0x6008]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_H_SYNC_A_START	12:0	0x0	First pixel of horizontal sync A. In normal cases, it is set to 0. It is only set to non-zero value when we want to test the higher bits of the H counter. This register should be ignored and set to 0x0 in VGA timing mode. Hardware does not support odd number value for this register.
D1CRTC_H_SYNC_A_END	28:16	0x0	Horizontal sync A end. Determines position of the next pixel after last pixel of horizontal sync A. The last pixel of horizontal sync A is D1CRTC_H_SYNC_A_END - 1. The first pixel of horizontal sync A is pixel 0. It should be programmed to a value one greater than the actual last pixel of horizontal sync A. Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Defines horizontal sync A position for CRTC1

D1CRTC_H_SYNC_A_CNTL - RW - 32 bits - [GpuF0MMReg:0x600C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_H_SYNC_A_POL	0	0x0	Polarity of H SYNC A 0 = active high 1 = active low Double-buffered with D1MODE_MASTER_UPDATE_LOCK
D1CRTC_COMP_SYNC_A_EN	16	0x0	Enables composite H sync A 0 = disabled 1 = enabled
D1CRTC_H_SYNC_A_CUTOFF	17	0x0	Cutoff H sync A at end of H BLANK when end of H sync A is beyond H BLANK 0 = cutoff is enabled 1 = cutoff is disabled

Controls the H SYNC A for CRTC1

D1CRTC_H_SYNC_B - RW - 32 bits - [GpuF0MMReg:0x6010]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_H_SYNC_B_START	12:0	0x0	First pixel of horizontal sync B
D1CRTC_H_SYNC_B_END	28:16	0x0	Horizontal sync B end. Determines position of the next pixel after last pixel of horizontal sync B. The last pixel of horizontal sync B is D1CRTC_H_SYNC_B_END - 1. This register value is exclusive. It should be programmed to a value one greater than the actual last pixel of horizontal sync B

Defines the position of horizontal sync B for CRTC1

D1CRTC_H_SYNC_B_CNTL - RW - 32 bits - [GpuF0MMReg:0x6014]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_H_SYNC_B_POL	0	0x0	Polarity of H SYNC B 0 = active high 1 = active low
D1CRTC_COMP_SYNC_B_EN	16	0x0	Enables composite H SYNC B 0 = disabled 1 = enabled
D1CRTC_H_SYNC_B_CUTOFF	17	0x0	Cutoff horizontal sync B at end of horizontal blank region when end of H SYNC B is beyond horizontal blank 0 = cutoff is enabled 1 = cutoff is disabled

Controls horizontal sync B for CRTC1

D1CRTC_V_TOTAL - RW - 32 bits - [GpuF0MMReg:0x6020]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_TOTAL	12:0	0x0	Vertical total minus one. Sum of vertical active display, top and bottom overscan, front and back porch and vertical sync width. E.g. for 525 lines set to 524 = 0x20C Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Defines the vertical dimension of display timing for CRTC1

D1CRTC_V_BLANK_START_END - RW - 32 bits - [GpuF0MMReg:0x6024]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_BLANK_START	12:0	0x0	Vertical blank start. Determines the position of the first blank line in a frame. Line 0 is the first line of vertical sync A. Double-buffered with D1MODE_MASTER_UPDATE_LOCK
D1CRTC_V_BLANK_END	28:16	0x0	Vertical blank end. Determines the position of the next line after the last line of vertical blank. The last line of vertical blank is D1CRTC_V_BLANK_END - 1. Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Defines the vertical blank region of the display timing for CRTC1

D1CRTC_V_SYNC_A - RW - 32 bits - [GpuF0MMReg:0x6028]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_SYNC_A_START	12:0	0x0	The first line of vertical sync A. In normal cases, it is set to 0. It is set to non-zero value only when trying to test the higher bits of the vertical counter

D1CRTC_V_SYNC_A_END	28:16	0x0	Vertical sync A end. Determines the position of the next line after the last line of vertical sync A. The last line of vertical sync A is D1CRTC_V_SYNC_A_END - 1. The first line of vertical sync A is line 0. This register value is exclusive. It should be programmed to a value one greater than the actual last line of vertical sync A Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Defines the position of vertical sync A for CRTC1

D1CRTC_V_SYNC_A_CNTL - RW - 32 bits - [GpuF0MMReg:0x602C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_SYNC_A_POL	0	0x0	Polarity of V SYNC A 0 = active high 1 = active low Double-buffered with D1MODE_MASTER_UPDATE_LOCK

Controls V SYNC A for CRTC1

D1CRTC_V_SYNC_B - RW - 32 bits - [GpuF0MMReg:0x6030]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_SYNC_B_START	12:0	0x0	Vertical sync B start. Determines the position of the first line of vertical sync B.
D1CRTC_V_SYNC_B_END	28:16	0x0	Vertical sync B end. Determines the position of the next line after the last line of vertical sync B. Last line of vertical sync B is D1CRTC_V_SYNC_B_END - 1. This register value is exclusive. It should be programmed to a value one greater than the actual last line of vertical sync B

Defines the position of vertical sync B for CRTC1

D1CRTC_V_SYNC_B_CNTL - RW - 32 bits - [GpuF0MMReg:0x6034]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_SYNC_B_POL	0	0x0	Controls polarity of vertical sync B 0 = active high 1 = active low

Controls vertical sync B for CRTC1

D1CRTC_TRIGA_CNTL - RW - 32 bits - [GpuF0MMReg:0x6060]			
Field Name	**Bits**	**Default**	**Description**

D1CRTC_TRIGA_SOURCE_SELECT	3:0	0x0	Select source of input signals for external trigger A 0 = logic 0 1 = VSYNCA from another CRTC of the chip 2 = HSYNCA from another CRTC of the chip 3 = VSYNCB from another CRTC of the chip 4 = HSYNCB from another CRTC of the chip 5 = GENERICA pin 6 = GENERICB pin 7 = VSYNCA pin 8 = HSYNCA pin 9 = VSYNCB pin 10 = HSYNCB pin 11 = HPD1 pin 12 = HPD2 pin 13 = DVALID pin 14 = PSYNC pin 15 = Video capture complete signal from VIP
D1CRTC_TRIGA_POLARITY_SELECT	6:4	0x0	Selects source of input signal from polarity of external trigger A 0 = logic 0 1 = interlace polarity from another CRTC of the chip 2 = GENERICA pin 3 = GENERICB pin 4 = HSYNCA pin 5 = HSYNCB pin 6 = video capture polarity input from VIP 7 = DVALID pin
D1CRTC_TRIGA_RESYNC_BYPASS_EN	8	0x0	Bypass the resync logic for the external trigger A signal and its polarity input signal 0 = do not bypass 1 = bypass the resync logic
D1CRTC_TRIGA_INPUT_STATUS (R)	9	0x0	Read back the value of the external trigger A input signal after the mux
D1CRTC_TRIGA_POLARITY_STATUS (R)	10	0x0	Reports the value of the external trigger A polarity signal after the mux
D1CRTC_TRIGA_OCCURRED (R)	11	0x0	Reports whether external trigger A has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D1CRTC_TRIGA_RISING_EDGE_DETECT_CNTL	13:12	0x0	Controls the detection of rising edge of the external trigger A signal 00 = do not detect rising edge 01 = always detect rising edge 10 = detect rising edge only when field polarity is low 11 = detect rising edge only when field polarity is high
D1CRTC_TRIGA_FALLING_EDGE_DETECT_CNTL	17:16	0x0	Controls the detection of falling edge of external trigger A signal 00 = do not detect falling edge 01 = always detect falling edge 10 = detect falling edge only when field polarity is low 11 = detect falling edge only when field polarity is high
D1CRTC_TRIGA_FREQUENCY_SELECT	21:20	0x0	Determines the frequency of the external trigger A signal 00 = send every signal 01 = send every 2 signals 10 = reserved 11 = send every 4 signals
D1CRTC_TRIGA_DELAY	28:24	0x0	A programmable PCLK_CRTC1 delay to send external trigger A signal.
D1CRTC_TRIGA_CLEAR (W)	31	0x0	Clears the sticky bit D1CRTC_TRIGA_OCCURRED when written with '1'

Controls for external trigger A signal in CRTC1

D1CRTC_TRIGA_MANUAL_TRIG - RW - 32 bits - [GpuF0MMReg:0x6064]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_TRIGA_MANUAL_TRIG (W)	0	0x0	One shot trigger for external trigger A signal when written with '1'

Manual trigger for external trigger A signal of CRTC1

D1CRTC_TRIGB_CNTL - RW - 32 bits - [GpuF0MMReg:0x6068]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_TRIGB_SOURCE_SELECT	3:0	0x0	Select source of input signals for external trigger B 0 = logic 0 1 = VSYNCA from another CRTC of the chip 2 = HSYNCA from another CRTC of the chip 3 = VSYNCB from another CRTC of the chip 4 = HSYNCB from another CR TC of the chip 5 = GENERICA pin 6 = GENERICB pin 7 = VSYNCA pin 8 = HSYNCA pin 9 = VSYNCB pin 10 = HSYNCB pin 11 = HPD1 pin 12 = HPD2 pin 13 = DVALID pin 14 = PSYNC pin 15 = Video capture complete signal from VIP
D1CRTC_TRIGB_POLARITY_SELECT	6:4	0x0	Selects source of input signal from polarity of external trigger A 0 = logic 0 1 = interlace polarity from another CRTC of the chip 2 = GENERICA pin 3 = GENERICB pin 4 = HSYNCA pin 5 = HSYNCB pin 6 = video capture polarity input from VIP 7 = DVALID pin
D1CRTC_TRIGB_RESYNC_BYPASS_EN	8	0x0	Bypass the resync logic for the external trigger A signal and its polarity input signal 0 = do not bypass 1 = bypass the resync logic
D1CRTC_TRIGB_INPUT_STATUS (R)	9	0x0	Read back the value of the external trigger B input signal after the mux
D1CRTC_TRIGB_POLARITY_STATUS (R)	10	0x0	Reports the value of the external trigger B polarity signal after the mux
D1CRTC_TRIGB_OCCURRED (R)	11	0x0	Reports whether external trigger B has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D1CRTC_TRIGB_RISING_EDGE_DETECT_CNTL	13:12	0x0	Controls the detection of rising edge of the external trigger B signal 00 = do not detect rising edge 01 = always detect rising edge 10 = detect rising edge only when field polarity is low 11 = detect rising edge only when field polarity is high

D1CRTC_TRIGB_FALLING_EDGE_DETECT_CNTL	17:16	0x0	Controls the detection of falling edge of external trigger B signal 00 = do not detect falling edge 01 = always detect falling edge 10 = detect falling edge only when field polarity is low 11 = detect falling edge only when field polarity is high
D1CRTC_TRIGB_FREQUENCY_SELECT	21:20	0x0	Determines the frequency of the external trigger B signal 00 = send every signal 01 = send every 2 signals 10 = reserved 11 = send every 4 signals
D1CRTC_TRIGB_DELAY	28:24	0x0	A programmable delay to send external trigger B signal
D1CRTC_TRIGB_CLEAR (W)	31	0x0	Clears the sticky bit D1CRTC_TRIGB_OCCURRED when written with '1'

Control for external trigger B signal of CRTC1

D1CRTC_TRIGB_MANUAL_TRIG - RW - 32 bits - [GpuF0MMReg:0x606C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_TRIGB_MANUAL_TRIG (W)	0	0x0	One shot trigger for external trigger B signal when written with '1'

Manual trigger for external trigger B signal for CRTC1

D1CRTC_FORCE_COUNT_NOW_CNTL - RW - 32 bits - [GpuF0MMReg:0x6070]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_FORCE_COUNT_NOW_MODE	1:0	0x0	Controls which timing counter is forced 0 = force counter now mode is disabled 1 = force H count now to H_TOTAL only 2 = force H count to H_TOTAL and V count to V_TOTAL in progressive mode and V_TOTAL-1 in interlaced mode 3 = reserved
D1CRTC_FORCE_COUNT_NOW_TRIG_SEL	8	0x0	Selects the trigger signal as force count now trigger 0 = selects CRTC_TRIG_A and CRTC_TRIG_A_POL 1 = selects CRTC_TRIG_B and CRTC_TRIG_B_POL
D1CRTC_FORCE_COUNT_NOW_OCCURRED (R)	16	0x0	Reports the status of force count now, a sticky bit. 0 = CRTC force count now has not occurred 1 = CRTC force count now has occurred
D1CRTC_FORCE_COUNT_NOW_CLEAR (W)	24	0x0	Resets D1CRTC_FORCE_COUNT_NOW_OCCURRED when written with '1'

Controls CRTC1 force count now logic

D1CRTC_FLOW_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6074]			
Field Name	**Bits**	**Default**	**Description**

D1CRTC_FLOW_CONTROL_SOURCE_SELECT	4:0	0x0	Selects the signal used for flow control in CRTC1 0 = logic 0 1 = GENERICA pin 2 = GENERICB pin 3 = HPD1 pin 4 = HPD2 pin 5 = DDC1DATA pin 6 = DDC1CLK pin 7 = DDC2DATA pin 8 = DDC2CLK pin 9 = DVOCLK pin 10 = VHAD(0] pin 11 = VHAD[1] pin 12 = VPHCTL pin 13 = VIPCLK pin 14 = DVALID pin 15 = PSYNC pin 16 = a GPIO pin for dual-GPU, TBD
D1CRTC_FLOW_CONTROL_POLARITY	8	0x0	Controls the polarity of the flow control input signal 0 = keep the signal the same polarity 1 = invert the polartiy of the input signal
D1CRTC_FLOW_CONTROL_GRANULARITY	16	0x0	Controls at which pixel position flow control can start to happen 0 = flow control only start to happen on odd-even pixel boundary 1 = flow control can start at any pixel position
D1CRTC_FLOW_CONTROL_INPUT_STATUS (R)	24	0x0	Reports the value of the flow control input signal 0 = output of source mux of flow control signal is low 1 = output of source mux of flow control signal is high

Controls flow control of CRTC1

D1CRTC_PIXEL_DATA_READBACK - RW - 32 bits - [GpuF0MMReg:0x6078]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_PIXEL_DATA_BLUE_CB (R)	9:0	0x0	B/Cb component sent to DISPOUT
D1CRTC_PIXEL_DATA_GREEN_Y (R)	19:10	0x0	G/Y component sent to DISPOUT
D1CRTC_PIXEL_DATA_RED_CR (R)	29:20	0x0	R/Cr component sent to DISPOUT

Read back of the CRTC1 pixel data sent to DISPOUT. This is a debug register. Intended for use in one shot clocking mode.

D1CRTC_STEREO_FORCE_NEXT_EYE - RW - 32 bits - [GpuF0MMReg:0x607C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_STEREO_FORCE_NEXT_EYE (W)	1:0	0x0	Force next frame eye view - One shot. 00: No force - next eye opposite of current eye 01: Right eye force - force right eye next field/frame 10: Left eye force - force right eye next field/frame 11: Reserved After a force has occured, readback of this register will be 00

Force Next Eye register

D1CRTC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6080]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_MASTER_EN	0	0x0	Enables/Disables CRTC1. H counter is at H_TOTAL and V counter is at first line of blank when CRTC is disabled. 0 = Disabled 1 = Enabled
D1CRTC_SYNC_RESET_SEL	4	0x1	Allows power management to lower CRTC1 enable.
D1CRTC_DISABLE_POINT_CNTL	9:8	0x1	When D1CRTC_MASTER_EN is set to 0, delay the disabling of CRTC1 until certain point within the frame 00 = disable CRTC immediately 01 = delay disable CRTC until the end of the current line 10 = reserved 11 = delay disable CRTC until end of the first line in the vertical blank region
D1CRTC_CURRENT_MASTER_EN_STATE (R)	16	0x0	Read-only field indicates the current status of the timing generator. Can be used to poll for when a delayed disable takes effect. 0 = CRTC is disabled 1 = CRTC is enabled
D1CRTC_DISP_READ_REQUEST_DISABLE	24	0x0	Disables data read request from the display controller. Can be used to stop display reads from system memory but keep display timing generation running. Has no effect if CRTC is disabled. 0 = do not disable data read request 1 = disable data read request
D1CRTC_PREFETCH_EN	28	0x0	Double bufferred. Enable data prefetch for display 1 0 = do not enable prefetch 1 = enable data prefetch
D1CRTC_SOF_PULL_EN	29	0x0	At SOF, LB level can be set a a programmable value (D1MODE_SOF_READ_PT), which is the point LB can make requests to. Between SOF to active line, CRTC needs to pull scaler/LB so that LB can make data requests beyond that programmable point. 0 = do not enable pulling 1 = enable pulling

Controls CRTC1 timing generator and data read request to display1

D1CRTC_BLANK_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6084]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_CURRENT_BLANK_STATE (R)	0	0x0	Read only status indicating current state of display blanking. 0 = screen not blanked 1 = screen is blanked
D1CRTC_BLANK_DATA_EN	8	0x0	Enable for blanking active display area. The active area of display that is forced will use the D1CRTC_BLACK_COLOR value. This field is optionally double buffered with D1CRTC_BLANK_DATA_DOUBLE_BUFFER_EN. 0 = disable blanking 1 = enable blanking

D1CRTC_BLANK_DE_MODE	16	0x0	Determines whether BLANK and DATA_ACTIVE signal keeps toggling when screen is blank 0 = toggles BLANK and DATA_ACTIVE 1 = keep BLANK active and DATA_ACTIVE inactive

Controls forced blanking of active area of display timing. Useful for display mode switches when corrupted image may be generated for a frame or two.

D1CRTC_INTERLACE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6088]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_INTERLACE_ENABLE	0	0x0	Enables interlaced timing 0 = Progressive timing 1 = Interlaced timing
D1CRTC_INTERLACE_FORCE_NEXT_FIELD (W)	17:16	0x0	One shot force next field polarity when written 00 = does not force next field 01 = force only next field to odd 10 = force only next field to even 11 = does not force next field

Interlaced timing control for CRTC1

D1CRTC_INTERLACE_STATUS - RW - 32 bits - [GpuF0MMReg:0x608C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_INTERLACE_CURRENT_FIELD (R)	0	0x0	Reports the polarity of current field 0 = even 1 = odd
D1CRTC_INTERLACE_NEXT_FIELD (R)	1	0x0	Reports the polarity of the next field. Normally the opposite of the current field. When D1CRTC_INTERLACE_FORCE_NEXT_FIELD is used to force polarity of next field, then next field can match current field. 0 = even 1 = odd

Read-only register reports the polarity of the current and next field for interlaced timing

D1CRTC_BLANK_DATA_COLOR - RW - 32 bits - [GpuF0MMReg:0x6090]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_BLANK_DATA_COLOR_BLUE_CB	9:0	0x0	B / Cb component
D1CRTC_BLANK_DATA_COLOR_GREEN_Y	19:10	0x0	G / Y component
D1CRTC_BLANK_DATA_COLOR_RED_CR	29:20	0x0	R / Cr component

Set the color for pixels in blank region

D1CRTC_OVERSCAN_COLOR - RW - 32 bits - [GpuF0MMReg:0x6094]

Field Name	Bits	Default	Description
D1CRTC_OVERSCAN_COLOR_BLUE	9:0	0x0	B or Cb component
D1CRTC_OVERSCAN_COLOR_GREEN	19:10	0x0	G or Y component
D1CRTC_OVERSCAN_COLOR_RED	29:20	0x0	R or Cr component

Defines color of the overscan region for CRTC1

D1CRTC_BLACK_COLOR - RW - 32 bits - [GpuF0MMReg:0x6098]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_BLACK_COLOR_B_CB	9:0	0x0	B / Cb component of the black color
D1CRTC_BLACK_COLOR_G_Y	19:10	0x0	G / Y component of the black color
D1CRTC_BLACK_COLOR_R_CR	29:20	0x0	R / Cr component of the black color

Black color applied to the active display region when blanking the screen

D1CRTC_STATUS - RW - 32 bits - [GpuF0MMReg:0x609C]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_V_BLANK (R)	0	0x0	Current vertical position 0 = outside vertical blank region 1 = within vertical blank region
D1CRTC_V_ACTIVE_DISP (R)	1	0x0	Current vertical position 0 = outside vertical active display region 1 = within vertical active display region
D1CRTC_V_SYNC_A (R)	2	0x0	Current vertical position 0 = outside VSYNC 1 = within VSYNC
D1CRTC_V_UPDATE (R)	3	0x0	Current vertical position 0 = outside the V_UPDATE region 1 = within the V_UPDATE region (between end of vertical active display and start_line)
D1CRTC_V_START_LINE (R)	4	0x0	Current vertical position 0 = outside start_line region 1 = within start_line region
D1CRTC_H_BLANK (R)	16	0x0	Current horizontal position 0 = outside horizontal blank region 1 = within horizontal blank region
D1CRTC_H_ACTIVE_DISP (R)	17	0x0	Current horizontal region 0 = outside horizontal active display region 1 = within horizontal active display region
D1CRTC_H_SYNC_A (R)	18	0x0	Current horizontal position 0 = outside horizontal sync 1 = within horizontal sync

Reports the position of CRTC1

D1CRTC_STATUS_POSITION - RW - 32 bits - [GpuF0MMReg:0x60A0]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_VERT_COUNT (R)	12:0	0x0	Reports current vertical count
D1CRTC_HORZ_COUNT (R)	28:16	0x0	Reports current horizontal count

Current horizontal and vertical count of CRTC1

D1CRTC_STATUS_FRAME_COUNT - RW - 32 bits - [GpuF0MMReg:0x60A4]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_FRAME_COUNT (R)	23:0	0x0	Reports current frame count

Current frame count for CRTC1

D1CRTC_STATUS_VF_COUNT - RW - 32 bits - [GpuF0MMReg:0x60A8]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_VF_COUNT (R)	28:0	0x0	Reports current vertical and frame count

Current composite vertical and frame count for CRTC1

D1CRTC_STATUS_HV_COUNT - RW - 32 bits - [GpuF0MMReg:0x60AC]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_HV_COUNT (R)	28:0	0x0	Reports current horizontal and vertical count

Current composite H/V count of CRTC1

D1CRTC_COUNT_RESET - RW - 32 bits - [GpuF0MMReg:0x60B0]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_RESET_FRAME_COUNT (W)	0	0x0	One-shot reset of frame counter of CRTC1 when written with '1'

Resets CRTC1 counters

D1CRTC_COUNT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60B4]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_HORZ_COUNT_BY2_EN	0	0x0	Set to 1 for DVI 30bpp mode only, set to 0 otherwise
D1CRTC_HORZ_REPETITION_COUNT	4:1	0x0	Enable horizontal repetition. CRTC increments the H counter every (COUNT+1) pixel clocks 0 = every clock 1 = every 2 clocks 2 = every 3 clocks etc

Controls the counters in CRTC1

D1CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE - RW - 32 bits - [GpuF0MMReg:0x60B8]

Field Name	Bits	Default	Description
D1CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE (W)	0	0x0	One shot force VSYNCA to happen next line when written with '1'

Manual force of VSYNC to happen next line

D1CRTC_VERT_SYNC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60BC]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_FORCE_VSYNC_NEXT_LINE_OCCURRED (R)	0	0x0	Reports whether force vsync next line event has occurred. Sticky bit. 0 = event has not occurred 1 = event has occurred
D1CRTC_FORCE_VSYNC_NEXT_LINE_CLEAR (W)	8	0x0	One shot clear to the sticky bit D1CRTC_FORCE_VSYNC_NEXT_LINE_OCCURRED when written with '1'
D1CRTC_AUTO_FORCE_VSYNC_MODE	17:16	0x0	Selection of auto mode for forcing vsync next line 00 = disables auto mode 01 = force VSYNC next line on CRTC trigger A signal 10 = force VSYNC next line on CRTC trigger B signal 11 = reserved

Controls the feature to force VSYNC next line for CRTC1

D1CRTC_STEREO_STATUS - RW - 32 bits - [GpuF0MMReg:0x60C0]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_STEREO_CURRENT_EYE (R)	0	0x0	Reports the polarity of the current frame/field 0 = right eye image 1 = left eye image
D1CRTC_STEREO_SYNC_OUTPUT (R)	8	0x0	Reports current value of STEREOSYNC signal (STEREOSYNC sent to the DISPOUT block)
D1CRTC_STEREO_SYNC_SELECT (R)	16	0x0	Reports current value of SYNC_SELECT signal (SYNC_SELECT sent to the SCL block)
D1CRTC_STEREO_FORCE_NEXT_EYE_PENDING (R)	25:24	0x0	Reports the status of D1CRTC_STEREO_FORCE_NEXT_EYE write. 00: No force pending 01: Right force pending 10: Left force pending 11: Reserved

Reports CRTC1 status in stereoscopic display

D1CRTC_STEREO_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60C4]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_STEREO_SYNC_OUTPUT_POLARITY	8	0x0	Controls polarity of the stereosync signal 0 = 0 means right eye image and 1 means left eye image 1 = 0 means left eye image and 1 means right eye image
D1CRTC_STEREO_SYNC_SELECT_POLARITY	16	0x0	Controls polarity of STEREO_SELECT signal sent to scaler 0 = 0 means right eye image and 1 means left eye image 1 = 0 means left eye image and 1 means right eye image

D1CRTC_STEREO_EN	24	0x0	Enables toggling of STEREOSYNC and STEREO_SELECT signals 0 = disable toggling. 1 = enable toggling at every frame (progressive) or every field (interlace) at leading edge of VSYNCA

Stereosync control for CRTC1

D1CRTC_SNAPSHOT_STATUS - RW - 32 bits - [GpuF0MMReg:0x60C8]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_SNAPSHOT_OCCURRED (R)	0	0x0	Reports status of snapshot. A sticky bit to be cleared by writing 1 to D1CRTC_SNAPSHOT_CLEAR 0 = snapshot has not occurred 1 = snapshot has occurred
D1CRTC_SNAPSHOT_CLEAR (W)	1	0x0	Clears the D1CRTC_SNAPSHOT_OCCURRED sticky bit when written with '1'
D1CRTC_SNAPSHOT_MANUAL_TRIGGER (W)	2	0x0	One shot trigger to perform snapshot when written with '1'

Controls CRTC1 snapshot

D1CRTC_SNAPSHOT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60CC]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_AUTO_SNAPSHOT_TRIG_SEL	1:0	0x0	Determines signal source for auto-snapshot 00 = auto-snapshot is disabled 01 = uses CRTC trigger A as trigger event in auto-snapshot mode 10 = uses CRTC trigger B as trigger event in auto-snapshot mode 11 = reserved

Controls snapshot mode for CRTC1

D1CRTC_SNAPSHOT_POSITION - RW - 32 bits - [GpuF0MMReg:0x60D0]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_SNAPSHOT_VERT_COUNT (R)	12:0	0x0	Reads back the snapshoted vertical count
D1CRTC_SNAPSHOT_HORZ_COUNT (R)	28:16	0x0	Reads back the snapshoted horizontal count

Snapshot H and V count for CRTC1

D1CRTC_SNAPSHOT_FRAME - RW - 32 bits - [GpuF0MMReg:0x60D4]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_SNAPSHOT_FRAME_COUNT (R)	23:0	0x0	Reports the snapshoted frame count

Snapshot frame count of CRTC1

D1CRTC_START_LINE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60D8]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_PROGRESSIVE_START_LINE_EARLY	0	0x0	move start_line signal by 1 line eariler in progressive mode
D1CRTC_INTERLACE_START_LINE_EARLY	8	0x1	move start_line signal by 1 line earlier in interlaced timing mode

move start_line signal earlier by 1 line in CRTC1

D1CRTC_INTERRUPT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60DC]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_SNAPSHOT_INT_MSK	0	0x0	Interrupt mask for CRTC snapshot event 0 = disables interrupt 1 = enables interrupt
D1CRTC_SNAPSHOT_INT_TYPE	1	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D1CRTC_V_UPDATE_INT_MSK	4	0x0	Interrupt mask for falling edge of V_UPDATE ^M 0 = disables interrupt^M 1 = enables interrupt
D1CRTC_V_UPDATE_INT_TYPE	5	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D1CRTC_FORCE_COUNT_NOW_INT_MSK	8	0x0	Interrupt mask for force count now event 0 = disables interrupt 1 = enables interrupt
D1CRTC_FORCE_COUNT_NOW_INT_TYPE	9	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D1CRTC_FORCE_VSYNC_NEXT_LINE_INT_MSK	16	0x0	Interrupt mask for force VSYNC next line event 0 = disables interrupt 1 = enables interrupt
D1CRTC_FORCE_VSYNC_NEXT_LINE_INT_TYPE	17	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D1CRTC_TRIGA_INT_MSK	24	0x0	Interrupt mask for CRTC external trigger A 0 = disables interrupt 1 = enables interrupt
D1CRTC_TRIGB_INT_MSK	25	0x0	Interrupt mask for CRTC external trigger B 0 = disables interrupt 1 = enables interrupt
D1CRTC_TRIGA_INT_TYPE	26	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D1CRTC_TRIGB_INT_TYPE	27	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt

Interrupt mask for CRTC1 events

D1MODE_MASTER_UPDATE_LOCK - RW - 32 bits - [GpuF0MMReg:0x60E0]			
Field Name	**Bits**	**Default**	**Description**
D1MODE_MASTER_UPDATE_LOCK	0	0x0	Set the master update lock for V_UPDATE signal 0 = no master lock, V_UPDATE signal will occur 1 = set master lock to prevent V_UPDATE signal occuring, thus prevent double buffering of display registers

Master update lock for CRTC1 V_UPDATE signal

D1MODE_MASTER_UPDATE_MODE - RW - 32 bits - [GpuF0MMReg:0x60E4]			
Field Name	**Bits**	**Default**	**Description**
D1MODE_MASTER_UPDATE_MODE	2:0	0x0	Controls the position of the V_UPDATE signal 000 = V_UPDATE occurs between end of active display region and start line signal 001 = V_UPDATE occurs at first leading edge of HSYNCA after leading edge of VSYNCA 010 = V_UPDATE occurs at the leading edge of VSYNC_A 011 = V_UPDATE occurs at the beginning of the first line of vertical front porch 100 = V_UPDATE occurs at end of the line before start line Others = Reserved
D1MODE_MASTER_UPDATE_INTERLACED_MODE	17:16	0x0	Controls generation of V_UPDATE signal in interlaced mode 00 = generates V_UPDATE at both even and odd field 01 = generates V_UPDATE only at even field. when D1MODE_MASTER_UPDATE_MODE = 00, V_UPDATE starts at odd field and ends at even field 10 = generates V_UPDATE only at odd field. when D1MODE_MASTER_UPDATE_MODE = 00, V_UPDATE starts at even field and ends at odd field 11 = reserved

Controls the generation of the V_UPDATE signal in CRTC1

D1CRTC_UPDATE_LOCK - RW - 32 bits - [GpuF0MMReg:0x60E8]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_UPDATE_LOCK	0	0x0	Set the lock for CRTC timing registers 0 = no lock, double buffering can occur 1 = set lock to prevent double buffering

Update lock for CRTC1 timing registers

D1CRTC_DOUBLE_BUFFER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x60EC]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_UPDATE_PENDING (R)	0	0x0	Reports the status of double-buffered timing registers in CRTC1 0 = update has completed 1 = update is still pending
D1CRTC_UPDATE_INSTANTLY	8	0x0	Disables double buffering of CRTC1 timing registers 0 = enables double buffering 1 = disables double buffering
D1CRTC_BLANK_DATA_DOUBLE_BUFFER_EN	16	0x0	Enables the double buffering of D1CRTC_BLANK_DATA_EN 0 = disables double buffering. D1CRTC_BLANK_DATA_EN is updated immediately 1 = enables double buffering of D1CRTC_BLANK_DATA_EN when V_UPDATE is active

Controls double buffering of CRTC1 registers

D1CRTC_VGA_PARAMETER_CAPTURE_MODE - RW - 32 bits - [GpuF0MMReg:0x60F0]			
Field Name	**Bits**	**Default**	**Description**
D1CRTC_VGA_PARAMETER_CAPTURE_MODE	0	0x0	Controls how VGA timing parameters are captured. 0: CRTC1 will continuously latch in timing parameters from VGA 1: CRTC1 will continuously latch in timing parameters from VGA except during VGA parameter recalculated window

Controls how VGA timing parameters are captured

2.8.2 Secondary Display CRTC Control Registers

D2CRTC_H_TOTAL - RW - 32 bits - [GpuF0MMReg:0x6800]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_H_TOTAL	12:0	0x0	Horizontal total minus one. Sum of display width, overscan left and right, front and back porch and H sync width. E.g. for 800 pixels set to 799 = 0x31F Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Defines horizontal dimension of the display timing for CRTC2

D2CRTC_H_BLANK_START_END - RW - 32 bits - [GpuF0MMReg:0x6804]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_H_BLANK_START	12:0	0x0	Start of the horizontal blank. The location of the first pixel of horizontal blank, relative to pixel zero. If right overscan border, then blank starts after border ends. Double-buffered with D2MODE_MASTER_UPDATE_LOCK
D2CRTC_H_BLANK_END	28:16	0x0	End of the horizontal blank. The location of the next pixel after the last pixel of horizontal blank, relative to pixel zero. Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Defines horizontal blank region of the display timing for CRTC2

D2CRTC_H_SYNC_A - RW - 32 bits - [GpuF0MMReg:0x6808]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_H_SYNC_A_START	12:0	0x0	First pixel of horizontal sync A. In normal cases, it is set to 0. It is only set to non-zero value when we want to test the higher bits of the H counter. This register should be ignored and set to 0x0 in VGA timing mode. Hardware does not support odd number value for this register.

D2CRTC_H_SYNC_A_END	28:16	0x0	Horizontal sync A end. Determines position of the next pixel after last pixel of horizontal sync A. The last pixel of horizontal sync A is D2CRTC_H_SYNC_A_END - 1. The first pixel of horizontal sync A is pixel 0. It should be programmed to a value one greater than the actual last pixel of horizontal sync A. Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Defines horizontal sync A position for CRTC2

D2CRTC_H_SYNC_A_CNTL - RW - 32 bits - [GpuF0MMReg:0x680C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_H_SYNC_A_POL	0	0x0	Polarity of H SYNC A 0 = active high 1 = active low Double-buffered with D2MODE_MASTER_UPDATE_LOCK
D2CRTC_COMP_SYNC_A_EN	16	0x0	Enables composite H sync A 0 = disabled 1 = enabled
D2CRTC_H_SYNC_A_CUTOFF	17	0x0	Cutoff H sync A at end of H BLANK when end of H sync A is beyond H BLANK 0 = cutoff is enabled 1 = cutoff is disabled

Controls the H SYNC A for CRTC1

D2CRTC_H_SYNC_B - RW - 32 bits - [GpuF0MMReg:0x6810]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_H_SYNC_B_START	12:0	0x0	First pixel of horizontal sync B
D2CRTC_H_SYNC_B_END	28:16	0x0	Horizontal sync B end. Determines position of the next pixel after last pixel of horizontal sync B. The last pixel of horizontal sync B is D2CRTC_H_SYNC_B_END - 1. This register value is exclusive. It should be programmed to a value one greater than the actual last pixel of horizontal sync B

Defines the position of horizontal sync B for CRTC2

D2CRTC_H_SYNC_B_CNTL - RW - 32 bits - [GpuF0MMReg:0x6814]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_H_SYNC_B_POL	0	0x0	Polarity of H SYNC B 0 = active high 1 = active low
D2CRTC_COMP_SYNC_B_EN	16	0x0	Enables composite H SYNC B 0 = disabled 1 = enabled
D2CRTC_H_SYNC_B_CUTOFF	17	0x0	Cutoff horizontal sync B at end of horizontal blank region when end of H SYNC B is beyond horizontal blank 0 = cutoff is enabled 1 = cutoff is disabled

Controls horizontal sync B for CRTC2

D2CRTC_VBI_END - RW - 32 bits - [GpuF0MMReg:0x6818]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_VBI_V_END	12:0	0x3	VBI drops when this number of complete horizontal line remains before the start of v active and D2CRTC_VBI_H_END reached
D2CRTC_VBI_H_END	28:16	0x0	VBI drops when this number of H pixel remains before the start of v active and D2CRTC_VBI_V_END reached

VBI goes to CG to tell CG when crtc is in non v active region (i.e. is asserted during VBLANK region + vertical overscan) and is safe to change mclk, VBI can be programmed to be de-asserted earlier than start of v active to prevent CG from changing mclk close to the start v active

D2CRTC_V_TOTAL - RW - 32 bits - [GpuF0MMReg:0x6820]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_TOTAL	12:0	0x0	Vertical total minus one. Sum of vertical active display, top and bottom overscan, front and back porch and vertical sync width. E.g. for 525 lines set to 524 = 0x20C Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Defines the vertical dimension of display timing for CRTC2

D2CRTC_V_BLANK_START_END - RW - 32 bits - [GpuF0MMReg:0x6824]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_BLANK_START	12:0	0x0	Vertical blank start. Determines the position of the first blank line in a frame. Line 0 is the first line of vertical sync A. Double-buffered with D2MODE_MASTER_UPDATE_LOCK
D2CRTC_V_BLANK_END	28:16	0x0	Vertical blank end. Determines the position of the next line after the last line of vertical blank. The last line of vertical blank is D2CRTC_V_BLANK_END - 1. Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Defines the position of the vertical blank region for CRTC2

D2CRTC_V_SYNC_A - RW - 32 bits - [GpuF0MMReg:0x6828]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_SYNC_A_START	12:0	0x0	The first line of vertical sync A. In normal cases, it is set to 0. It is set to non-zero value only when trying to test the higher bits of the vertical counter

D2CRTC_V_SYNC_A_END	28:16	0x0	Vertical sync A end. Determines the position of the next line after the last line of vertical sync A. The last line of vertical sync A is D2CRTC_V_SYNC_A_END - 1. The first line of vertical sync A is line 0. This register value is exclusive. It should be programmed to a value one greater than the actual last line of vertical sync A Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Defines the position of vertical sync A for CRTC2

D2CRTC_V_SYNC_A_CNTL - RW - 32 bits - [GpuF0MMReg:0x682C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_SYNC_A_POL	0	0x0	Polarity of V SYNC A 0 = active high 1 = active low Double-buffered with D2MODE_MASTER_UPDATE_LOCK

Controls V SYNC A for CRTC2

D2CRTC_V_SYNC_B - RW - 32 bits - [GpuF0MMReg:0x6830]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_SYNC_B_START	12:0	0x0	Vertical sync B start. Determines the position of the first line of vertical sync B.
D2CRTC_V_SYNC_B_END	28:16	0x0	Vertical sync B end. Determines the position of the next line after the last line of vertical sync B. Last line of vertical sync B is D2CRTC_V_SYNC_B_END - 1. This register value is exclusive. It should be programmed to a value one greater than the actual last line of vertical sync B

Defines the position of vertical sync B for CRTC2

D2CRTC_V_SYNC_B_CNTL - RW - 32 bits - [GpuF0MMReg:0x6834]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_SYNC_B_POL	0	0x0	Controls polarity of vertical sync B 0 = active high 1 = active low

Controls vertical sync B for CRTC2

D2CRTC_TRIGA_CNTL - RW - 32 bits - [GpuF0MMReg:0x6860]			
Field Name	**Bits**	**Default**	**Description**

D2CRTC_TRIGA_SOURCE_SELECT	3:0	0x0	Select source of input signals for external trigger A 0 = logic 0 1 = VSYNCA from another CRTC of the chip 2 = HSYNCA from another CRTC of the chip 3 = VSYNCB from another CRTC of the chip 4 = HSYNCB from another CRTC of the chip 5 = GENERICA pin 6 = GENERICB pin 7 = VSYNCA pin 8 = HSYNCA pin 9 = VSYNCB pin 10 = HSYNCB pin 11 = HPD1 pin 12 = HPD2 pin 13 = DVALID pin 14 = PSYNC pin 15 = Video capture complete signal from VIP
D2CRTC_TRIGA_POLARITY_SELECT	6:4	0x0	Selects source of input signal from polarity of external trigger A 0 = logic 0 1 = interlace polarity from another CRTC of the chip 2 = GENERICA pin 3 = GENERICB pin 4 = HSYNCA pin 5 = HSYNCB pin 6 = video capture polarity input from VIP 7 = DVALID pin
D2CRTC_TRIGA_RESYNC_BYPASS_EN	8	0x0	Bypass the resync logic for the external trigger A signal and its polarity input signal 0 = do not bypass 1 = bypass the resync logic
D2CRTC_TRIGA_INPUT_STATUS (R)	9	0x0	Read back the value of the external trigger A input signal after the mux
D2CRTC_TRIGA_POLARITY_STATUS (R)	10	0x0	Reports the value of the external trigger A polarity signal after the mux
D2CRTC_TRIGA_OCCURRED (R)	11	0x0	Reports whether external trigger A has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D2CRTC_TRIGA_RISING_EDGE_DETECT_CNTL	13:12	0x0	Controls the detection of rising edge of the external trigger A signal 00 = do not detect rising edge 01 = always detect rising edge 10 = detect rising edge only when field polarity is low 11 = detect rising edge only when field polarity is high
D2CRTC_TRIGA_FALLING_EDGE_DETECT_CNTL	17:16	0x0	Controls the detection of falling edge of external trigger A signal 00 = do not detect falling edge 01 = always detect falling edge 10 = detect falling edge only when field polarity is low 11 = detect falling edge only when field polarity is high
D2CRTC_TRIGA_FREQUENCY_SELECT	21:20	0x0	Determines the frequency of the external trigger A signal 00 = send every signal 01 = send every 2 signals 10 = reserved 11 = send every 4 signals
D2CRTC_TRIGA_DELAY	28:24	0x0	A programmable delay to send external trigger A signal
D2CRTC_TRIGA_CLEAR (W)	31	0x0	Clears the sticky bit D2CRTC_TRIGA_OCCURRED when written with '1'

Controls for external trigger A signal in CRTC2

D2CRTC_TRIGA_MANUAL_TRIG - RW - 32 bits - [GpuF0MMReg:0x6864]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_TRIGA_MANUAL_TRIG (W)	0	0x0	One shot trigger for external trigger A signal when written with '1'

Manual trigger for external trigger A signal of CRTC2

D2CRTC_TRIGB_CNTL - RW - 32 bits - [GpuF0MMReg:0x6868]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_TRIGB_SOURCE_SELECT	3:0	0x0	Select source of input signals for external trigger B 0 = logic 0 1 = VSYNCA from another CRTC of the chip 2 = HSYNCA from another CRTC of the chip 3 = VSYNCB from another CRTC of the chip 4 = HSYNCB from another CRTC of the chip 5 = GENERICA pin 6 = GENERICB pin 7 = VSYNCA pin 8 = HSYNCA pin 9 = VSYNCB pin 10 = HSYNCB pin 11 = HPD1 pin 12 = HPD2 pin 13 = DVALID pin 14 = PSYNC pin 15 = Video capture complete signal from VIP
D2CRTC_TRIGB_POLARITY_SELECT	6:4	0x0	Selects source of input signal from polarity of external trigger B 0 = logic 0 1 = interlace polarity from another CRTC of the chip 2 = GENERICA pin 3 = GENERICB pin 4 = HSYNCA pin 5 = HSYNCB pin 6 = video capture polarity input from VIP 7 = DVALID pin
D2CRTC_TRIGB_RESYNC_BYPASS_EN	8	0x0	Bypass the resync logic for the external trigger B signal and its polarity input signal 0 = do not bypass 1 = bypass the resync logic
D2CRTC_TRIGB_INPUT_STATUS (R)	9	0x0	Read back the value of the external trigger B input signal after the mux
D2CRTC_TRIGB_POLARITY_STATUS (R)	10	0x0	Reports the value of the external trigger B polarity signal after the mux
D2CRTC_TRIGB_OCCURRED (R)	11	0x0	Reports whether external trigger B has occurred. A sticky bit. 0 = has not occurred 1 = has occurred
D2CRTC_TRIGB_RISING_EDGE_DETECT_CNTL	13:12	0x0	Controls the detection of rising edge of the external trigger B signal 00 = do not detect rising edge 01 = always detect rising edge 10 = detect rising edge only when field polarity is low 11 = detect rising edge only when field polarity is high

D2CRTC_TRIGB_FALLING_EDGE_DETECT_CNTL	17:16	0x0	Controls the detection of falling edge of external trigger B signal 00 = do not detect falling edge 01 = always detect falling edge 10 = detect falling edge only when field polarity is low 11 = detect falling edge only when field polarity is high
D2CRTC_TRIGB_FREQUENCY_SELECT	21:20	0x0	Determines the frequency of the external trigger B signal 00 = send every signal 01 = send every 2 signals 10 = reserved 11 = send every 4 signals
D2CRTC_TRIGB_DELAY	28:24	0x0	A programmable delay to send external trigger B signal
D2CRTC_TRIGB_CLEAR (W)	31	0x0	Clears the sticky bit D2CRTC_TRIGB_OCCURRED when written with '1'

Control for external trigger B signal of CRTC2

D2CRTC_TRIGB_MANUAL_TRIG - RW - 32 bits - [GpuF0MMReg:0x686C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_TRIGB_MANUAL_TRIG (W)	0	0x0	One shot trigger for external trigger B signal when written with '1'

Manual trigger for external trigger B signal of CRTC2

D2CRTC_FORCE_COUNT_NOW_CNTL - RW - 32 bits - [GpuF0MMReg:0x6870]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_FORCE_COUNT_NOW_MODE	1:0	0x0	Controls which timing counter is forced 0 = force counter now mode is disabled 1 = force H count now to H_TOTAL only 2 = force H count to H_TOTAL and V count to V_TOTAL in progressive mode and V_TOTAL-1 in interlaced mode 3 = reserved
D2CRTC_FORCE_COUNT_NOW_TRIG_SEL	8	0x0	Selects the trigger signal as force count now trigger 0 = selects CRTC_TRIG_A and CRTC_TRIG_A_POL 1 = selects CRTC_TRIG_B and CRTC_TRIG_B_POL
D2CRTC_FORCE_COUNT_NOW_OCCURRED (R)	16	0x0	Reports the status of force count now, a sticky bit. 0 = CRTC force count now has not occurred 1 = CRTC force count now has occurred
D2CRTC_FORCE_COUNT_NOW_CLEAR (W)	24	0x0	Resets D2CRTC_FORCE_COUNT_NOW_OCCURRED when written with '1'

Controls CRTC2 force count now logic

D2CRTC_FLOW_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6874]			
Field Name	**Bits**	**Default**	**Description**

D2CRTC_FLOW_CONTROL_SOURCE_SELECT	4:0	0x0	Selects the signal used for flow control in CRTC2 0 = logic 0 1 = GENERICA pin 2 = GENERICB pin 3 = HPD1 pin 4 = HPD2 pin 5 = DDC1DATA pin 6 = DDC1CLK pin 7 = DDC2DATA pin 8 = DDC2CLK pin 9 = DVOCLK(1) pin 10 = VHAD(0] pin 11 = VHAD[1] pin 12 = VPHCTL pin 13 = VIPCLK pin 14 = DVALID pin 15 = PSYNC pin 16 = a GPIO pin for dual-GPU, TBD
D2CRTC_FLOW_CONTROL_POLARITY	8	0x0	Reports the status of force count now, a sticky bit. 0 = CRTC force count now has not occurred 1 = CRTC force count now has occurred
D2CRTC_FLOW_CONTROL_GRANULARITY	16	0x0	Controls at which pixel position flow control can start to happen 0 = flow control only start to happen on odd-even pixel boundary 1 = flow control can start at any pixel position
D2CRTC_FLOW_CONTROL_INPUT_STATUS (R)	24	0x0	Reports the value of the flow control input signal 0 = output of source mux of flow control signal is low 1 = output of source mux of flow control signal is high

Controls flow control of CRTC2

D2CRTC_PIXEL_DATA_READBACK - RW - 32 bits - [GpuF0MMReg:0x6878]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_PIXEL_DATA_BLUE_CB (R)	9:0	0x0	B/Cb component sent to DISPOUT
D2CRTC_PIXEL_DATA_GREEN_Y (R)	19:10	0x0	G/Y component sent to DISPOUT
D2CRTC_PIXEL_DATA_RED_CR (R)	29:20	0x0	R/Cr component sent to DISPOUT

Read back of the CRTC2 pixel data sent to DISPOUT. This is a debug register. Intended for use in one shot clocking mode.

D2CRTC_STEREO_FORCE_NEXT_EYE - RW - 32 bits - [GpuF0MMReg:0x687C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_STEREO_FORCE_NEXT_EYE (W)	1:0	0x0	Force next frame eye view - One shot. 00: No force - next eye opposite of current eye 01: Right eye force - force right eye next field/frame 10: Left eye force - force right eye next field/frame 11: Reserved After a force has occured, readback of this register will be 00

Force Next Eye register

D2CRTC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6880]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_MASTER_EN	0	0x0	Enables/Disables CRTC2. H counter is at H_TOTAL and V counter is at first line of blank when CRTC is disabled. 0 = disabled 1 = enabled
D2CRTC_SYNC_RESET_SEL	4	0x1	Allows power management to lower CRTC2 enable.
D2CRTC_DISABLE_POINT_CNTL	9:8	0x1	When D2CRTC_MASTER_EN is set to 0, delay the disabling of CRTC2 until certain point within the frame 00 = disable CRTC immediately 01 = delay disable CRTC until the end of the current line 10 = reserved 11 = delay disable CRTC until end of the first line in the vertical blank region
D2CRTC_CURRENT_MASTER_EN_STATE (R)	16	0x0	Read-only field indicates the current status of the timing generator. Can be used to poll for when a delayed disable takes effect. 0 = CRTC is disabled 1 = CRTC is enabled
D2CRTC_DISP_READ_REQUEST_DISABLE	24	0x0	Disables data read request from the display controller. Can be used to stop display reads from system memory but keep display timing generation running. Has no effect if CRTC is disabled. 0 = do not disable data read request 1 = disable data read request
D2CRTC_PREFETCH_EN	28	0x0	Double buffered. Enable data prefetch for display 1 0 = do not enable prefetch 1 = enable data prefetch
D2CRTC_SOF_PULL_EN	29	0x0	At SOF, LB level can be set a a programmable value (D2MODE_SOF_READ_PT), which is the point LB can make requests to. Between SOF to active line, CRTC needs to pull scaler/LB so that LB can make data requests beyond that programmable point. 0 = do not enable pulling 1 = enable pulling

Controls CRTC2 timing generator and data read request to display2

D2CRTC_BLANK_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6884]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_CURRENT_BLANK_STATE (R)	0	0x0	Read only status indicating current state of display blanking. 0 = screen not blanked 1 = screen is blanked
D2CRTC_BLANK_DATA_EN	8	0x0	Enable for blanking active display area. The active area of display that is forced will use the D2CRTC_BLACK_COLOR value. This field is optionally double buffered with D2CRTC_BLANK_DATA_DOUBLE_BUFFER_EN. 0 = disable blanking 1 = enable blanking

D2CRTC_BLANK_DE_MODE	16	0x0	Determines whether BLANK and DATA_ACTIVE signal keeps toggling when screen is blank 0 = toggles BLANK and DATA_ACTIVE 1 = keep BLANK active and DATA_ACTIVE inactive

Controls forced blanking of active area of display timing. Useful for display mode switches when corrupted image may be generated for a frame or two.

D2CRTC_INTERLACE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x6888]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_INTERLACE_ENABLE	0	0x0	Enables interlaced timing 0 = Progressive timing 1 = Interlaced timing
D2CRTC_INTERLACE_FORCE_NEXT_FIELD (W)	17:16	0x0	One shot force next field polarity when written 00 = does not force next field 01 = force only next field to odd 10 = force only next field to even 11 = does not force next field

Interlaced timing control for CRTC2

D2CRTC_INTERLACE_STATUS - RW - 32 bits - [GpuF0MMReg:0x688C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_INTERLACE_CURRENT_FIELD (R)	0	0x0	Reports the polarity of current field 0 = even 1 = odd
D2CRTC_INTERLACE_NEXT_FIELD (R)	1	0x0	Reports the polarity of the next field. Normally the opposite of the current field. When D2CRTC_INTERLACE_FORCE_NEXT_FIELD is used to force polarity of next field, then next field can match current field. 0 = even 1 = odd

Read-only register reports the polarity of the current and next field for interlaced timing

D2CRTC_BLANK_DATA_COLOR - RW - 32 bits - [GpuF0MMReg:0x6890]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_BLANK_DATA_COLOR_BLUE_CB	9:0	0x0	B / Cb component
D2CRTC_BLANK_DATA_COLOR_GREEN_Y	19:10	0x0	G / Y component
D2CRTC_BLANK_DATA_COLOR_RED_CR	29:20	0x0	R / Cr component

Set the color for pixels in blank region

D2CRTC_OVERSCAN_COLOR - RW - 32 bits - [GpuF0MMReg:0x6894]

Field Name	Bits	Default	Description
D2CRTC_OVERSCAN_COLOR_BLUE	9:0	0x0	B or Cb component
D2CRTC_OVERSCAN_COLOR_GREEN	19:10	0x0	G or Y component
D2CRTC_OVERSCAN_COLOR_RED	29:20	0x0	R or Cr component

Defines color of the overscan region for CRTC2

D2CRTC_BLACK_COLOR - RW - 32 bits - [GpuF0MMReg:0x6898]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_BLACK_COLOR_B_CB	9:0	0x0	B / Cb component of the black color
D2CRTC_BLACK_COLOR_G_Y	19:10	0x0	G / Y component of the black color
D2CRTC_BLACK_COLOR_R_CR	29:20	0x0	R / Cr component of the black color

Black color applied to the active display region when blanking the screen

D2CRTC_STATUS - RW - 32 bits - [GpuF0MMReg:0x689C]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_V_BLANK (R)	0	0x0	Current vertical position 0 = outside vertical blank region 1 = within vertical blank region
D2CRTC_V_ACTIVE_DISP (R)	1	0x0	Current vertical position 0 = outside vertical active display region 1 = within vertical active display region
D2CRTC_V_SYNC_A (R)	2	0x0	Current vertical position 0 = outside VSYNC 1 = within VSYNC
D2CRTC_V_UPDATE (R)	3	0x0	Current vertical position 0 = outside the V_UPDATE region 1 = within the V_UPDATE region (between end of vertical active display and start_line]
D2CRTC_V_START_LINE (R)	4	0x0	Current vertical position 0 = outside start_line region 1 = within start_line region
D2CRTC_H_BLANK (R)	16	0x0	Current horizontal position 0 = outside horizontal blank region 1 = within horizontal blank region
D2CRTC_H_ACTIVE_DISP (R)	17	0x0	Current horizontal region 0 = outside horizontal active display region 1 = within horizontal active display region
D2CRTC_H_SYNC_A (R)	18	0x0	Current horizontal position 0 = outside horizontal sync 1 = within horizontal sync

Reports the position of CRTC2

D2CRTC_STATUS_POSITION - RW - 32 bits - [GpuF0MMReg:0x68A0]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_VERT_COUNT (R)	12:0	0x0	Reports current vertical count
D2CRTC_HORZ_COUNT (R)	28:16	0x0	Reports current horizontal count

Current horizontal and vertical count of CRTC2

D2CRTC_STATUS_FRAME_COUNT - RW - 32 bits - [GpuF0MMReg:0x68A4]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_FRAME_COUNT (R)	23:0	0x0	Reports current frame count

Current frame count for CRTC2

D2CRTC_STATUS_VF_COUNT - RW - 32 bits - [GpuF0MMReg:0x68A8]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_VF_COUNT (R)	28:0	0x0	Reports current vertical and frame count

Current composite vertical and frame count for CRTC2

D2CRTC_STATUS_HV_COUNT - RW - 32 bits - [GpuF0MMReg:0x68AC]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_HV_COUNT (R)	28:0	0x0	Reports current horizontal and vertical count

Current composite H/V count of CRTC2

D2CRTC_COUNT_RESET - RW - 32 bits - [GpuF0MMReg:0x68B0]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_RESET_FRAME_COUNT (W)	0	0x0	One-shot reset of frame counter of CRTC2 when written with '1'

Resets CRTC2 counters

D2CRTC_COUNT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68B4]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_HORZ_COUNT_BY2_EN	0	0x0	Set to 1 for DVI 30bpp mode only, set to 0 otherwise
D2CRTC_HORZ_REPETITION_COUNT	7:4	0x0	Enable horizontal repetition. CRTC increments the H counter every (COUNT+1) pixel clocks 0 = every clock 1 = every 2 clocks 2 = every 3 clocks etc

Controls the counters in CRTC2

D2CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE - RW - 32 bits - [GpuF0MMReg:0x68B8]

Field Name	Bits	Default	Description
D2CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE (W)	0	0x0	One shot force VSYNCA to happen next line when written with '1'

Manual force of VSYNC to happen next line

D2CRTC_VERT_SYNC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68BC]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_FORCE_VSYNC_NEXT_LINE_OCCURRED (R)	0	0x0	Reports whether force vsync next line event has occurred. Sticky bit. 0 = event has not occurred 1 = event has occurred
D2CRTC_FORCE_VSYNC_NEXT_LINE_CLEAR (W)	8	0x0	One shot clear to the sticky bit D1CRTC_FORCE_VSYNC_NEXT_LINE_OCCURRED when written with '1'
D2CRTC_AUTO_FORCE_VSYNC_MODE	17:16	0x0	Selection of auto mode for forcing vsync next line 00 = disables auto mode 01 = force VSYNC next line on CRTC trigger A signal 10 = force VSYNC next line on CRTC trigger B signal 11 = reserved

Controls the feature to force VSYNC next line for CRTC2

D2CRTC_STEREO_STATUS - RW - 32 bits - [GpuF0MMReg:0x68C0]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_STEREO_CURRENT_EYE (R)	0	0x0	Reports the polarity of the current frame/field 0 = right eye image 1 = left eye image
D2CRTC_STEREO_SYNC_OUTPUT (R)	8	0x0	Reports current value of STEREOSYNC signal
D2CRTC_STEREO_SYNC_SELECT (R)	16	0x0	Reports current value of SYNC_SELECT signal
D2CRTC_STEREO_FORCE_NEXT_EYE_PENDING (R)	25:24	0x0	Reports the status of D2CRTC_STEREO_FORCE_NEXT_EYE write. 00: No force pending 01: Right force pending 10: Left force pending 11: Reserved

Reports CRTC2 status in stereoscopic display

D2CRTC_STEREO_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68C4]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_STEREO_SYNC_OUTPUT_POLARITY	8	0x0	Controls polarity of the stereosync signal 0 = 0 means right eye image and 1 means left eye image 1 = 0 means left eye image and 1 means right eye image
D2CRTC_STEREO_SYNC_SELECT_POLARITY	16	0x0	Controls polarity of STEREO_SELECT signal sent to scaler 0 = 0 means right eye image and 1 means left eye image 1 = 0 means left eye image and 1 means right eye image

D2CRTC_STEREO_EN	24	0x0	Enables toggling of STEREOSYNC and STEREO_SELECT signals 0 = disable toggling. 1 = enable toggling at every frame (progressive) or every field (interlace) at leading edge of VSYNCA

Stereosync control for CRTC2

D2CRTC_SNAPSHOT_STATUS - RW - 32 bits - [GpuF0MMReg:0x68C8]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_SNAPSHOT_OCCURRED (R)	0	0x0	Reports status of snapshot. A sticky bit to be cleared by writing 1 to D2CRTC_SNAPSHOT_CLEAR 0 = snapshot has not occurred 1 = snapshot has occurred
D2CRTC_SNAPSHOT_CLEAR (W)	1	0x0	Clears the D2CRTC_SNAPSHOT_OCCURRED sticky bit when written with '1'
D2CRTC_SNAPSHOT_MANUAL_TRIGGER (W)	2	0x0	One shot trigger to perform snapshot when written with '1'

Controls CRTC2 snapshot

D2CRTC_SNAPSHOT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68CC]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_AUTO_SNAPSHOT_TRIG_SEL	1:0	0x0	Determines signal source for auto-snapshot 00 = auto-snapshot is disabled 01 = uses CRTC trigger A as trigger event in auto-snapshot mode 10 = uses CRTC trigger B as trigger event in auto-snapshot mode 11 = reserved

Controls snapshot mode for CRTC2

D2CRTC_SNAPSHOT_POSITION - RW - 32 bits - [GpuF0MMReg:0x68D0]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_SNAPSHOT_VERT_COUNT (R)	12:0	0x0	Reads back the snapshoted vertical count
D2CRTC_SNAPSHOT_HORZ_COUNT (R)	28:16	0x0	Reads back the snapshoted horizontal count

Snapshot H and V count for CRTC2

D2CRTC_SNAPSHOT_FRAME - RW - 32 bits - [GpuF0MMReg:0x68D4]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_SNAPSHOT_FRAME_COUNT (R)	23:0	0x0	Reports the snapshoted frame count

Snapshot frame count of CRTC2

D2CRTC_START_LINE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68D8]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_PROGRESSIVE_START_LINE_EARLY	0	0x0	move start_line signal by 1 line eariler in progressive mode
D2CRTC_INTERLACE_START_LINE_EARLY	8	0x1	move start_line signal by 1 line earlier in interlaced timing mode

move start_line signal earlier by 1 line in CRTC2

D2CRTC_INTERRUPT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68DC]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_SNAPSHOT_INT_MSK	0	0x0	Interrupt mask for CRTC snapshot event 0 = disables interrupt 1 = enables interrupt
D2CRTC_SNAPSHOT_INT_TYPE	1	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D2CRTC_V_UPDATE_INT_MSK	4	0x0	Interrupt mask for falling edge of V_UPDATE ^M 0 = disables interrupt^M 1 = enables interrupt
D2CRTC_V_UPDATE_INT_TYPE	5	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D2CRTC_FORCE_COUNT_NOW_INT_MSK	8	0x0	Interrupt mask for force count now event 0 = disables interrupt 1 = enables interrupt
D2CRTC_FORCE_COUNT_NOW_INT_TYPE	9	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D2CRTC_FORCE_VSYNC_NEXT_LINE_INT_MSK	16	0x0	Interrupt mask for force VSYNC next line event 0 = disables interrupt 1 = enables interrupt
D2CRTC_FORCE_VSYNC_NEXT_LINE_INT_TYPE	17	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D2CRTC_TRIGA_INT_MSK	24	0x0	Interrupt mask for CRTC external trigger A 0 = disables interrupt 1 = enables interrupt
D2CRTC_TRIGB_INT_MSK	25	0x0	Interrupt mask for CRTC external trigger B 0 = disables interrupt 1 = enables interrupt
D2CRTC_TRIGA_INT_TYPE	26	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt
D2CRTC_TRIGB_INT_TYPE	27	0x0	0 is legacy level based interrupt, 1 is pulse based interrupt

Interrupt mask for CRTC2 events

D2MODE_MASTER_UPDATE_LOCK - RW - 32 bits - [GpuF0MMReg:0x68E0]			
Field Name	**Bits**	**Default**	**Description**
D2MODE_MASTER_UPDATE_LOCK	0	0x0	Set the master update lock for V_UPDATE signal 0 = no master lock, V_UPDATE signal will occur 1 = set master lock to prevent V_UPDATE signal occuring, thus prevent double buffering of display registers

Master update lock for CRTC2 V_UPDATE signal

D2MODE_MASTER_UPDATE_MODE - RW - 32 bits - [GpuF0MMReg:0x68E4]			
Field Name	**Bits**	**Default**	**Description**
D2MODE_MASTER_UPDATE_MODE	2:0	0x0	Controls the position of the V_UPDATE signal 000 = V_UPDATE occurs between end of active display region and start line signal 001 = V_UPDATE occurs when leading edge of HSYNCA meets leading edge of VSYNCA 010 = V_UPDATE occurs at the leading edge of VSYNC_A 011 = V_UPDATE occurs at the beginning of the first line of vertical front porch 100 = V_UPDATE occurs at end of the line before start line Others = Reserved
D2MODE_MASTER_UPDATE_INTERLACED_MODE	17:16	0x0	Controls generation of V_UPDATE signal in interlaced mode 00 = generates V_UPDATE at both even and odd field 01 = generates V_UPDATE only at even field. when D1MODE_MASTER_UPDATE_MODE = 00, V_UPDATE starts at odd field and ends at even field 10 = generates V_UPDATE only at odd field. when D1MODE_MASTER_UPDATE_MODE = 00, V_UPDATE starts at even field and ends at odd field 11 = reserved

Controls the generation of the V_UPDATE signal in CRTC2

D2CRTC_UPDATE_LOCK - RW - 32 bits - [GpuF0MMReg:0x68E8]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_UPDATE_LOCK	0	0x0	Set the lock for CRTC timing registers 0 = no lock, double buffering can occur 1 = set lock to prevent double buffering

Update lock for CRTC2 timing registers

D2CRTC_DOUBLE_BUFFER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x68EC]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_UPDATE_PENDING (R)	0	0x0	Reports the status of double-buffered timing registers in CRTC2 0 = update has completed 1 = update is still pending
D2CRTC_UPDATE_INSTANTLY	8	0x0	Disables double buffering of CRTC2 timing registers 0 = enables double buffering 1 = disables double buffering
D2CRTC_BLANK_DATA_DOUBLE_BUFFER_EN	16	0x0	Enables the double buffering of D2CRTC_BLANK_DATA_EN 0 = disables double buffering. D2CRTC_BLANK_DATA_EN is updated immediately 1 = enables double buffering of D2CRTC_BLANK_DATA_EN when V_UPDATE is active

Controls double buffering of CRTC2 registers

D2CRTC_VGA_PARAMETER_CAPTURE_MODE - RW - 32 bits - [GpuF0MMReg:0x68F0]			
Field Name	**Bits**	**Default**	**Description**
D2CRTC_VGA_PARAMETER_CAPTURE_MODE	0	0x0	Controls how VGA timing parameters are captured. 0: CRTC2 will continuously latch in timing parameters from VGA 1: CRTC2 will continuously latch in timing parameters from VGA except during VGA parameter recalculated window

Controls how VGA timing parameters are captured

2.9 Display Output Registers

2.9.1 Digital to Analog Converter (DAC) Registers

Registers for controlling the DAC links.

DAC A Registers

DACA_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7800]			
Field Name	**Bits**	**Default**	**Description**
DACA_ENABLE	0	0x0	0=Disable 1=Enable

Turn on/off DACA

DACA_SOURCE_SELECT - RW - 32 bits - [GpuF0MMReg:0x7804]			
Field Name	**Bits**	**Default**	**Description**
DACA_SOURCE_SELECT	1:0	0x0	0=Source is CRTC1 1=Source is CRTC2 2=Source is TV Encoder 3=Reserved

Select between 1st display, 2nd display & TV encoder streams

DACA_CRC_EN - RW - 32 bits - [GpuF0MMReg:0x7808]			
Field Name	**Bits**	**Default**	**Description**
DACA_CRC_EN	0	0x0	Enable signal for DACA CRC 0=Disable 1=Enable
DACA_CRC_CONT_EN	16	0x0	Determines whether CRC is calculated continuously or for one frame (one shot) 0=CRC is calculated over 1 frame 1=CRC is continuously calculated for every frame

DACA CRC enable signals

DACA_CRC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x780C]			
Field Name	**Bits**	**Default**	**Description**
DACA_CRC_FIELD	0	0x0	Controls which field polarity starts the DACA CRC block after DACA_CRC_EN is set high. Used only for interlaced mode CRCs 0=Even field begins CRC calculation 1=Odd field begins CRC calculation
DACA_CRC_ONLY_BLANKb	8	0x0	Determines whether CRC is calculated for the whole frame or only during non-blank period for DACA 0=CRC calculated over entire field 1=CRC calculated only during BLANKb

DACA CRC controls signals

DACA_CRC_SIG_RGB_MASK - RW - 32 bits - [GpuF0MMReg:0x7810]			
Field Name	**Bits**	**Default**	**Description**
DACA_CRC_SIG_BLUE_MASK	9:0	0x3ff	Mask bits for DACA B channel CRC
DACA_CRC_SIG_GREEN_MASK	19:10	0x3ff	Mask bits for DACA G channel CRC
DACA_CRC_SIG_RED_MASK	29:20	0x3ff	Mask bits for DACA R channel CRC

Mask bits for R, G & B CRC calculations

DACA_CRC_SIG_CONTROL_MASK - RW - 32 bits - [GpuF0MMReg:0x7814]			
Field Name	**Bits**	**Default**	**Description**
DACA_CRC_SIG_CONTROL_MASK	5:0	0x3f	Mask bits for DACA control signal CRC

Mask bits for DACA control signal CRC

DACA_CRC_SIG_RGB - RW - 32 bits - [GpuF0MMReg:0x7818]			
Field Name	**Bits**	**Default**	**Description**
DACA_CRC_SIG_BLUE (R)	9:0	0x3ff	CRC signature value for DACA blue component
DACA_CRC_SIG_GREEN (R)	19:10	0x3ff	CRC signature value for DACA green component
DACA_CRC_SIG_RED (R)	29:20	0x3ff	CRC signature value for DACA red component

DACA CRC R, G & B results

DACA_CRC_SIG_CONTROL - RW - 32 bits - [GpuF0MMReg:0x781C]			
Field Name	**Bits**	**Default**	**Description**
DACA_CRC_SIG_CONTROL (R)	5:0	0x3f	CRC signature value for DACA control signals

CRC signature value for DACA control signals

DACA_SYNC_TRISTATE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7820]			
Field Name	**Bits**	**Default**	**Description**
DACA_HSYNCA_TRISTATE	0	0x0	DACA hsync tristate. Used to determine hsynca enable
DACA_VSYNCA_TRISTATE	8	0x0	DACA vsync tristate. Used to determine vsynca enable
DACA_SYNCA_TRISTATE	16	0x0	DACA sync tristate. Used to determine sync enables

DACA SYNC Tristate control

DACA_SYNC_SELECT - RW - 32 bits - [GpuF0MMReg:0x7824]

Field Name	Bits	Default	Description
DACA_SYNC_SELECT	0	0x0	0: selects sync_a 1: selects sync_b. Used in conjunction with DACA_SOURCE_SEL(0). 0=DACA uses HSYNC_A & VSYNC_A 1=DACA used HSYNC_B & VSYNC_B
DACA_STEREOSYNC_SELECT	8	0x0	0: selects crtc1 stereosync 1: selects crtc2 stereosync 0=DACA uses CRTC1 STEREOSYNC 1=DACA uses CRTC2 STEREOSYNC

DACA ...SYNC selection

DACA_AUTODETECT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7828]			
Field Name	**Bits**	**Default**	**Description**
DACA_AUTODETECT_MODE	1:0	0x0	Operation control of DACA Autodetect logic: 0: No checking 1: Connection checking 2: Disconnection checking
DACA_AUTODETECT_FRAME_TIME_COUNTER	15:8	0x0	If an enabled display pipe is connected to DACA, autodetect logic will count number of frames before DACA comparator enabled.Otherwise, the autodetect logic will count number of 0.1-second units.
DACA_AUTODETECT_CHECK_MASK	18:16	0x7	Mask to select which of the 3 RGB channels will be checked for connection or disconnection. Bit 18: Check R/C channel if bit set to 1. Bit 17: Check G/Y channel if bit set to 1. Bit 16: Check B/Comp channel if bit set to 1.

DACA_AUTODETECT_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x782C]			
Field Name	**Bits**	**Default**	**Description**
DACA_AUTODETECT_POWERUP_COUNTER	7:0	0xb	DACA macro Bandgap voltage reference power up time. Default = 11 microseconds.
DACA_AUTODETECT_TESTMODE	8	0x0	0: Normal operation 1: Test mode - count in 1us units

DACA_AUTODETECT_CONTROL3 - RW - 32 bits - [GpuF0MMReg:0x7830]			
Field Name	**Bits**	**Default**	**Description**
DACA_AUTODET_COMPARATOR_IN_DELAY	7:0	0x19	DACA comparator delay for inputs to settle in autodetect mode. Default = 25us
DACA_AUTODET_COMPARATOR_OUT_DELAY	15:8	0x5	DACA comparator delay for outputs to settle in autodetect mode. Default = 5us

DACA_AUTODETECT_STATUS - RW - 32 bits - [GpuF0MMReg:0x7834]			
Field Name	**Bits**	**Default**	**Description**
DACA_AUTODETECT_STATUS (R)	0	0x0	Result from autodetect logic sequence: 0: DACA was looking for a connection and has yet found a connection or DACA was looking for a disconnection has not yet found a disconnection 1: DACA was looking for a connection and found a connection or DACA was looking for a disconnection and found a disconnection
DACA_AUTODETECT_CONNECT (R)	4	0x0	1: At least one channel has a properly terminated device connected. 0: No devices are connected
DACA_AUTODETECT_RED_SENSE (R)	9:8	0x0	Two bit result from last Red/C compare: 0: Channel is disconnected 1: Channel is connected 2: Channel is not checked 3: Reserved
DACA_AUTODETECT_GREEN_SENSE (R)	17:16	0x0	Two bit result from last Green/Y compare: 0: Channel is disconnected 1: Channel is connected 2: Channel is not checked 3: Reserved
DACA_AUTODETECT_BLUE_SENSE (R)	25:24	0x0	Two bit result from last Blue/Comp compare: 0: Channel is disconnected 1: Channel is connected 2: Channel is not checked 3: Reserved

DACA_AUTODETECT_INT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7838]			
Field Name	**Bits**	**Default**	**Description**
DACA_AUTODETECT_ACK (W)	0	0x0	Auto detect interrupt acknowledge and clear DACA_AUTODETECT_STATUS bit.
DACA_AUTODETECT_INT_ENABLE	16	0x0	Enable for auto detect interrupt 0=Disable 1=Enable

DACA_FORCE_OUTPUT_CNTL - RW - 32 bits - [GpuF0MMReg:0x783C]			
Field Name	**Bits**	**Default**	**Description**
DACA_FORCE_DATA_EN	0	0x0	Enable synchronous force option on DACA. 0=Disable 1=Enable
DACA_FORCE_DATA_SEL	10:8	0x0	Select which DACA channels have data forced 0=Don't Force, 1=ForceBit 0: Blue channelBit 1: Green channelBit 2: Red channel
DACA_FORCE_DATA_ON_BLANKb_ONLY	24	0x0	Data is force only during active region. 0=Disable 1=Enable

Data Force Control

DACA_FORCE_DATA - RW - 32 bits - [GpuF0MMReg:0x7840]			
Field Name	**Bits**	**Default**	**Description**
DACA_FORCE_DATA	9:0	0x0	Data to be forced on R, G & B channels. When auto detect logic is enabled, this must be programmed to 0x000 (Default).

DACA_POWERDOWN - RW - 32 bits - [GpuF0MMReg:0x7850]			
Field Name	**Bits**	**Default**	**Description**
DACA_POWERDOWN	0	0x0	Bandgap Voltage Reference Power down enable (BGSLEEP)
DACA_POWERDOWN_BLUE	8	0x0	Blue channel power down enable (BDACPD)
DACA_POWERDOWN_GREEN	16	0x0	Green channel power down enable (GDACPD)
DACA_POWERDOWN_RED	24	0x0	Red channel power down enable (RDACPD)

Controls for DACA Start-Up & Power-Down sequences

DACA_CONTROL1 - RW - 32 bits - [GpuF0MMReg:0x7854]			
Field Name	**Bits**	**Default**	**Description**
DACA_WHITE_LEVEL	1:0	0x0	Video Standard Select bits - STD(1:0) 0x0: PAL 0x1: NTSC PS2 (VGA) 0x3 HDTV (Component Video)
DACA_WHITE_FINE_CONTROL	13:8	0x20	Full-scale Output Adjustment - DACADJ(4:0)
DACA_BANDGAP_ADJUSTMENT	21:16	0x20	Bandgap Reference Voltage Adjustment - BGADJ(3:0)
DACA_ANALOG_MONITOR	27:24	0x0	Analog test mux select - MON(3:0)
DACA_COREMON	28	0x0	Core voltage monitor input port

DACA_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x7858]			
Field Name	**Bits**	**Default**	**Description**
DACA_DFORCE_EN	0	0x0	DACA asynchronous data force enable. Can be used for sync force as well but DACA_FORCE_OUTPUT_CNTL achieves the same goal with a more complete feature set. Asynchronous force requires DACA_x_ASYNC_ENABLE in DACA_COMPARATOR_ENABLE to be set as well. Drives DFORCE_EN pin on macro. Forces all DACA channels to DACA_FORCE_DATA value. Overrides DACA_FORCE_OUTPUT_CNTL/DACA_FORCE_DATA_EN control.
DACA_TV_ENABLE	8	0x0	

DACA_ZSCALE_SHIFT	16	0x0	DACA zero scale shift enable. Causes DACA to add a small offset to the levels of all outputs. Drives DACA ZSCALE_SHIFT pin.

DACA_COMPARATOR_ENABLE - RW - 32 bits - [GpuF0MMReg:0x785C]			
Field Name	**Bits**	**Default**	**Description**
DACA_COMP_DDET_REF_EN	0	0x0	Enables DACA comparators for analog termination checking with DDETECT_REF as the reference. The DDETECT reference level is lower than SDETECT_REF to allow termination checking on an active channel while the data being driven is the ZSCALE_SHIFT offset.Must be used in conjunction with ZSCALE_SHIFT=1 and with some forced data on the DAC inputs. Only one of COMP_DDET_REF_EN and COMP_SDET_REF_EN should be active at a time.Used in conjunction with core logic to drive the DAC DDETECT pin. 0=Disable 1=Enable
DACA_COMP_SDET_REF_EN	8	0x0	Enables DACA comparators for analog termination checking with SDETECT_REF as the reference. The data must be forced to a sufficiently high value using one of the DAC force features. Only one of COMP_DDET_REF_EN and COMP_SDET_REF_EN should be active at a time.Goes directly to the DAC SDETECT pin. 0=Disable 1=Enable
DACA_R_ASYNC_ENABLE	16	0x0	DACA red channel asynchronous mode enable.Allows DAC outputs to be updated without a clock.Used in conjunction with core logic to drive the DAC R_ASYNC_EN pin. 0=Disable 1=Enable
DACA_G_ASYNC_ENABLE	17	0x0	DACA green channel asynchronous mode enable.Used in conjunction with core logic to drive the DAC G_ASYNC_EN pin. 0=Disable 1=Enable
DACA_B_ASYNC_ENABLE	18	0x0	DACA blue channel asynchronous mode enable.Used in conjunction with core logic to drive the DAC B_ASYNC_EN pin. 0=Disable 1=Enable

DACA_COMPARATOR_OUTPUT - RW - 32 bits - [GpuF0MMReg:0x7860]			
Field Name	**Bits**	**Default**	**Description**
DACA_COMPARATOR_OUTPUT (R)	0	0x0	Monitor Detect Output. This signal is an AND of 3 DAC macro signals: R_CDET, G_YDET & B_COMPDET.
DACA_COMPARATOR_OUTPUT_BLUE (R)	1	0x0	DACA blue channel comparator output ? value comes from DAC R_CDET pin
DACA_COMPARATOR_OUTPUT_GREEN (R)	2	0x0	DACA green channel comparator output ? value comes from DAC G_YDET pin

DACA_COMPARATOR_OUTPUT_RED (R)	3	0x0	DACA red channel comparator output ? value comes from DAC B_COMPDET pin

DACA_TEST_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7864]			
Field Name	**Bits**	**Default**	**Description**
DACA_TEST_ENABLE	0	0x0	DACATEST Enable 0=Disable 1=Enable

DACA_PWR_CNTL - RW - 32 bits - [GpuF0MMReg:0x7868]			
Field Name	**Bits**	**Default**	**Description**
DACA_BG_MODE	1:0	0x0	Bandgap macro configuration - BGMODE(1:0)
DACA_PWRCNTL	17:16	0x0	Macro bias current level control - PWRCNTL(1:0)

DACA_DFT_CONFIG - RW - 32 bits - [GpuF0MMReg:0x786C]			
Field Name	**Bits**	**Default**	**Description**
DACA_DFT_CONFIG	31:0	0x0	Configuration for DACA DFT block

DAC B Registers

DACB_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7A00]			
Field Name	**Bits**	**Default**	**Description**
DACB_ENABLE	0	0x0	0=Disable 1=Enable

Turn on/off DACB

DACB_SOURCE_SELECT - RW - 32 bits - [GpuF0MMReg:0x7A04]			
Field Name	**Bits**	**Default**	**Description**

DACB_SOURCE_SELECT	1:0	0x0	0=Source is CRTC1 1=Source is CRTC2 2=Source is TV Encoder 3=Reserved

Select between 1st display, 2nd display & TV encoder streams

DACB_CRC_EN - RW - 32 bits - [GpuF0MMReg:0x7A08]			
Field Name	**Bits**	**Default**	**Description**
DACB_CRC_EN	0	0x0	Enable signal for DACB CRC 0=Disable 1=Enable
DACB_CRC_CONT_EN	16	0x0	Determines whether CRC is calculated for the whole frame or only during non-blank period for DACB 0=Disable 1=Enable

DACB CRC enable signals

DACB_CRC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7A0C]			
Field Name	**Bits**	**Default**	**Description**
DACB_CRC_FIELD	0	0x0	Controls which field polarity starts the DACB CRC block after DACA_CRC_EN is set high. Used only for interlaced mode CRCs. 0=Even field begins CRC calculation 1=Odd field begins CRC calculation
DACB_CRC_ONLY_BLANKb	8	0x0	CRC only during the Non-blank region 0=CRC calculated over entire field 1=CRC calculated only during BLANKb

DACB CRC controls signals

DACB_CRC_SIG_RGB_MASK - RW - 32 bits - [GpuF0MMReg:0x7A10]			
Field Name	**Bits**	**Default**	**Description**
DACB_CRC_SIG_BLUE_MASK	9:0	0x3ff	Mask bits for DACB B channel CRC
DACB_CRC_SIG_GREEN_MASK	19:10	0x3ff	Mask bits for DACB G channel CRC
DACB_CRC_SIG_RED_MASK	29:20	0x3ff	Mask bits for DACB R channel CRC

Mask bits for R, G & B CRC calculations

DACB_CRC_SIG_CONTROL_MASK - RW - 32 bits - [GpuF0MMReg:0x7A14]			
Field Name	**Bits**	**Default**	**Description**
DACB_CRC_SIG_CONTROL_MASK	5:0	0x3f	Mask bits for DACB control signal CRC

Mask bits for DACB control signal CRC

DACB_CRC_SIG_RGB - RW - 32 bits - [GpuF0MMReg:0x7A18]			
Field Name	**Bits**	**Default**	**Description**
DACB_CRC_SIG_BLUE (R)	9:0	0x3ff	CRC signature value for DACB blue component
DACB_CRC_SIG_GREEN (R)	19:10	0x3ff	CRC signature value for DACB green component
DACB_CRC_SIG_RED (R)	29:20	0x3ff	CRC signature value for DACB red component

DACB CRC R, G & B results

DACB_CRC_SIG_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7A1C]			
Field Name	**Bits**	**Default**	**Description**
DACB_CRC_SIG_CONTROL (R)	5:0	0x3f	CRC signature value for DACB control signals

CRC signature value for DACB control signals

DACB_SYNC_TRISTATE_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7A20]			
Field Name	**Bits**	**Default**	**Description**
DACB_HSYNCB_TRISTATE	0	0x0	DACB hsync tristate. Used to determine hsyncb enable
DACB_VSYNCB_TRISTATE	8	0x0	DACB vsync tristate. Used to determine vsyncb enable
DACB_SYNCB_TRISTATE	16	0x0	DACB sync tristate. Used to determine sync enables

DACB SYNC Tristate control

DACB_SYNC_SELECT - RW - 32 bits - [GpuF0MMReg:0x7A24]			
Field Name	**Bits**	**Default**	**Description**
DACB_SYNC_SELECT	0	0x0	0=DACB uses HSYNC_A & VSYNC_A 1=DACB used HSYNC_B & VSYNC_B
DACB_STEREOSYNC_SELECT	8	0x0	0=DACB uses CRTC1 STEREOSYNC 1=DACB uses CRTC2 STEREOSYNC

DACB_AUTODETECT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7A28]			
Field Name	**Bits**	**Default**	**Description**
DACB_AUTODETECT_MODE	1:0	0x0	Operation control of DACB Autodetect logic: 0: No checking 1: Connection checking 2: Disconnection checking
DACB_AUTODETECT_FRAME_TIME_COUNTER	15:8	0x0	If an enabled display pipe is connected to DACB, autodetect logic will count number of frames before DACB comparator enabled.Otherwise, the autodetect logic will count number of 0.1-second units.

DACB_AUTODETECT_CHECK_MASK	18:16	0x7	Mask to select which of the 3 RGB channels will be checked for connection or disconnection. Bit 18: Check R/C channel if bit set to 1. Bit 17: Check G/Y channel if bit set to 1. Bit 16: Check B/Comp channel if bit set to 1.

DACB_AUTODETECT_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x7A2C]			
Field Name	**Bits**	**Default**	**Description**
DACB_AUTODETECT_POWERUP_COUNTER	7:0	0xb	DACB macro Bandgap voltage reference power up time. Default = 11 microseconds.
DACB_AUTODETECT_TESTMODE	8	0x0	0: Normal operation 1: Test mode - count in 1us units

DACB_AUTODETECT_CONTROL3 - RW - 32 bits - [GpuF0MMReg:0x7A30]			
Field Name	**Bits**	**Default**	**Description**
DACB_AUTODET_COMPARATOR_IN_DELAY	7:0	0x19	DACB comparator delay for inputs to settle in autodetect mode. Default = 25us
DACB_AUTODET_COMPARATOR_OUT_DELAY	15:8	0x5	DACB comparator delay for outputs to settle in autodetect mode. Default = 5us

DACB_AUTODETECT_STATUS - RW - 32 bits - [GpuF0MMReg:0x7A34]			
Field Name	**Bits**	**Default**	**Description**
DACB_AUTODETECT_STATUS (R)	0	0x0	Result from autodetect logic sequence: 0: DACB was looking for a connection and has yet found a connection or DACB was looking for a disconnection has not yet found a disconnection 1: DACB was looking for a connection and found a connection or DACB was looking for a disconnection and did not find a disconnection
DACB_AUTODETECT_CONNECT (R)	4	0x0	0=No devices are connected 1=At least one channel has a properly terminated device connected
DACB_AUTODETECT_RED_SENSE (R)	9:8	0x0	Two bit result from last Red/C compare: 0: Channel is disconnected 1: Channel is connected 2: Channel is not checked 3: Reserved
DACB_AUTODETECT_GREEN_SENSE (R)	17:16	0x0	Two bit result from last Green/Y compare: 0: Channel is disconnected 1: Channel is connected 2: Channel is not checked 3: Reserved

DACB_AUTODETECT_BLUE_SENSE (R)	25:24	0x0	Two bit result from last Blue/Comp compare: 0: Channel is disconnected 1: Channel is connected 2: Channel is not checked 3: Reserved

DACB_AUTODETECT_INT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7A38]			
Field Name	**Bits**	**Default**	**Description**
DACB_AUTODETECT_ACK (W)	0	0x0	Auto detect interrupt acknowledge and clear DACB_AUTODETECT_STATUS bit.
DACB_AUTODETECT_INT_ENABLE	16	0x0	Enable for auto detect interrupt 0=Disable 1=Enable

DACB_FORCE_OUTPUT_CNTL - RW - 32 bits - [GpuF0MMReg:0x7A3C]			
Field Name	**Bits**	**Default**	**Description**
DACB_FORCE_DATA_EN	0	0x0	Enable synchronous force option on DACB 0=Disable 1=Enable
DACB_FORCE_DATA_SEL	10:8	0x0	Select which DACB channels have data forced 0=Don't Force, 1=Force Bit 0: Blue channel Bit 1: Green channel Bit 2: Red channel
DACB_FORCE_DATA_ON_BLANKb_ONLY	24	0x0	Data is force only during active region. 0=Disable 1=Enable

Data Force Control

DACB_FORCE_DATA - RW - 32 bits - [GpuF0MMReg:0x7A40]			
Field Name	**Bits**	**Default**	**Description**
DACB_FORCE_DATA	9:0	0x0	Data to be forced on R, G & B channels

DACB_POWERDOWN - RW - 32 bits - [GpuF0MMReg:0x7A50]			
Field Name	**Bits**	**Default**	**Description**
DACB_POWERDOWN	0	0x0	Bandgap Voltage Reference Power down enable (BGSLEEP)ANDed with controls from power management and LVTMA power sequencer.
DACB_POWERDOWN_BLUE	8	0x0	Blue channel power-down enable - BDACPD
DACB_POWERDOWN_GREEN	16	0x0	Green channel power-down enable - GDACPD
DACB_POWERDOWN_RED	24	0x0	Red channel power-down enable - RDACPD

Controls for DACB Start-Up & Power-Down sequences

DACB_CONTROL1 - RW - 32 bits - [GpuF0MMReg:0x7A54]			
Field Name	**Bits**	**Default**	**Description**
DACB_WHITE_LEVEL	1:0	0x0	Video Standard Select bits - STD(1:0) 0x0: PAL 0x1: NTSC PS2 (VGA) 0x3 HDTV (Component Video)
DACB_WHITE_FINE_CONTROL	13:8	0x20	Full-scale Output Adjustment - DACADJ(4:0)
DACB_BANDGAP_ADJUSTMENT	21:16	0x20	Bandgap Reference Voltage Adjustment - BGADJ(3:0)
DACB_ANALOG_MONITOR	27:24	0x0	Analog test mux select - MON(3:0)
DACB_COREMON	28	0x0	Core voltage monitor input port

DACB_CONTROL2 - RW - 32 bits - [GpuF0MMReg:0x7A58]			
Field Name	**Bits**	**Default**	**Description**
DACB_DFORCE_EN	0	0x0	DACB asynchronous data force enable. Can be used for sync force as well but DACB_FORCE_OUTPUT_CNTL achieves the same goal with a more complete feature set. Async force requires async bits in DACB_COMPARATOR_ENABLE to be set as well. Drives DFORCE_EN pin on macro. Forces all DACB channels to DACB_FORCE_DATA value. Overrides DACB_FORCE_OUTPUT_CNTL/DACB_FORCE_DATA_EN control.
DACB_TV_ENABLE	8	0x0	DACB tv enable. Controls DACB output demux. R/G/B is selected when TV_ENABLE=0, Y/C/Comp when TV_ENABLE=1. Drives DAC TVENABLE input.
DACB_ZSCALE_SHIFT	16	0x0	DACB zero scale shift enable. Causes DAC to add a small offset to the levels of all outputs. Drives DAC ZSCALE_SHIFT pin.

DACB_COMPARATOR_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7A5C]			
Field Name	**Bits**	**Default**	**Description**
DACB_COMP_DDET_REF_EN	0	0x0	Enables DACB comparators for analog termination checking with DDETECT_REF as the reference. The DDETECT reference level is lower than SDETECT_REF to allow termination checking on an active channel while the data being driven is the ZSCALE_SHIFT offset.Must be used in conjunction with ZSCALE_SHIFT=1 or with some forced data on the DAC inputs. Only one of COMP_DDET_REF_EN and COMP_SDET_REF_EN should be active at a time.Used in conjunction with core logic to drive the DAC DDETECT pin. 0=Disable 1=Enable

DACB_COMP_SDET_REF_EN	8	0x0	Enables DACB comparators for analog termination checking with SDETECT_REF as the reference. The data must be forced to a sufficiently high value using one of the DAC force features. Only one of COMP_DDET_REF_EN and COMP_SDET_REF_EN should be active at a time.Goes directly to the DAC SDETECT pin. 0=Disable 1=Enable
DACB_R_ASYNC_ENABLE	16	0x0	DACB red channel asynchronous mode enable.Allows DAC outputs to be updated without a clock.Used in conjunction with core logic to drive the DAC R_ASYNC_EN pin. 0=Disable 1=Enable
DACB_G_ASYNC_ENABLE	17	0x0	DACB green channel asynchronous mode enable.Used in conjunction with core logic to drive the DAC G_ASYNC_EN pin. 0=Disable 1=Enable
DACB_B_ASYNC_ENABLE	18	0x0	DACB blue channel asynchronous mode enable.Used in conjunction with core logic to drive the DAC B_ASYNC_EN pin. 0=Disable 1=Enable

DACB_COMPARATOR_OUTPUT - RW - 32 bits - [GpuF0MMReg:0x7A60]			
Field Name	**Bits**	**Default**	**Description**
DACB_COMPARATOR_OUTPUT (R)	0	0x0	Monitor Detect Output. This signal is an AND of 4 dac macro signals: DETECT, RDACDET, GDACDET & BDACDET.
DACB_COMPARATOR_OUTPUT_BLUE (R)	1	0x0	DACB blue channel comparator output ? value comes from DAC BDACDET pin
DACB_COMPARATOR_OUTPUT_GREEN (R)	2	0x0	DACB green channel comparator output ? value comes from DAC GDACDET pin
DACB_COMPARATOR_OUTPUT_RED (R)	3	0x0	DACB red channel comparator output ? value comes from DAC RDACDET pin

DACB_TEST_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7A64]			
Field Name	**Bits**	**Default**	**Description**
DACB_TEST_ENABLE	0	0x0	0=Disable 1=Enable

DACBTEST Enable. Use for DAC test only. Drives DAC

DACB_PWR_CNTL - RW - 32 bits - [GpuF0MMReg:0x7A68]			
Field Name	**Bits**	**Default**	**Description**
DACB_BG_MODE	1:0	0x0	DACB bandgap macro configuration. Allows bandgap macro to be configured to optimize performance.Goes directly to DAC BG_MODE[1:0] input.

DACB_PWRCNTL	17:16	0x0	DACB bias current level control. Allows analog bias current levels to be adjusted for performance vs. power consumption tradeoff.Goes directly to DAC PWRCNTL[1:0] input.

Transition-Minimized Digital Stream (TMDS) Registers

TMDSA_CNTL - RW - 32 bits - [GpuF0MMReg:0x7880]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_ENABLE	0	0x0	Enable for the reduction/encoding logic 0=Disable 1=Enable
TMDSA_HDMI_EN	2	0x0	Select DVI or HDMI mode 0=DVI 1=HDMI
TMDSA_ENABLE_HPD_MASK	4	0x0	0:Disallow 1:Allow override of TMDSA_ENABLE by HPD on disconnect 0=Result from HPD circuit can not override TMDSA_ENABLE 1=Result from HPD circuit can override TMDSA_ENABLE on disconnect
TMDSA_HPD_SELECT	9:8	0x0	Select which hot plug detect unit to use for TMDSA. This selection is only relevant if one of the HPD mask bits in this and other other registers is enabled. 0=Use HPD1 1=Use HPD2 2=use HPD3
TMDSA_SYNC_PHASE	12	0x1	Determine whether to reset phase signal on frame pulse 0: don't reset 1: reset
TMDSA_PIXEL_ENCODING	16	0x0	0=RGB 4:4:4 or YCBCR 4:4:4 1=YCbCr 4:2:2
TMDSA_DUAL_LINK_ENABLE	24	0x0	Enable dual-link 0=Disable 1=Enable
TMDSA_SWAP	28	0x0	Swap upper and lower data channels 0=Disable 1=Enable

TMDSA_SOURCE_SELECT - RW - 32 bits - [GpuF0MMReg:0x7884]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_SOURCE_SELECT	0	0x0	Select between display stream 1 & display stream 2 0=CRTC1 data is used 1=CRTC2 data is used

TMDSA_SYNC_SELECT	8	0x0	Select between SYNCA and SYNCB signals 0=HSYNC_A & VSYNC_A from the selected CRTC are used 1=HSYNC_B & VSYNC_B from the selected CRTC are used
TMDSA_STEREOSYNC_SELECT	16	0x0	Select between CRTC1 and CRTC2 sterosync signals 0=CRTC1 STEREOSYNC used 1=CRTC2 STEREOSYNC used

Source Select control for Data, H/VSYNC & Stereosync

TMDSA_COLOR_FORMAT - RW - 32 bits - [GpuF0MMReg:0x7888]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_COLOR_FORMAT	1:0	0x0	Controls TMDSA output colour format. Formats 0 and 1 work in single or dual link. Format 2 requires dual link (MSBs on primary link, LSBs on secondary link). 0=Normal (24bpp), Twin-Single 30bpp (8 MSBs of each component), or Dual-Link 48bpp 1=Twin-Link 30bpp (2 LSB of each component) 2=Dual-Link 30bpp 3=Reserved

TMDSA_FORCE_OUTPUT_CNTL - RW - 32 bits - [GpuF0MMReg:0x788C]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_FORCE_DATA_EN	0	0x0	Enable force option on TMDSA 0=Disable 1=Enable
TMDSA_FORCE_DATA_SEL	10:8	0x0	Select TMDSA channels that have data forced0=Don't Force, 1=Force Bit 0: Blue channeli Bit 1: Green channel Bit 2: Red channel
TMDSA_FORCE_DATA_ON_BLANKb_ONLY	16	0x0	Data is forced only during active region. 0=Disable 1=Enable

Data Force Control

TMDSA_FORCE_DATA - RW - 32 bits - [GpuF0MMReg:0x7890]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_FORCE_DATA	7:0	0x0	8 bit Data put on TMDS output data channels accordinging to TMDSA_FORCE_DATA_SEL when Force feature enabled (TMDSA_FORCE_DATA_EN = 1)

TMDSA_BIT_DEPTH_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7894]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_TRUNCATE_EN	0	0x0	Enable bit reduction by truncation 0=Disable 1=Enable
TMDSA_TRUNCATE_DEPTH	4	0x0	Controls bits per pixel 0=18bpp 1=24bpp
TMDSA_SPATIAL_DITHER_EN	8	0x0	Enable bit reduction by spatial (random) dither 0=Disable 1=Enable
TMDSA_SPATIAL_DITHER_MODE	10:9	0x0	LFSR seed selection. 0: Seed pattern A(a,a), 1: seed pattern A(a, ~a), 2: seed pattern AABB(a, ~a, b, ~b), 3: seed pattern AABBCC(a, ~a, b, ~b, c, ~c)
TMDSA_SPATIAL_DITHER_DEPTH	12	0x0	Controls bits per pixel 0=18bpp 1=24bpp
TMDSA_FRAME_RANDOM_ENABLE	13	0x0	Control the LFSR reset, every frame or once at startup 0=0: RGB LFSR are reset every frame, 1: reset once at startup/no reset on every frame
TMDSA_RGB_RANDOM_ENABLE	14	0x0	Control the pseudo-random number to be dithered on RGB 0=0: RGB use x^28+x^3+1 random number 1=1: R dithered with x^28+x^3+1, G dithered with x^28+x^9+1 and B dithered with x^28+x^13+1
TMDSA_HIGHPASS_RANDOM_ENABLE	15	0x0	Highpass filter on RGB dithered channels 0=0: highpass filter is disable, 1: highpass filter is enable on RGB
TMDSA_TEMPORAL_DITHER_EN	16	0x0	Enable bit reduction by temporal dither (frame mod.) 0=Disable 1=Enable
TMDSA_TEMPORAL_DITHER_DEPTH	20	0x0	Controls bits per pixel 0=18bpp 1=24bpp
TMDSA_TEMPORAL_DITHER_OFFSET	22:21	0x0	Add offset to RGB channel before temporal dithering operation 0=For 24bpp: add offset[1:0] to RGB channels 1=For 18bpp: Add offset[1:0]x4 to RGB channels
TMDSA_TEMPORAL_LEVEL	24	0x0	Gray level select (2 or 4 levels) 0=Gray level 2(1 bit - LSB) 1=Gray level 4(2 bits - 2 LSBs)
TMDSA_TEMPORAL_DITHER_RESET	25	0x0	Reset temporal dither (frame modulation) 0=Temporal Dither Ready 1=Reset Temporal Dither Circuit

Control the method in which the data input into the TMDS block is reduced and the length it is reduced to.

TMDSA_CONTROL_CHAR - RW - 32 bits - [GpuF0MMReg:0x7898]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CONTROL_CHAR0_OUT_EN	0	0x0	Programmable sync character 0 enable
TMDSA_CONTROL_CHAR1_OUT_EN	1	0x0	Programmable sync character 1 enable
TMDSA_CONTROL_CHAR2_OUT_EN	2	0x0	Programmable sync character 2 enable
TMDSA_CONTROL_CHAR3_OUT_EN	3	0x0	Programmable sync character 3 enable

SYNC Character Enable. Each bit represents the use of register defined sync character.

TMDSA_CONTROL0_FEEDBACK - RW - 32 bits - [GpuF0MMReg:0x789C]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CONTROL0_FEEDBACK_SELECT	1:0	0x0	Select input of CTL0 for TMDSA
TMDSA_CONTROL0_FEEDBACK_DELAY	9:8	0x0	Select delay of CTL0 for TMDSA

TMDSA_STEREOSYNC_CTL_SEL - RW - 32 bits - [GpuF0MMReg:0x78A0]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_STEREOSYNC_CTL_SEL	1:0	0x0	Controls which CTL signal STEREOSYNC goes on to 0=TMDS CTL registers have normal functionality 1=Stereosync will use TMDS CTL1 register 2=Stereosync will use TMDS CTL2 register 3=Stereosync will use TMDS CTL3 register

TMDSA_SYNC_CHAR_PATTERN_SEL - RW - 32 bits - [GpuF0MMReg:0x78A4]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_SYNC_CHAR_PATTERN_SEL Not Currently Connected	3:0	0x0	Reserved

TMDSA_SYNC_CHAR_PATTERN_0_1 - RW - 32 bits - [GpuF0MMReg:0x78A8]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_SYNC_CHAR_PATTERN0	9:0	0x0	TMDSA SYNC character set 0
TMDSA_SYNC_CHAR_PATTERN1	25:16	0x0	TMDSA SYNC character set 1

TMDSA_SYNC_CHAR_PATTERN_2_3 - RW - 32 bits - [GpuF0MMReg:0x78AC]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_SYNC_CHAR_PATTERN2	9:0	0x0	TMDSA SYNC character set 2
TMDSA_SYNC_CHAR_PATTERN3	25:16	0x0	TMDSA SYNC character set 3

TMDSA_CRC_CNTL - RW - 32 bits - [GpuF0MMReg:0x78B0]			
Field Name	**Bits**	**Default**	**Description**

TMDSA_CRC_EN	0	0x0	Enable TMDSA primary CRC calculation 0=Disable 1=Enable
TMDSA_CRC_CONT_EN	4	0x0	Select continuous or one-shot mode for primary CRC 0=CRC is calculated over 1 frame 1=CRC is continuously calculated for every frame
TMDSA_CRC_ONLY_BLANKb	8	0x0	Determines whether primary CRC is calculated for the whole frame or only during non-blank period. 0=CRC calculated over entire field 1=CRC calculated only during BLANKb
TMDSA_CRC_FIELD	12	0x0	Controls which field polarity starts the TMDSA CRC block after TMDSA_CRC_EN is set to 1. Used only for interlaced mode CRCs 0=Even field begins CRC calculation 1=Odd field begins CRC calculation
TMDSA_2ND_CRC_EN	16	0x0	Enable TMDSA 2nd CRC calculation 0=Disable 1=Enable
TMDSA_2ND_CRC_LINK_SEL	20	0x0	Select which TMDS link to perform CRC on. 0=Perform CRC on link0 1=Perform CRC on link1
TMDSA_2ND_CRC_DATA_SEL	25:24	0x1	Select whether to perform CRC on all data or a subset of the video frame. 0=2ND CRC calculated over entire field 1=2ND CRC calculated only during video data enable (plus preamble and guard band in HDMI mode) 2=2ND CRC calculated over vertical blank region, including VBI preamble and guard band region, excluding horizontal blank 3=2ND CRC calculated only during audio data enable

Enable TMDSA CRC Calculation

TMDSA_CRC_SIG_MASK - RW - 32 bits - [GpuF0MMReg:0x78B4]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CRC_SIG_BLUE_MASK	7:0	0xff	CRC mask bits for TMDSA blue component
TMDSA_CRC_SIG_GREEN_MASK	15:8	0xff	CRC mask bits for TMDSA green component
TMDSA_CRC_SIG_RED_MASK	23:16	0xff	CRC mask bits for TMDSA red component
TMDSA_CRC_SIG_CONTROL_MASK	26:24	0x7	CRC mask bits for TMDSA control signals 3-bit input value: bit 2 = Vsync bit 1 = Hsync bit 0 =Data Enable

RGB and Control CRC Mask

TMDSA_CRC_SIG_RGB - RW - 32 bits - [GpuF0MMReg:0x78B8]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CRC_SIG_BLUE (R)	7:0	0x0	CRC signature value for TMDSA blue component
TMDSA_CRC_SIG_GREEN (R)	15:8	0x0	CRC signature value for TMDSA green component
TMDSA_CRC_SIG_RED (R)	23:16	0x0	CRC signature value for TMDSA red component
TMDSA_CRC_SIG_CONTROL (R)	26:24	0x0	CRC signature value for TMDSA control signals3-bit input value: bit 2 = Vsync bit 1 = Hsync bit 0 =Data Enable

RGB and Control CRC Result

TMDSA_2ND_CRC_RESULT - RW - 32 bits - [GpuF0MMReg:0x78BC]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_2ND_CRC_RESULT (R)	29:0	0x0	Secondary TMDS CRC Result

TMDSA_TEST_PATTERN - RW - 32 bits - [GpuF0MMReg:0x78C0]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_TEST_PATTERN_OUT_EN	0	0x0	Controls the TMDSA output test pattern 0=Normal functionality determined by value of TMDSA_RANDOM_PATTERN_OUT_EN register 1=Test pattern output mode. The value of TMDSA_HALF_CLOCK_PATTERN_SEL determines whether a static 10-bit test data pattern or an alternating half-clock pattern will be output.
TMDSA_HALF_CLOCK_PATTERN_SEL	1	0x0	Controls between static pattern output and alternating static pattern output 0=10 bit test pattern from TMDSA_STATIC_TEST_PATTERN is sent for TMDS output during every pixel clock 1=Alternating pattern of TMDSA_STATIC_TEST_PATTERN and !(TMDSA_STATIC_TEST_PATTERN) on each subsequent pixel clock cycle is sent during every pixel clock
TMDSA_RANDOM_PATTERN_OUT_EN	4	0x0	Enable for random pattern output 0=Normal 1=TMDS Random Pixel Data Generator circuit generates 24-bit pixel data to be encoded and transmitted
TMDSA_RANDOM_PATTERN_RESET	5	0x1	Reset random pattern to pattern seed 0=Enable Random Pixel Data Generator 1=Random Pixel Data Generator is Reset to the value in TMDSA_RANDOM_PATTERN_SEED
TMDSA_TEST_PATTERN_EXTERNAL_RESET_EN	6	0x1	0: Normal 1: Hold non-static test pattern (random, half clock) in reset when external signal is asserted 0=Normal 1=External signal resets random and half clock patterns
TMDSA_STATIC_TEST_PATTERN	25:16	0x0	TMDSA test pixel. Replace the pixel value when TMDSA_TEST_PATTERN_OUT_EN=1

TMDSA_RANDOM_PATTERN_SEED - RW - 32 bits - [GpuF0MMReg:0x78C4]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_RANDOM_PATTERN_SEED	23:0	0x222222	Initial pattern for eye pattern measurement

TMDSA_RAN_PAT_DURING_DE_ONLY	24	0x0	Controls between random pattern out during entire field and DE 0=TMDS Random Data Pattern is output for all pixels 1=TMDS Random Data Pattern is only output when DE is high

TMDSA_DEBUG - RW - 32 bits - [GpuF0MMReg:0x78C8]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_DEBUG_EN	0	0x0	Set to 1 to enable debug mode
TMDSA_DEBUG_HSYNC	8	0x0	Debug mode HSYNC
TMDSA_DEBUG_HSYNC_EN	9	0x0	Set to 1 to enable debug mode HSYNC
TMDSA_DEBUG_VSYNC	16	0x0	Debug mode VSYNC
TMDSA_DEBUG_VSYNC_EN	17	0x0	Set to 1 to enable debug mode VSYNC
TMDSA_DEBUG_DE	24	0x0	Debug mode display enable
TMDSA_DEBUG_DE_EN	25	0x0	Set to 1 to enable debug mode display enable

TMDSA_CTL_BITS - RW - 32 bits - [GpuF0MMReg:0x78CC]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CTL0	0	0x0	Control signal for TMDSA (encoded in Green channel).
TMDSA_CTL1	8	0x0	Control signal for TMDSA (encoded in Green channel).
TMDSA_CTL2	16	0x0	Control signal for TMDSA (encoded in Red channel).
TMDSA_CTL3	24	0x0	Control signal for TMDSA (encoded in Red channel).

TMDSA_DCBALANCER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x78D0]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_DCBALANCER_EN	0	0x1	DC Balancer Enable 0=Disable 1=Enable
TMDSA_DCBALANCER_TEST_EN	8	0x0	DC Balancer Test Enable
TMDSA_DCBALANCER_TEST_IN	19:16	0x0	DC Balancer Test Input
TMDSA_DCBALANCER_FORCE	24	0x0	DC Balancer select value to use when DCBALANCER_EN=0

TMDSA_RED_BLUE_SWITCH - RW - 32 bits - [GpuF0MMReg:0x78D4]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_RB_SWITCH_EN	0	0x0	Switch Red and Blue encoding position. 0=Disable 1=Enable

TMDSA_DATA_SYNCHRONIZATION - RW - 32 bits - [GpuF0MMReg:0x78DC]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_DSYNSEL	0	0x1	Data synchronization circuit select enable 0=Disable 1=Enable
TMDSA_PFREQCHG (W)	8	0x0	Write to 1 to restarts read and write address generation logic. Write of 0 has no effect. Read value is always 0. PFREQCHG must be written to 1 when the data synchronizer is started by setting DSYNSEL to 1, TMDSA_DUAL_LINK_ENABLE is reprogrammed, or either PCLK_TMDSA or PCLK_TMDSA_DIRECT (IDCLK) is reprogrammed or stopped and restarted.

TMDSA Data Sychronization Control

TMDSA_CTL0_1_GEN_CNTL - RW - 32 bits - [GpuF0MMReg:0x78E0]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CTL0_DATA_SEL	3:0	0x0	Select data to be used to generate CTL0 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Random data
TMDSA_CTL0_DATA_DELAY	6:4	0x0	Number of pixel clocks to delay CTL0 data 0=CTL0 data is delayed 0 pixel clocks 1=CTL0 data is delayed 1 pixel clocks 2=CTL0 data is delayed 2 pixel clocks 3=CTL0 data is delayed 3 pixel clocks 4=CTL0 data is delayed 4 pixel clocks 5=CTL0 data is delayed 5 pixel clocks 6=CTL0 data is delayed 6 pixel clocks 7=CTL0 data is delayed 7 pixel clocks
TMDSA_CTL0_DATA_INVERT	7	0x0	Set to 1 to invert CTL0 data 0=CTL0 data is normal 1=CTL0 data is inverted
TMDSA_CTL0_DATA_MODULATION	9:8	0x0	CTL0 data modulation control 0=CTL0 data is not modulated 1=CTL0 data is modulated by bit 0 of 2 bit counter 2=CTL0 data is modulated by bit 1 of 2 bit counter 3=CTL0 data is modulated every time 2 bit counter overflows
TMDSA_CTL0_USE_FEEDBACK_PATH	10	0x0	Set to 1 to enable CTL0 internal feedback path
TMDSA_CTL0_FB_SYNC_CONT	11	0x0	Set to 1 to force continunous toggle on CTL0 internal feedback path
TMDSA_CTL0_PATTERN_OUT_EN	12	0x0	Select CTL0 output data 0=Register value 1=Pattern generator output
TMDSA_CTL1_DATA_SEL	19:16	0x0	Select data to be used to generate CTL1 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Always (blank time)

TMDSA_CTL1_DATA_DELAY	22:20	0x0	Number of pixel clocks to delay CTL1 data 0=CTL1 data is delayed 0 pixel clocks 1=CTL1 data is delayed 1 pixel clocks 2=CTL1 data is delayed 2 pixel clocks 3=CTL1 data is delayed 3 pixel clocks 4=CTL1 data is delayed 4 pixel clocks 5=CTL1 data is delayed 5 pixel clocks 6=CTL1 data is delayed 6 pixel clocks 7=CTL1 data is delayed 7 pixel clocks
TMDSA_CTL1_DATA_INVERT	23	0x0	Set to 1 to invert CTL1 data 0=CTL1 data is normal 1=CTL1 data is inverted
TMDSA_CTL1_DATA_MODULATION	25:24	0x0	CTL1 data modulation control 0=CTL1 data is not modulated 1=CTL1 data is modulated by bit 0 of 2 bit counter 2=CTL1 data is modulated by bit 1 of 2 bit counter 3=CTL1 data is modulated every time 2 bit counter overflows
TMDSA_CTL1_USE_FEEDBACK_PATH	26	0x0	Set to 1 to enable CTL1 internal feedback path
TMDSA_CTL1_FB_SYNC_CONT	27	0x0	Set to 1 to force continunous toggle on CTL1 internal feedback path
TMDSA_CTL1_PATTERN_OUT_EN	28	0x0	Select CTL1 output data 0=Register value 1=Pattern generator output
TMDSA_2BIT_COUNTER_EN	31	0x0	Set to 1 to enable 2-bit data modulation counter 0=Disable 1=Enable

TMDSA_CTL2_3_GEN_CNTL - RW - 32 bits - [GpuF0MMReg:0x78E4]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_CTL2_DATA_SEL	3:0	0x0	Select data to be used to generate CTL2 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Always (blank time)
TMDSA_CTL2_DATA_DELAY	6:4	0x0	Number of pixel clocks to delay CTL2 data 0=CTL2 data is delayed 0 pixel clocks 1=CTL2 data is delayed 1 pixel clocks 2=CTL2 data is delayed 2 pixel clocks 3=CTL2 data is delayed 3 pixel clocks 4=CTL2 data is delayed 4 pixel clocks 5=CTL2 data is delayed 5 pixel clocks 6=CTL2 data is delayed 6 pixel clocks 7=CTL2 data is delayed 7 pixel clocks
TMDSA_CTL2_DATA_INVERT	7	0x0	Set to 1 to invert CTL2 data 0=CTL2 data is normal 1=CTL2 data is inverted
TMDSA_CTL2_DATA_MODULATION	9:8	0x0	CTL2 data modulation control 0=CTL2 data is not modulated 1=CTL2 data is modulated by bit 0 of 2 bit counter 2=CTL2 data is modulated by bit 1 of 2 bit counter 3=CTL2 data is modulated every time 2 bit counter overflows
TMDSA_CTL2_USE_FEEDBACK_PATH	10	0x0	Set to 1 to enable CTL2 internal feedback path

TMDSA_CTL2_FB_SYNC_CONT	11	0x0	Set to 1 to force continunous toggle on CTL2 internal feedback path
TMDSA_CTL2_PATTERN_OUT_EN	12	0x0	Select CTL2 output data 0=Register value 1=Pattern generator output
TMDSA_CTL3_DATA_SEL	19:16	0x0	Select data to be used to generate CTL3 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Always (blank time)
TMDSA_CTL3_DATA_DELAY	22:20	0x0	Number of pixel clocks to delay CTL3 data 0=CTL3 data is delayed 0 pixel clocks 1=CTL3 data is delayed 1 pixel clocks 2=CTL3 data is delayed 2 pixel clocks 3=CTL3 data is delayed 3 pixel clocks 4=CTL3 data is delayed 4 pixel clocks 5=CTL3 data is delayed 5 pixel clocks 6=CTL3 data is delayed 6 pixel clocks 7=CTL3 data is delayed 7 pixel clocks
TMDSA_CTL3_DATA_INVERT	23	0x0	Set to 1 to invert CTL3 data 0=CTL3 data is normal 1=CTL3 data is inverted
TMDSA_CTL3_DATA_MODULATION	25:24	0x0	CTL3 data modulation control 0=CTL3 data is not modulated 1=CTL3 data is modulated by bit 0 of 2 bit counter 2=CTL3 data is modulated by bit 1 of 2 bit counter 3=CTL3 data is modulated every time 2 bit counter overflows
TMDSA_CTL3_USE_FEEDBACK_PATH	26	0x0	Set to 1 to enable CTL3 internal feedback path
TMDSA_CTL3_FB_SYNC_CONT	27	0x0	Set to 1 to force continunous toggle on CTL3 internal feedback path
TMDSA_CTL3_PATTERN_OUT_EN	28	0x0	Select CTL3 output data 0=Register value 1=Pattern generator output

TMDSA_TRANSMITTER_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7904]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_TX0_ENABLE	0	0x0	TMDSA link0 enable (ILNK0EN)(set to 1 whenever TMDS is enabled) 0=Disable 1=Enable
TMDSA_LNKC0EN	1	0x0	TMDSA clock channel enable (ICHCEN)(set to 1 whenever TMDS is enabled)
TMDSA_LNKD00EN	2	0x0	TMDSA link0 data channel 0 enable (ICHD0EN)(set to 1 whenever TMDS is enabled)
TMDSA_LNKD01EN	3	0x0	TMDSA link0 data channel 1 enable (ICHD1EN)(set to 1 whenever TMDS is enabled)
TMDSA_LNKD02EN	4	0x0	TMDSA link0 data channel 2 enable (ICHD2EN)(set to 1 whenever TMDS is enabled)
TMDSA_TX1_ENABLE	8	0x0	TMDSA link1 enable (ILNKD1EN)(set to 1 whenever TMDS is enabled in dual-link mode)
TMDSA_LNKC1EN	9	0x0	
TMDSA_LNKD10EN	10	0x0	TMDSA link1 data channel 0 enable (ICHD3EN)(set to 1 whenever TMDS is enabled in dual-link mode)
TMDSA_LNKD11EN	11	0x0	TMDSA link1 data channel 1 enable (ICHD4EN)(set to 1 whenever TMDS is enabled in dual-link mode)

TMDSA_LNKD12EN	12	0x0	TMDSA link1 data channel 2 enable (ICHD5EN)(set to 1 whenever TMDS is enabled in dual-link mode)
TMDSA_TX_ENABLE_HPD_MASK	16	0x0	0:Disallow 1:Allow override of TMDSA_TXX_ENABLE by HPD on disconnect 0=Result from HPD circuit can not override TMDSA_TXX_ENABLE 1=Result from HPD circuit can override TMDSA_TXX_ENABLE on disconnect
TMDSA_LNKCEN_HPD_MASK	17	0x0	0:Disallow 1:Allow override of TMDSA_LNKCXEN by HPD on disconnect 0=Result from HPD circuit can not override TMDSA_LNKC0EN 1=Result from HPD circuit overrides TMDSA_LNKC0EN on disconnect
TMDSA_LNKDEN_HPD_MASK	18	0x0	0:Disallow 1:Allow override of TMDSA_LNKDXEN by HPD on disconnect 0=Result from HPD circuit can not override TMDSA_LNKDXEN 1=Result from HPD circuit overrides TMDSA_LNKDXEN on disconnect

TMDSA_LOAD_DETECT - RW - 32 bits - [GpuF0MMReg:0x7908]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_LOAD_DETECT_ENABLE	0	0x1	0: Disable 1: Enable TMDSA macro load detect functionDrives IMSEN macro input
TMDSA_LOAD_DETECT (R)	4	0x0	From TMDSA macro OMSEN output 0: No load detected 1: Load detected

TMDSA_PLL_ADJUST - RW - 32 bits - [GpuF0MMReg:0x790C]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_PLL_CP_GAIN	5:0	0xb	tmds macro channel A&B charge pump gain control
TMDSA_PLL_VCO_GAIN	13:8	0x7	tmds macro channel A&B vco control
TMDSA_PLL_DUTY_CYCLE	17:16	0x0	tmds macro channel A&B clock duty cycle control

TMDSA_TRANSMITTER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7910]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_PLL_ENABLE	0	0x0	TMDSA transmitter's PLL enable. This can power down the PLL. 0=Disable 1=Enable

TMDSA_PLL_RESET	1	0x1	TMDSA transmitter's PLL reset. PLL will start the locking acquisition process once this becomes low.
TMDSA_PLL_ENABLE_HPD_MASK	3:2	0x0	Determines whether result from HPD circuit can override TMDSA_PLL_ENABLE and TMDSA_PLL_RESET Bit 0: Set to 1 to enable override on disconnect Bit 1: Set to 1 to enable override on connect. 0=Result from HPD circuit can not override TMDSA_PLL_ENABLE 1=Result from HPD circuit overrides TMDSA_PLL_ENABLE on disconnect 2=Result from HPD circuit overrides TMDSA_PLL_ENABLE on connect 3=Result from HPD circuit overrides TMDSA_PLL_ENABLE
TMDSA_IDSCKSELA	4	0x1	Select TMDSA tramsmitter A to use IPIXCLK or IDCLK 0=TMDS Transmitter A uses pclk_tmdsa (IPIXCLK) 1=TMDS Transmitter A uses pclk_tmdsa_direct (IDCLK)
TMDSA_IDSCKSELB	5	0x1	Select TMDSA tramsmitter B to use IPIXCLK or IDCLK. This bit applies only to TMDS dual link or/and TMDS SWAP mode. 0=TMDS Transmitter B uses pclk_tmdsa (IPIXCLK) 1=TMDS Transmitter B uses pclk_tmdsa_direct (IDCLK)
TMDSA_PLL_PWRUP_SEQ_EN	6	0x0	Enable hardware delay of PLL enable / reset on power up / down to match macro timing requirements. When TMDSA_PLL_PWRUP_SEQ_EN=1, PLL will be reset 1 us before PLL enable is deasserted, and PLL reset will be asserted for 10 us after PLL enable is asserted. This timing is provided to match the TMDS macro timing specification. 0=Disabled 1=Delay Enable/ Reset for clean PLL power up/down
TMDSA_PLL_RESET_HPD_MASK	7	0x0	Set this bit to automatically reset TMDS macro PLL on HPD connect event, reset pulse= 10us 0=TMDS macro pll is not reset on hot plug detect connect 1=TMDS macro pll reset on hot plug detect for 10 us
TMDSA_TMCLK	12:8	0x0	For macro debug only
TMDSA_TMCLK_FROM_PADS	13	0x0	Controls input to ITMCLK pin on macro for macro debug only 0=Input to ITMCLK pins on macro come from TMDSA_TMCLK field 1=Input to ITMCLK pins on macro come from pads
TMDSA_TDCLK	14	0x0	For macro debug only
TMDSA_TDCLK_FROM_PADS	15	0x0	Controls input to ITDCLK pin on macro for macro debug only 0=Input to ITDCLK pin on macro comes from TMDSA_TDCLK field 1=Input to ITDCLK pin on macro comes from pads
TMDSA_PLLSEL_OVERWRITE_EN	16	0x0	Enable overwrite of TMDSA_PLSELA & TMDSA_PLSELA, because normally harware automatically set TMDSA_PLLSELA & TMDSA_PLLSELB bits 0=Hardware automatically selects PLLSELA and PLLSELB based on diff. modes. OVERWRITE is disabled 1=Overwrite hardware pll selection in TMDSA_PLSELA & TMDSA_PLSELB fiels. Enable software overwrite.
TMDSA_PLLSELA	17	0x0	Normally, this bit is automatically selected in hardware. Effective only when TMDSA_PLLSEL_OVERWRITE_EN is 1
TMDSA_PLLSELB	18	0x0	Normally, this bit is automatically selected in hardware. Effective only when TMDSA_PLLSEL_OVERWRITE_EN is 1

TMDSA_BYPASS_PLLA	28	0x1	Controls ICHCSELA pin on TMDSA macro 0: Coherent mode: transmitted A clock is PLL output 1: Incoherent mode: transmitted A clock is PLL input 0=0: TMDS transmitter A is in coherent mode 1=1: Tmds transmitter A is in incoherent mode
TMDSA_BYPASS_PLLB	29	0x1	Controls ICHCSELA pin on TMDSA macro 0: Coherent mode: transmitted B clock is PLL output 1: Incoherent mode: transmitted B clock is PLL input 0=0: TMDS transmitter B is in coherent mode 1=1: Tmds transmitter B is in incoherent mode
TMDSA_INPUT_TEST_CLK_SEL1	30	0x0	Controls ITCLKSEL1 pin on TMDSA macro
TMDSA_INPUT_TEST_CLK_SEL2	31	0x0	Controls ITCLKSEL2 pin on TMDSA macro

TMDSA_REG_TEST_OUTPUTA - RW - 32 bits - [GpuF0MMReg:0x7914]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_REG_TEST_OUTPUTA (R)	9:0	0x0	Outputs of the 10 shift registers (OTDATX[9:0]) from one of the channels during test mode.
TMDSA_TEST_CNTLA	17:16	0x0	Selects which of 3 register test output channels from TMDSA macro is visible in TMDSA_REG_TEST_OUTPUTA. 0=OTDATA0 1=OTDATA1 2=OTDATA2 3=N/A
TMDSA_TEST_OUTPUT_SELECT	20	0x0	ENABLE TEST_OUTPUTA & TEST_OUTPUTB

TMDSA_TRANSMITTER_DEBUG - RW - 32 bits - [GpuF0MMReg:0x7918]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_PLL_DEBUG	7:0	0x0	Drives ITPL pins on TMDSA macro
TMDSA_TX_DEBUG	15:8	0x0	Drives ITX pins on TMDSA macro

Reserved for debugging purposes

TMDSA_DITHER_RAND_SEED - RW - 32 bits - [GpuF0MMReg:0x791C]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_RAND_R_SEED	7:0	0x0	Seed for random red, the random seed is 1'b1, TMDSA_RAND_R_SEED[2:0], 3TMDSA_RAND_R_SEED = 28 bits
TMDSA_RAND_G_SEED	15:8	0x99	Seed for random green, the random seed is 1'b1, TMDSA_RAND_G_SEED[2:0], 3TMDSA_RAND_G_SEED = 28 bits
TMDSA_RAND_B_SEED	23:16	0xdd	Seed for random bleu, the random seed is 1'b1, TMDSA_RAND_B_SEED[2:0], 3TMDSA_RAND_B_SEED = 28 bits

programmable seed for random dithering

TMDSA_TRANSMITTER_ADJUST - RW - 32 bits - [GpuF0MMReg:0x7920]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_TX_VOLTAGE_SWING_A	3:0	0xa	tmds macro transmitter A, voltage swing control
TMDSA_TX_VOLTAGE_SWING_B	7:4	0xa	tmds macro transmitter B, voltage swing control
TMDSA_TXPCA	9:8	0x0	tmds macro transmitter A, pulse current control
TMDSA_TXPCB	13:12	0x0	tmds macro transmitter B, pulse current control
TMDSA_TXPWA	17:16	0x0	tmds macro transmitter A, pulse width control
TMDSA_TXPWB	21:20	0x0	tmds macro transmitter A, pulse width control
TMDSA_TX_VS_COMPA	25:24	0x0	tmds macro transmitter A, voltage swing compensation control
TMDSA_TX_VS_COMPB	29:28	0x0	tmds macro transmitter B, voltage swing compensation control

TMDSA_REG_TEST_OUTPUTB - RW - 32 bits - [GpuF0MMReg:0x7924]			
Field Name	**Bits**	**Default**	**Description**
TMDSA_REG_TEST_OUTPUTB (R)	9:0	0x0	Outputs of the 10 shift registers (OTDATX[9:0]) from one of the channels during test mode.
TMDSA_TEST_CNTLB	17:16	0x0	Selects which of 3 register test output channels from TMDSA macro is visible in TMDSA_REG_TEST_OUTPUTB. 0=OTDATB0 1=OTDATB1 2=OTDATB2 3=N/A

Digital Video Output (DVO) Registers

DVOA_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7980]			
Field Name	**Bits**	**Default**	**Description**
DVOA_ENABLE	0	0x0	Enable for DVO 0=Disable 1=Enable
DVOA_PIXEL_ENCODING	8	0x0	Selects pixel encoding format 0=RGB 4:4:4 or YCBCR 4:4:4 1=YCbCr 4:2:2

DVOA_SOURCE_SELECT - RW - 32 bits - [GpuF0MMReg:0x7984]			
Field Name	**Bits**	**Default**	**Description**
DVOA_SOURCE_SELECT	0	0x0	Select between 1st and 2nd display streams 0=CRTC1 data is used 1=CRTC2 data is used

DVOA_SYNC_SELECT	8	0x0	Select between SYNCA and SYNCB signals from CRTC 0=HSYNC_A & VSYNC_A from the selected CRTC are used 1=HSYNC_B & VSYNC_B from the selected CRTC are used
DVOA_STEREOSYNC_SELECT	16	0x0	Select between CRTC1 and CRTC2 stereosync signals 0=DVOA Stereosync from CRTC1 used 1=DVOA Stereosync from CRTC2 used

Source Select control for Data, H/VSYNC & Stereosync

DVOA_BIT_DEPTH_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7988]			
Field Name	**Bits**	**Default**	**Description**
DVOA_TRUNCATE_EN	0	0x0	Enable bit reduction by truncation 0=Disable 1=Enable
DVOA_TRUNCATE_DEPTH	4	0x0	Select truncation depth 0=18bpp 1=24bpp
DVOA_SPATIAL_DITHER_EN	8	0x0	Enable bit reduction by spatial (random) dither 0=Disable 1=Enable
DVOA_SPATIAL_DITHER_MODE	10:9	0x0	LFSR seed selection. 0: Seed pattern A(a,a), 1: seed pattern A(a, ~a), 2: seed pattern AABB(a, ~a, b, ~b), 3: seed pattern AABBCC(a, ~a, b, ~b, c, ~c)
DVOA_SPATIAL_DITHER_DEPTH	12	0x0	Select spatial dither depth 0=18bpp 1=24bpp
DVOA_FRAME_RANDOM_ENABLE	13	0x0	Control the LFSR reset, every frame or once at startup 0=0: RGB LFSR are reset every frame, 1: reset once at startup/no reset on every frame
DVOA_RGB_RANDOM_ENABLE	14	0x0	Control the pseudo-random number to be dithered on RGB 0=0: RGB use x^28+x^3+1 random number 1=1: R dithered with x^28+x^3+1, G dithered with x^28+x^9+1 and B dithered with x^28+x^13+1
DVOA_HIGHPASS_RANDOM_ENABLE	15	0x0	Highpass filter on RGB dithered channels 0=0: highpass filter is disable, 1: highpass filter is enable on RGB
DVOA_TEMPORAL_DITHER_EN	16	0x0	Enable bit reduction by temporal dither (frame mod.) 0=Disable 1=Enable
DVOA_TEMPORAL_DITHER_DEPTH	20	0x0	Select temporal dither depth 0=18bpp 1=24bpp
DVOA_TEMPORAL_DITHER_OFFSET	22:21	0x0	Add offset to RGB channel before temporal dithering operation 0=For 24bpp: add offset[1:0] to RGB channels 1=For 18bpp: Add offset[1:0]x4 to RGB channels
DVOA_TEMPORAL_LEVEL	24	0x0	Gray level select (2 or 4 levels) 0=Gray level 2 1=Gray level 4
DVOA_TEMPORAL_DITHER_RESET	25	0x0	Reset temporal dither (frame modulation) 0=Temporal Dither Ready 1=Reset Temporal Dither Circuit

Control the method in which the data input into the DVO block is reduced and the length it is reduced to.

DVOA_OUTPUT - RW - 32 bits - [GpuF0MMReg:0x798C]			
Field Name	**Bits**	**Default**	**Description**
DVOA_OUTPUT_ENABLE_MODE	1:0	0x0	Ouput mode for DVO 0=disabled 1=lower 12 output en 2=upper 12 output en 3=all 24 output enable
DVOA_CLOCK_MODE	8	0x0	Reserved 0=differential clocking enabled 1=single ended clocking enabled

Output enable control for DVO pads.

DVOA_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7990]			
Field Name	**Bits**	**Default**	**Description**
DVOA_RATE_SELECT	0	0x0	Select between DDR and SDR modes 0=DDR Speed 1=SDR Speed using NPL A pin
DVOA_SDRCLK_SEL	1	0x0	Select SDR DVOCLK using clk from A pin or CLK0 pin in NPL 0=use NPL A input clock 1=use CLK0 input clock
DVOA_DUAL_CHANNEL_EN	8	0x0	Enable Dual Channel DVO Mode 0=Disable 1=Enable
DVOA_RESET_FIFO (W)	16	0x0	Write 1 to force reset of DVO mesochronous fifo
DVOA_SYNC_PHASE	17	0x1	Determine whether to reset phase signal on frame pulse 0=Disable 1=Enable
DVOA_INVERT_DVOCLK	18	0x0	Set to 1 to invert 'clock' going to d input of dvoclk pad 0=Don't Invert 1=Invert
DVOA_COLOR_FORMAT	25:24	0x0	0=8-bit DVO display 1=Twin Single Link 10-bit mode 2=Dual-Link 10-bit mode 3=Reserved
DVOA_REORDER_BITS	28	0x0	Reorder DVO bits output = input 1:0, input 7:2 0=Disable 1=Enable

DVOA_CRC_EN - RW - 32 bits - [GpuF0MMReg:0x7994]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC_EN	0	0x0	Enable DVO CRC 0=Disable 1=Enable
DVOA_CRC_CONT_EN	8	0x0	Select between one shot and continous mode 0=CRC is calculated over 1 frame 1=CRC is continuously calculated for every frame
DVOA_CRC2_EN	16	0x0	Enable DVO output CRC2 0=Disable 1=Enable

DVOA_CRC_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7998]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC_FIELD	0	0x0	Controls which field polarity starts the DVO CRC block after DAC_CRC_EN is set high 0=Even field begins CRC calculation 1=Odd field begins CRC calculation
DVOA_CRC_ONLY_BLANKb	8	0x0	Determines whether CRC is calculated for the whole frame or only during non-blank period for DVO 0=CRC calculated over entire field 1=CRC calculated only during BLANKb

DVOA_CRC_SIG_MASK1 - RW - 32 bits - [GpuF0MMReg:0x799C]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC_SIG_BLUE_MASK	7:0	0xff	Mask bits for DVO B channel CRC.
DVOA_CRC_SIG_GREEN_MASK	23:16	0xff	Mask bits for DVO G channel CRC.

Select which data the CRC calculation is performed on.

DVOA_CRC_SIG_MASK2 - RW - 32 bits - [GpuF0MMReg:0x79A0]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC_SIG_RED_MASK	7:0	0xff	Mask bits for DVO R channel CRC.
DVOA_CRC_SIG_CONTROL_MASK	18:16	0x7	Mask bits for DVO control signal CRC Bit 18: Vsync signal Bit 17: Hsync Signal Bit 16: Data Enable

Select which control signals the CRC calculation is performed on.

DVOA_CRC_SIG_RESULT1 - RW - 32 bits - [GpuF0MMReg:0x79A4]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC_SIG_BLUE (R)	7:0	0x0	CRC signature value for DVO B channel CRC.
DVOA_CRC_SIG_GREEN (R)	23:16	0x0	CRC signature value for DVO G channel CRC.

DVOA Data CRC Results

DVOA_CRC_SIG_RESULT2 - RW - 32 bits - [GpuF0MMReg:0x79A8]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC_SIG_RED (R)	7:0	0x0	CRC signature value for DVO R channel CRC.
DVOA_CRC_SIG_CONTROL (R)	18:16	0x0	CRC signature value for DVO control CRC.

DVOA DATA and Control CRC Results

DVOA_CRC2_SIG_MASK - RW - 32 bits - [GpuF0MMReg:0x79AC]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC2_SIG_MASK	26:0	0x7fffffff	Mask bits for DVO output CRC2 Bit 26: Vsync signal Bit 25: Hsync Signal Bit 24: Data Enable Bit 23-0:DVO Data

Control for secondary DVO CRC

DVOA_CRC2_SIG_RESULT - RW - 32 bits - [GpuF0MMReg:0x79B0]			
Field Name	**Bits**	**Default**	**Description**
DVOA_CRC2_SIG_RESULT (R)	26:0	0x0	CRC2 signature value for DVO output

CRC2 signature value for DVO output

DVOA_STRENGTH_CONTROL - RW - 32 bits - [GpuF0MMReg:0x79B4]			
Field Name	**Bits**	**Default**	**Description**
DVOA_SP	3:0	0x0	Strength of pull-up section of output buffer for DVO signals.
DVOA_SN	7:4	0x6	Strength of pull-down section of output buffer for DVO signals.
DVOACLK_SP	11:8	0x0	Strength of pull-up section of output buffer for DVO clock output.
DVOACLK_SN	15:12	0x6	Strength of pull-down section of output buffer for DVO clock output.
DVOA_SRP	16	0x1	Increases slew rate to pull-up section of output buffer for DVO signals.
DVOA_SRN	17	0x1	Increases slew rate to pull-down section of output buffer for DVO signals.
DVOACLK_SRP	24	0x1	Increases slew rate to pull-up section of output buffer for DVO clock.
DVOACLK_SRN	25	0x1	Increases slew rate to pull-down section of output buffer for DVO clock.
DVOA_LSB_VMODE	28	0x1	This pin controls the DVO I/O pad's internal level shifter voltage Should be set based on pad output voltage (determined by board voltage regulator) This field controls DVODATA[11:0], DVOCNTL and DVOCLK Sense is inverted for BIF debug 0: 1.8V 1: 3.3V
DVOA_MSB_VMODE	29	0x1	This pin controls the DVO I/O pad's internal level shifter voltage Should be set based on pad output voltage (determined by board voltage regulator) This field controls DVODATA[23:12], MVP_DVOCNTL[1:0] Sense is inverted for BIF debug 0: 1.8V 1: 3.3V

DVOA_FORCE_OUTPUT_CNTL - RW - 32 bits - [GpuF0MMReg:0x79B8]			
Field Name	**Bits**	**Default**	**Description**

DVOA_FORCE_DATA_EN	0	0x0	Enable force option on DVOA 0=Disable 1=Enable
DVOA_FORCE_DATA_SEL	10:8	0x0	Select which DVOA channels have data forced0=Don't Force, 1=Force Bit 0: Blue channel Bit 1: Green channel Bit 2: Red channel
DVOA_FORCE_DATA_ON_BLANKb_ONLY	16	0x0	Data is forced only during active region. 0=Disable 1=Enable
DVOA_HDCP_RGB_PASSTHRU_IN_NONEACTIVE	20	0x0	Enable the DVO to let RGB data pass thru in non active data zone when encryption is enable, otherwise RGB data are zero in non active area
DVOA_HDCP_RANDOM_DATA_EN	21	0x0	Enable random data generation (snow) when encryption is required and cipher is not valid.

DVOA Force Data control register

DVOA_FORCE_DATA - RW - 32 bits - [GpuF0MMReg:0x79BC]			
Field Name	**Bits**	**Default**	**Description**
DVOA_FORCE_DATA	7:0	0x0	Data to be forced on R, G & B channels

Data to be forced on R, G & B channels

LVTM Registers

LVTMA_CNTL - RW - 32 bits - [GpuF0MMReg:0x7A80]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_ENABLE	0	0x0	Enable for the reduction/encoding logic 0=Disable 1=Enable
LVTMA_HDMI_EN	2	0x0	Select DVI or HDMI mode 0=DVI 1=HDMI
LVTMA_ENABLE_HPD_MASK	4	0x0	0:Disallow 1:Allow override of LVTMA_ENABLE by HPD on disconnect 0=Result from HPD circuit can not override LVTMA_ENABLE_HPD_MASK 1=Result from HPD circuit can override LVTMA_ENABLE_HPD_MASK on disconnect
LVTMA_HPD_SELECT	9:8	0x0	Select which hot plug detect unit to use for LVTMA. This selection is only relevant if one of the HPD mask bits in this and other other registers is enabled. 0=Use HPD1 1=Use HPD2 2=Use HPD3
LVTMA_SYNC_PHASE	12	0x1	Determine whether to reset phase signal on frame pulse 0: don't reset 1: reset
LVTMA_PIXEL_ENCODING	16	0x0	Pixel encoding format 0=RGB 4:4:4 or YCBCR 4:4:4 1=YCbCr 4:2:2
LVTMA_DUAL_LINK_ENABLE	24	0x0	Enable dual-link 0=Disable 1=Enable

LVTMA_SWAP	28	0x0	Swap upper and lower data channels 0=Disable 1=Enable
LVTMA_SPLIT	29	0x0	Set this bit to allow TMDS macro channel B as an independant link. This link use LVTMA stream 0=Disable 1=Enable
LVTMA_SPLIT_HPD_SELECT	31:30	0x0	Select which hot plug detect unit to use for LVTMA SPLIT. This selection is only relevant if one of the HPD mask bits in this and other other registers is enabled. 0=Use HPD1 1=Use HPD2 2=use HPD3

LVTMA_SOURCE_SELECT - RW - 32 bits - [GpuF0MMReg:0x7A84]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SOURCE_SELECT	0	0x0	Select between display stream 1 & display stream 2 0=CRTC1 data is used 1=CRTC2 data is used
LVTMA_SYNC_SELECT	8	0x0	Select between SYNCA and SYNCB signals 0=HSYNC_A & VSYNC_A from the selected CRTC are used 1=HSYNC_B & VSYNC_B from the selected CRTC are used
LVTMA_STEREOSYNC_SELECT	16	0x0	Select between CRTC1 and CRTC2 stereosync signals 0=CRTC1 STEREOSYNC used 1=CRTC2 STEREOSYNC used

Source Select control for Data, H/VSYNC & Stereosync

LVTMA_COLOR_FORMAT - RW - 32 bits - [GpuF0MMReg:0x7A88]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_COLOR_FORMAT	1:0	0x0	Controls LVTMA output colour format. Formats 0 and 1 work in single or dual link. Format 2 requires dual link (MSBs on primary link, LSBs on secondary link). 0=Normal (24bpp), Twin-Single 30bpp (8 MSBs of each component), or Dual-Link 48bpp 1=Twin-Link 30bpp (2 LSB of each component) 2=Dual-Link 30bpp 3=Reserved

LVTMA_FORCE_OUTPUT_CNTL - RW - 32 bits - [GpuF0MMReg:0x7A8C]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_FORCE_DATA_EN	0	0x0	Enable force option on LVTMA 0=Disable 1=Enable

LVTMA_FORCE_DATA_SEL	10:8	0x0	Select LVTMA channels that have data forced0=Don't Force, 1=Force Bit 0: Blue channel Bit 1: Green channel Bit 2: Red channel
LVTMA_FORCE_DATA_ON_BLANKb_ONLY Data Force Control	16	0x0	Force Data during active region 0=Disable 1=Enable

LVTMA_FORCE_DATA - RW - 32 bits - [GpuF0MMReg:0x7A90]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_FORCE_DATA	7:0	0x0	Data to be forced on R, G & B channels

LVTMA_BIT_DEPTH_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7A94]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_TRUNCATE_EN	0	0x0	Enable bit reduction by truncation 0=Disable 1=Enable
LVTMA_TRUNCATE_DEPTH	4	0x0	Controls truncation depth 0=18bpp 1=24bpp
LVTMA_SPATIAL_DITHER_EN	8	0x0	Enable bit reduction by spatial (random) dither 0=Disable 1=Enable
LVTMA_SPATIAL_DITHER_MODE	10:9	0x0	LFSR seed selection. 0: Seed pattern A(a,a), 1: seed pattern A(a, ~a), 2: seed pattern AABB(a, ~a, b, ~b), 3: seed pattern AABBCC(a, ~a, b, ~b, c, ~c)
LVTMA_SPATIAL_DITHER_DEPTH	12	0x0	Controls reduction depth 0=18bpp 1=24bpp
LVTMA_FRAME_RANDOM_ENABLE	13	0x0	Control the LFSR reset, every frame or once at startup 0=0: RGB LFSR are reset every frame, 1: reset once at startup/no reset on every frame
LVTMA_RGB_RANDOM_ENABLE	14	0x0	Control the pseudo-random number to be dithered on RGB 0=0: RGB use x^28+x^3+1 random number 1=1: R dithered with x^28+x^3+1, G dithered with x^28+x^9+1 and B dithered with x^28+x^13+1
LVTMA_HIGHPASS_RANDOM_ENABLE	15	0x0	Highpass filter on RGB dithered channels 0=0: highpass filter is disable, 1: highpass filter is enable on RGB
LVTMA_TEMPORAL_DITHER_EN	16	0x0	Enable bit reduction by temporal dither (frame mod.) 0=Disable 1=Enable
LVTMA_TEMPORAL_DITHER_DEPTH	20	0x0	Controls dither Depth 0=18bpp 1=24bpp
LVTMA_TEMPORAL_DITHER_OFFSET	22:21	0x0	Add offset to RGB channel before temporal dithering operation 0=For 24bpp: add offset[1:0] to RGB channels 1=For 18bpp: Add offset[1:0]x4 to RGB channels

LVTMA_TEMPORAL_LEVEL	24	0x0	Gray level select (2 or 4 levels) 0=Gray level 2(1 bit - LSB) 1=Gray level 4(2 bits - 2 LSBs)
LVTMA_TEMPORAL_DITHER_RESET	25	0x0	Reset temporal dither (frame modulation) 0=Temporal Dither Ready 1=Reset Temporal Dither Circuit

Control the method in which the data input into the TMDS block is reduced and the length it is reduced to.

LVTMA_CONTROL_CHAR - RW - 32 bits - [GpuF0MMReg:0x7A98]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CONTROL_CHAR0_OUT_EN	0	0x0	Programmable sync character 0 enable
LVTMA_CONTROL_CHAR1_OUT_EN	1	0x0	Programmable sync character 1 enable
LVTMA_CONTROL_CHAR2_OUT_EN	2	0x0	Programmable sync character 2 enable
LVTMA_CONTROL_CHAR3_OUT_EN	3	0x0	Programmable sync character 3 enable

SYNC Character Enable. Each bit represents the use of register defined sync character.

LVTMA_CONTROL0_FEEDBACK - RW - 32 bits - [GpuF0MMReg:0x7A9C]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CONTROL0_FEEDBACK_SELECT	1:0	0x0	Select input of CTL0 for TMDS
LVTMA_CONTROL0_FEEDBACK_DELAY	9:8	0x0	Select delay of CTL0 for TMDS

LVTMA_STEREOSYNC_CTL_SEL - RW - 32 bits - [GpuF0MMReg:0x7AA0]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_STEREOSYNC_CTL_SEL	1:0	0x0	Controls with CTL signal STEREOSYNC goes on to 0=TMDS CTL registers have normal functionality 1=Stereosync will use TMDS CTL1 register 2=Stereosync will use TMDS CTL2 register 3=Stereosync will use TMDS CTL3 register

LVTMA_SYNC_CHAR_PATTERN_SEL - RW - 32 bits - [GpuF0MMReg:0x7AA4]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SYNC_CHAR_PATTERN_SEL	3:0	0x0	Reserved

Not Currently Connected

LVTMA_SYNC_CHAR_PATTERN_0_1 - RW - 32 bits - [GpuF0MMReg:0x7AA8]

Field Name	Bits	Default	Description
LVTMA_SYNC_CHAR_PATTERN0	9:0	0x0	LVTMA SYNC character set 0
LVTMA_SYNC_CHAR_PATTERN1	25:16	0x0	LVTMA SYNC character set 1

LVTMA_SYNC_CHAR_PATTERN_2_3 - RW - 32 bits - [GpuF0MMReg:0x7AAC]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SYNC_CHAR_PATTERN2	9:0	0x0	LVTMA SYNC character set 2
LVTMA_SYNC_CHAR_PATTERN3	25:16	0x0	LVTMA SYNC character set 3

LVTMA_CRC_CNTL - RW - 32 bits - [GpuF0MMReg:0x7AB0]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CRC_EN	0	0x0	Enable LVTMA CRC calculation 0=Disable 1=Enable
LVTMA_CRC_CONT_EN	4	0x0	Select continuous or one-shot mode for primary CRC 0=CRC is calculated over 1 frame 1=CRC is continuously calculated for every frame
LVTMA_CRC_ONLY_BLANKb	8	0x0	Determines whether primary CRC is calculated for the whole frame or only during non-blank period. 0=CRC calculated over entire field 1=CRC calculated only during BLANKb
LVTMA_CRC_FIELD	12	0x0	Controls which field polarity starts the LVTMA CRC block after LVTMA_CRC_EN is set to 1. Used only for interlaced mode CRCs 0=Even field begins CRC calculation 1=Odd field begins CRC calculation
LVTMA_2ND_CRC_EN	16	0x0	Enable LVTMA 2nd CRC calculation 0=Disable 1=Enable
LVTMA_2ND_CRC_LINK_SEL	20	0x0	Select which LVTM link to perform CRC on 0=Perform CRC on link0 1=Perform CRC on link1
LVTMA_2ND_CRC_DATA_SEL	25:24	0x1	Select whether to perform CRC on all data or a subset of the video frame. 0=2ND CRC calculated over entire field 1=2ND CRC calculated only during video data enable (plus preamble and guard band in HDMI mode) 2=2ND CRC calculated over vertical blank region, including VBI preamble and guard band region, excluding horizontal blank 3=2ND CRC calculated only during audio data enable

Enable LVTMA CRC Calculation

LVTMA_CRC_SIG_MASK - RW - 32 bits - [GpuF0MMReg:0x7AB4]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CRC_SIG_BLUE_MASK	7:0	0xff	CRC mask bits for LVTMA blue component

LVTMA_CRC_SIG_GREEN_MASK	15:8	0xff	CRC mask bits for LVTMA green component
LVTMA_CRC_SIG_RED_MASK	23:16	0xff	CRC mask bits for LVTMA red component
LVTMA_CRC_SIG_CONTROL_MASK	26:24	0x7	CRC mask bits for LVTMA control signals 3-bit input value: bit 2 = Vsync bit 1 = Hsync bit 0 =Data Enable

RGB and Control CRC Mask

LVTMA_CRC_SIG_RGB - RW - 32 bits - [GpuF0MMReg:0x7AB8]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CRC_SIG_BLUE (R)	7:0	0x0	CRC signature value for LVTMA blue component
LVTMA_CRC_SIG_GREEN (R)	15:8	0x0	CRC signature value for LVTMA green component
LVTMA_CRC_SIG_RED (R)	23:16	0x0	CRC signature value for LVTMA red component
LVTMA_CRC_SIG_CONTROL (R)	26:24	0x0	CRC signature value for LVTMA control signals3-bit input value: bit 2 = Vsync bit 1 = Hsync bit 0 =Data Enable

RGB and Control CRC Result

LVTMA_2ND_CRC_RESULT - RW - 32 bits - [GpuF0MMReg:0x7ABC]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_2ND_CRC_RESULT (R)	29:0	0x0	LVTMA 2ND CRC readback

LVTMA_TEST_PATTERN - RW - 32 bits - [GpuF0MMReg:0x7AC0]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_TEST_PATTERN_OUT_EN	0	0x0	LVTMA Test ouput pattern 0=Normal functionality determined by value of LVTMA_RANDOM_PATTERN_OUT_EN register 1=Test pattern output mode. The value of LVTMA_HALF_CLOCK_PATTERN_SEL determines whether a static 10-bit test data pattern or an alternating half-clock pattern will be output.
LVTMA_HALF_CLOCK_PATTERN_SEL	1	0x0	Controls between Static test pattern output and alternating static output pattern 0=10 bit test pattern from LVTMA_STATIC_TEST_PATTERN is sent during every pixel clock 1=Alternating pattern of LVTMA_STATIC_TEST_PATTERN and !(LVTMA_STATIC_TEST_PATTERN) on each subsequent pixel clock cycle is sent during every pixel clock
LVTMA_LVTM_TEST_CLOCK_DATA	2	0x0	Controls between normal output and clock output test pattern 0=Normal 1=Replace clock pattern with test data

LVTMA_RANDOM_PATTERN_OUT_EN	4	0x0	0: Output normal data or eye pattern (LVTM only) 1: Output random data 0=Normal 1=Random Pixel Data Generator circuit generates 24-bit pixel data to be encoded and transmitted
LVTMA_RANDOM_PATTERN_RESET	5	0x1	Reset random pattern to pattern seed 0=Enable Random Pixel Data Generator 1=Random Pixel Data Generator is Reset to the value in LVTMA_RANDOM_PATTERN_SEED
LVTMA_TEST_PATTERN_EXTERNAL_RESET_EN	6	0x1	0=Normal 1=External signal resets random and half clock patterns
LVTMA_LVTM_EYE_PATTERN	8	0x0	Controls between normal output and LVTM eye pattern 0=Normal 1=Replace data with eye pattern
LVTMA_STATIC_TEST_PATTERN	25:16	0x0	LVTMA test pixel. Replace the pixel value when LVTMA_TEST_PATTERN_OUT_EN=1

LVTMA_RANDOM_PATTERN_SEED - RW - 32 bits - [GpuF0MMReg:0x7AC4]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_RANDOM_PATTERN_SEED	23:0	0x222222	Initial pattern for eye pattern measurement
LVTMA_RAN_PAT_DURING_DE_ONLY	24	0x0	Controls when to output random data 0=TMDS Random Data Pattern is output for all pixels 1=TMDS Random Data Pattern is only output when DE is high

LVTMA_DEBUG - RW - 32 bits - [GpuF0MMReg:0x7AC8]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_DEBUG_EN	0	0x0	Set to 1 to enable debug mode
LVTMA_DEBUG_HSYNC	8	0x0	Debug mode HSYNC
LVTMA_DEBUG_HSYNC_EN	9	0x0	Set to 1 to enable debug mode HSYNC
LVTMA_DEBUG_VSYNC	16	0x0	Debug mode VSYNC
LVTMA_DEBUG_VSYNC_EN	17	0x0	Set to 1 to enable debug mode VSYNC
LVTMA_DEBUG_DE	24	0x0	Debug mode display enable
LVTMA_DEBUG_DE_EN	25	0x0	Set to 1 to enable debug mode display enable

LVTMA_CTL_BITS - RW - 32 bits - [GpuF0MMReg:0x7ACC]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CTL0	0	0x0	Control signal for LVTMA (encoded in Green channel).
LVTMA_CTL1	8	0x0	Control signal for LVTMA (encoded in Green channel).
LVTMA_CTL2	16	0x0	Control signal for LVTMA (encoded in Red channel).
LVTMA_CTL3	24	0x0	Control signal for LVTMA (encoded in Red channel).

LVTMA_DCBALANCER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7AD0]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_DCBALANCER_EN	0	0x1	DC Balancer Enable 0=Disable 1=Enable
LVTMA_DCBALANCER_TEST_EN	8	0x0	DC Balancer Test Enable
LVTMA_DCBALANCER_TEST_IN	19:16	0x0	DC Balancer Test Input
LVTMA_DCBALANCER_FORCE	24	0x0	DC Balancer select value to use when DCBALANCER_EN=0

LVTMA_RED_BLUE_SWITCH - RW - 32 bits - [GpuF0MMReg:0x7AD4]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_RB_SWITCH_EN	0	0x0	Switch Red and Blue encoding position. 0=Disable 1=Enable

LVTMA_DATA_SYNCHRONIZATION - RW - 32 bits - [GpuF0MMReg:0x7ADC]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_DSYNSEL	0	0x1	Data synchronization circuit select enable 0=Disable 1=Enable
LVTMA_PFREQCHG (W)	8	0x0	Write to 1 to restarts read and write address generation logic. Write of 0 has no effect. Read value is always 0. PFREQCHG must be written to 1 when the data synchronizer is started by setting DSYNSEL to 1, whenever LVTMA_DUAL_LINK_ENABLE is reprogrammed, or either PCLK_LVTMA or PCLK_LVTMA_DIRECT (IDCLK) is reprogrammed or stopped and restarted.

LVTMA Data Sychronization Control

LVTMA_CTL0_1_GEN_CNTL - RW - 32 bits - [GpuF0MMReg:0x7AE0]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CTL0_DATA_SEL	3:0	0x0	Select data to be used to generate CTL0 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Random data

LVTMA_CTL0_DATA_DELAY	6:4	0x0	Number of pixel clocks to delay CTL0 data 0=CTL0 data is delayed 0 pixel clocks 1=CTL0 data is delayed 1 pixel clocks 2=CTL0 data is delayed 2 pixel clocks 3=CTL0 data is delayed 3 pixel clocks 4=CTL0 data is delayed 4 pixel clocks 5=CTL0 data is delayed 5 pixel clocks 6=CTL0 data is delayed 6 pixel clocks 7=CTL0 data is delayed 7 pixel clocks
LVTMA_CTL0_DATA_INVERT	7	0x0	Set to 1 to invert CTL0 data 0=CTL0 data is normal 1=CTL0 data is inverted
LVTMA_CTL0_DATA_MODULATION	9:8	0x0	CTL0 data modulation control 0=CTL0 data is not modulated 1=CTL0 data is modulated by bit 0 of 2 bit counter 2=CTL0 data is modulated by bit 1 of 2 bit counter 3=CTL0 data is modulated every time 2 bit counter overflows
LVTMA_CTL0_USE_FEEDBACK_PATH	10	0x0	Set to 1 to enable CTL0 internal feedback path
LVTMA_CTL0_FB_SYNC_CONT	11	0x0	Set to 1 to force continunous toggle on CTL0 internal feedback path
LVTMA_CTL0_PATTERN_OUT_EN	12	0x0	Select CTL0 output data 0=Register value 1=Pattern generator output
LVTMA_CTL1_DATA_SEL	19:16	0x0	Select data to be used to generate CTL1 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Always (blank time)
LVTMA_CTL1_DATA_DELAY	22:20	0x0	Number of pixel clocks to delay CTL1 data 0=CTL1 data is delayed 0 pixel clocks 1=CTL1 data is delayed 1 pixel clocks 2=CTL1 data is delayed 2 pixel clocks 3=CTL1 data is delayed 3 pixel clocks 4=CTL1 data is delayed 4 pixel clocks 5=CTL1 data is delayed 5 pixel clocks 6=CTL1 data is delayed 6 pixel clocks 7=CTL1 data is delayed 7 pixel clocks
LVTMA_CTL1_DATA_INVERT	23	0x0	Set to 1 to invert CTL1 data 0=CTL1 data is normal 1=CTL1 data is inverted
LVTMA_CTL1_DATA_MODULATION	25:24	0x0	CTL1 data modulation control 0=CTL1 data is not modulated 1=CTL1 data is modulated by bit 0 of 2 bit counter 2=CTL1 data is modulated by bit 1 of 2 bit counter 3=CTL1 data is modulated every time 2 bit counter overflows
LVTMA_CTL1_USE_FEEDBACK_PATH	26	0x0	Set to 1 to enable CTL1 internal feedback path
LVTMA_CTL1_FB_SYNC_CONT	27	0x0	Set to 1 to force continunous toggle on CTL1 internal feedback path
LVTMA_CTL1_PATTERN_OUT_EN	28	0x0	Select CTL1 output data 0=Register value 1=Pattern generator output
LVTMA_2BIT_COUNTER_EN	31	0x0	Set to 1 to enable 2-bit data modulation counter 0=Disable 1=Enable

LVTMA_CTL2_3_GEN_CNTL - RW - 32 bits - [GpuF0MMReg:0x7AE4]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_CTL2_DATA_SEL	3:0	0x0	Select data to be used to generate CTL2 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Always (blank time)
LVTMA_CTL2_DATA_DELAY	6:4	0x0	Number of pixel clocks to delay CTL2 data 0=CTL2 data is delayed 0 pixel clocks 1=CTL2 data is delayed 1 pixel clocks 2=CTL2 data is delayed 2 pixel clocks 3=CTL2 data is delayed 3 pixel clocks 4=CTL2 data is delayed 4 pixel clocks 5=CTL2 data is delayed 5 pixel clocks 6=CTL2 data is delayed 6 pixel clocks 7=CTL2 data is delayed 7 pixel clocks
LVTMA_CTL2_DATA_INVERT	7	0x0	Set to 1 to invert CTL2 data 0=CTL2 data is normal 1=CTL2 data is inverted
LVTMA_CTL2_DATA_MODULATION	9:8	0x0	CTL2 data modulation control 0=CTL2 data is not modulated 1=CTL2 data is modulated by bit 0 of 2 bit counter 2=CTL2 data is modulated by bit 1 of 2 bit counter 3=CTL2 data is modulated every time 2 bit counter overflows
LVTMA_CTL2_USE_FEEDBACK_PATH	10	0x0	Set to 1 to enable CTL2 internal feedback path
LVTMA_CTL2_FB_SYNC_CONT	11	0x0	Set to 1 to force continunous toggle on CTL2 internal feedback path
LVTMA_CTL2_PATTERN_OUT_EN	12	0x0	Select CTL2 output data 0=Register value 1=Pattern generator output
LVTMA_CTL3_DATA_SEL	19:16	0x0	Select data to be used to generate CTL3 pattern (selected fields are ORed together) [0]: Display Enable [1]: VSYNC [2]: HSYNC [3] Always (blank time)
LVTMA_CTL3_DATA_DELAY	22:20	0x0	Number of pixel clocks to delay CTL3 data 0=CTL3 data is delayed 0 pixel clocks 1=CTL3 data is delayed 1 pixel clocks 2=CTL3 data is delayed 2 pixel clocks 3=CTL3 data is delayed 3 pixel clocks 4=CTL3 data is delayed 4 pixel clocks 5=CTL3 data is delayed 5 pixel clocks 6=CTL3 data is delayed 6 pixel clocks 7=CTL3 data is delayed 7 pixel clocks
LVTMA_CTL3_DATA_INVERT	23	0x0	Set to 1 to invert CTL3 data 0=CTL3 data is normal 1=CTL3 data is inverted
LVTMA_CTL3_DATA_MODULATION	25:24	0x0	CTL3 data modulation control 0=CTL3 data is not modulated 1=CTL3 data is modulated by bit 0 of 2 bit counter 2=CTL3 data is modulated by bit 1 of 2 bit counter 3=CTL3 data is modulated every time 2 bit counter overflows
LVTMA_CTL3_USE_FEEDBACK_PATH	26	0x0	Set to 1 to enable CTL3 internal feedback path
LVTMA_CTL3_FB_SYNC_CONT	27	0x0	Set to 1 to force continunous toggle on CTL3 internal feedback path

LVTMA_CTL3_PATTERN_OUT_EN	28	0x0	Select CTL3 output data 0=Register value 1=Pattern generator output

LVTMA_PWRSEQ_REF_DIV - RW - 32 bits - [GpuF0MMReg:0x7AE8]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PWRSEQ_REF_DIV	11:0	0x0	Determines frequency of reference for power sequencing/nFrequency = REF/(PWREQ_REF_DIV+1)
LVTMA_BL_MOD_REF_DIV	27:16	0x0	Determines frequency of modulated BLON/nFrequency = REF/((BL_MOD_REF_DIV+1)*(BL_MOD_RES+1))

LVTMA_PWRSEQ_DELAY1 - RW - 32 bits - [GpuF0MMReg:0x7AEC]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PWRUP_DELAY1	7:0	0x0	Number of LVTMA_PWRSEQ_REF pulses to delay from DIGON enable to SYNCEN (LVTMA transmitter macro) enable during powerupMust be long enough for bandgap reference and PLL startup (= reset assertion time + PLL lock time).
LVTMA_PWRUP_DELAY2	15:8	0x0	Number of LVTMA_PWRSEQ_REF pulses to delay from SYNCEN enable to BLON enable during powerup
LVTMA_PWRDN_DELAY1	23:16	0x0	Number of LVTMA_PWRSEQ_REF pulses to delay from BLON disable to SYNCEN disable during power down
LVTMA_PWRDN_DELAY2	31:24	0x0	Number of LVTMA_PWRSEQ_REF pulses to delay from SYNCEN disable to DIGON disable during power down

LVTMA_PWRSEQ_DELAY2 - RW - 32 bits - [GpuF0MMReg:0x7AF0]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PWRDN_MIN_LENGTH	7:0	0x0	Number of LVTMA_PWRSEQ_REF pulses to delay from completion of powerdown to powerup

LVTMA_PWRSEQ_CNTL - RW - 32 bits - [GpuF0MMReg:0x7AF4]			
Field Name	**Bits**	**Default**	**Description**

LVTMA_PWRSEQ_EN	0	0x0	Set to 1 to enable the power sequencer. When disabled, SYNCEN will be set to 1 and all other outputs set to 0. This allows software or HPD logic to control the LVTM macro transmitter enable bits.Software should set the target state and macro enable bits in LVTMA_TRANSMITTER_ENABLE to 0 (off) before enabling this bit. When enabled, the power sequencer will be in the POWERDOWN_DONE state. Once the power sequencer is enabled, the macro enable bits in LVTMA_TRANSMITTER_ENABLE should be set to 1 if the LVTMA macro is to be used. This allows the power sequencer to control the macro enable bits.The macro will not be turned on until LVTMA_PWRSEQ_TARGET_STATE is set to 1.
LVTMA_PWRSEQ_DISABLE_SYNCEN_CONTROL_OF_TX_EN	1	0x0	Select whether SYNCEN will disable LVTM Tx drivers. 0: Enable SYNCEN control of Tx enables. 1: Disable SYNCEN control of Tx enables.
LVTMA_PLL_ENABLE_PWRSEQ_MASK	2	0x0	Determines whether power sequencer can force LVTMA_PLL_ENABLE to 0. When LVTMA_PLL_ENABLE_PWRSEQ_MASK=1, enable will be deasserted 1 us after hardware asserts PLL reset to match LVTMA macro timing requirements 0=Power Sequencer cannot override PLL enable 1=Power Sequencer can override PLL enable
LVTMA_PLL_RESET_PWRSEQ_MASK	3	0x0	Determines whether power sequencer can force LVTMA_PLL_RESET to 1. When LVTMA_PLL_RESET_PWRSEQ_MASK=1, reset will be asserted for 10 us after hardware enables PLL to match LVTMA macro timing requirements 0=Power Sequencer cannot override PLL reset 1=Power Sequencer can override PLL reset
LVTMA_PWRSEQ_TARGET_STATE	4	0x0	0:LCD off 1:LCD on PM_PWRSEQ_TARGET_STATE (from power management) must also be 1 to enable the panel.
LVTMA_SYNCEN	8	0x0	LVTM transmitter enable
LVTMA_SYNCEN_OVRD	9	0x0	Enable override of power sequencer SYNCEN by register value 0=Disable 1=Enable
LVTMA_SYNCEN_POL	10	0x0	Polarity of output SYNCEN signal 0=Non-invert 1=Invert
LVTMA_DIGON	16	0x0	LVTM digital voltage 0:off 1:on
LVTMA_DIGON_OVRD	17	0x0	Enable override of power sequencer DIGON by register value 0=Disable 1=Enable
LVTMA_DIGON_POL	18	0x0	Polarity of output DIGON signal 0=Non-invert 1=Invert
LVTMA_BLON	24	0x0	LVTM backlight voltage 0:off 1:on
LVTMA_BLON_OVRD	25	0x0	Enable override of power sequencer BLON (before modulation) by register value 0=Disable 1=Enable
LVTMA_BLON_POL	26	0x0	Polarity of output BLON signal 0=Non-invert 1=Invert

LVTMA_PWRSEQ_STATE - RW - 32 bits - [GpuF0MMReg:0x7AF8]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PWRSEQ_TARGET_STATE_R (R)	0	0x0	Power sequencer target state (0=powerdown, 1=powerup)This is an AND of LVTMA_PWRSEQ_TARGET_STATE and the enable from power management logic. 0=power down 1=power up
LVTMA_PWRSEQ_DIGON (R)	1	0x0	Power sequencer DIGON state 0=off 1=on
LVTMA_PWRSEQ_SYNCEN (R)	2	0x0	Power sequencer SYNCEN state 0=off 1=on
LVTMA_PWRSEQ_BLON (R)	3	0x0	Power sequencer BLON state 0=off 1=on
LVTMA_PWRSEQ_DONE (R)	4	0x0	Indicates that power sequencer has reached target state (either DISABLED, POWERDOWN_DONE, or POWERUP_DONE depending on LVTMA_PWRSEQ_EN and LVTMA_PWRSEQ_TARGET_STATE_R) 0=active 1=done
LVTMA_PWRSEQ_STATE (R)	11:8	0x0	Indicates power sequencer's state 0=DISABLED: D=0, B=0, S=LVTMA_PWRSEQ_TARGET_STATE_R 1=POWERUP0: D=0 S=0 B=0 2=POWERUP1: D=1 S=0 B=0 3=POWERUP2: D=1 S=1 B=0 4=POWERUP_DONE: D=1 S=1 B=1 5=POWERDOWN0: D=1 S=1 B=1 6=POWERDOWN1: D=1 S=1 B=0 7=POWERDOWN2: D=1 S=0 B=0 8=POWERDOWN_DELAY: D=0 S=0 B=0 Ignore powerup request 9=POWERDOWN_DONE: D=0 S=0 B=0

LVTMA_BL_MOD_CNTL - RW - 32 bits - [GpuF0MMReg:0x7AFC]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_BL_MOD_EN	0	0x0	Enable backlight modulation 0=Disable LCD backlight modulation 1=Enable LCD backlight modulation
LVTMA_BL_MOD_LEVEL	15:8	0x0	Determines duty cycle of BLON signal/nDuty cycle = BL_MOD_LEVEL/(BL_MOD_RES+1)
LVTMA_BL_MOD_RES	23:16	0xff	Determines resolution of BLON signal/nNumber of steps of backlight brightness = (BL_MOD_RES+1)

LVTMA_LVTM_DATA_CNTL - RW - 32 bits - [GpuF0MMReg:0x7B00]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_LVTM_24BIT_ENABLE	0	0x0	Enable 4th data channel for 24-bit output 0=Disable 1=Enable
LVTMA_LVTM_24BIT_FORMAT	4	0x0	0: Use LDI format for 888 RGB: (LSB first) CH0: G2, R7-2 CH1: B3-2, G7-3 CH2: DE, VSYN, HSYN, B7-4 CH3: N/A, B1-0, G1-0, R1-0 1: Use FPDI format for 888 RGB: (LSB first) CH0: G0, R5-0 CH1: B1-0, G5-1 CH2: DE, VSYN, HSYN, B5-2 CH3: N/A, B7-6, G7-6, R7-6
LVTMA_LVTM_2ND_CHAN_DE	8	0x0	0=Control bit[0] in third channel of 2nd link 1=DE in third data port of 2nd channel
LVTMA_LVTM_2ND_CHAN_VS	9	0x0	0=Control bit[1] in third channel of 2nd link 1=VS in third data port of 2nd channel
LVTMA_LVTM_2ND_CHAN_HS	10	0x0	0=Control bit[2] in third channel of 2nd link 1=HS in third data port of 2nd channel
LVTMA_LVTM_2ND_LINK_CNTL_BITS	14:12	0x0	Bits to put in place of DE (bit 2), VS (bit 1), HS (bit 0) based on programming of bits 8, 9 & 11 above
LVTMA_LVTM_FP_POL	16	0x0	Polarity of frame pulse encoded in LVTM data stream 0=active high Frame Pulse / Vsync 1=active low Frame Pulse / Vsync
LVTMA_LVTM_LP_POL	17	0x0	Polarity of line pulse encoded in LVTM data stream 0=active high Line Pulse / Hsync 1=active low Line Pulse / Hsync
LVTMA_LVTM_DTMG_POL	18	0x0	Polarity of display enable encoded in LVTM data stream 0=active high Display Enable / MOD 1=active low Display Enable / MOD

LVTMA_MODE - RW - 32 bits - [GpuF0MMReg:0x7B04]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_TMDS_LVTMb	0	0x0	Selects whether LVTMA path is in LVTM or TMDS mode.Also drives LVTMA macro ITXSEL pin. HW for mode that is not selected is disabled. 0=LVTM 1=TMDS

LVTMA_TRANSMITTER_ENABLE - RW - 32 bits - [GpuF0MMReg:0x7B08]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_LNKC0EN	1	0x0	LVTMA link0 clock channel enable (ICH3EN)(set to 1 whenever LVTM is enabled)
LVTMA_LNKD00EN	2	0x0	LVTMA link0 data channel 0 enable (ICH0EN)(set to 1 whenever LVTM is enabled)
LVTMA_LNKD01EN	3	0x0	LVTMA link0 data channel 1 enable (ICH1EN)(set to 1 whenever LVTM is enabled)
LVTMA_LNKD02EN	4	0x0	LVTMA link0 data channel 2 enable (ICH2EN)(set to 1 whenever LVTM is enabled)
LVTMA_LNKD03EN	5	0x0	LVTMA link0 data channel 3 enable (ICH4EN)(set to 1 when LVTM is enabled and in 24bpp LVTM mode)

LVTMA_LNKC1EN	9	0x0	LVTMA link1 clock channel enable (ICH8EN)(set to 1 when LVTM is enabled and in LVTM dual-link mode)
LVTMA_LNKD10EN	10	0x0	LVTMA link1 data channel 0 enable (ICH5EN)(set to 1 when LVTM is enabled and in dual-link mode)
LVTMA_LNKD11EN	11	0x0	LVTMA link1 data channel 1 enable (ICH6EN)(set to 1 when LVTM is enabled and in dual-link mode)
LVTMA_LNKD12EN	12	0x0	LVTMA link1 data channel 2 enable (ICH7EN)(set to 1 when LVTM is enabled and in dual-link mode)
LVTMA_LNKD13EN	13	0x0	LVTMA link1 data channel 3 enable (ICH9EN)(set to 1 when LVTM is enabled and in 24bpp LVTM dual-link mode)
LVTMA_LNKCEN_HPD_MASK	17	0x0	0:Disallow 1:Allow override of LVTMA_LNKCXEN by HPD on disconnect 0=Result from HPD circuit can not override LVTMA_LNKC0EN 1=Result from HPD circuit overrides LVTMA_LNKC0EN on disconnect
LVTMA_LNKDEN_HPD_MASK	18	0x0	0:Disallow 1:Allow override of LVTMA_LNKDXEN by HPD on disconnect 0=Result from HPD circuit can not override LVTMA_LNKDXEN 1=Result from HPD circuit overrides LVTMA_LNKDXEN on disconnect

LVTMA_LOAD_DETECT - RW - 32 bits - [GpuF0MMReg:0x7B0C]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_LOAD_DETECT_ENABLE	0	0x1	0: Disable 1: Enable LVTMA macro load detect functionDrives IMSEN macro inputNote: macro doesn't currently have this function, but leave register or placeholder here for future implementations
LVTMA_LOAD_DETECT (R)	4	0x0	From LVTMA macro load detect output 0: No load detected 1: Load detectedNote: macro doesn't currently have this function, but leave register or placeholder here for future implementations

RTL support for this feature is included although LVTM macro doesn't support it yet

LVTMA_MACRO_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7B10]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PLL_CP_GAIN	5:0	0x3	LVTMA PLL charge-pump current adjustment. LVTM mode: IDCLK > 50MHz = 5, IDCLK =< 50MHz = 12. TMDS mode: IDCLK >= 70MHz = 17, IDCLK < 70MHz = 32
LVTMA_PLL_VCO_GAIN	13:8	0x4	LVTMA PLL VCO gain adjustment. LVTM mode: IDCLK > 50MHz = 5, IDCLK =< 50MHz = 1. TMDS mode: IDCLK >= 70MHz = 5, IDCLK < 70MHz = 1
LVTMA_PLL_DUTY_CYCLE	19:16	0x2	LVTMA PLL output clock duty cycle adjustment. Should be programmed to 3
LVTMA_IPLT	28:24	0x0	reserved pins for LVTM macro, LVTM mode: LVTMA_IPLT = 2. TMDS mode: LVTMA_IPLT = 1
LVTMA_ICOSEL	31	0x0	PLL ICO select

LVTMA_TRANSMITTER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7B14]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PLL_ENABLE	0	0x0	LVTMA transmitter PLL enable. Normal operation = 1, Power down = 0 0=LVTMA Transmitter PLL is disabled 1=LVTMA Transmitter PLL is enabled
LVTMA_PLL_RESET	1	0x1	Transmitter PLL A reset signal (active high, level sentive). A low level puts the transmitter PL in normal operating mode. A high level resets the PLL. The macro will not operate properly before going through the reset sequence
LVTMA_PLL_ENABLE_HPD_MASK	3:2	0x0	Determines whether result from HPD circuit can override LVTMA_PLL_ENABLE and LVTMA_ PLL_RESET. 0=Result from HPD circuit can not override LVTMA_PLL_ENABLE 1=Result from HPD circuit overrides LVTMA_PLL_ENABLE on disconnect 2=Result from HPD circuit overrides LVTMA_PLL_ENABLE on connect 3=Result from HPD circuit overrides LVTMA_PLL_ENABLE
LVTMA_IDSCKSEL	4	0x1	0=LVTM Transmitter uses pclk_LVTMa (IPIXCLK) 1=LVTM Transmitter uses pclk_LVTMa_direct (IDCLK)
LVTMA_BGSLEEP	5	0x0	LVTMA Bandgap macro disable. Set to 0 for normal operation (hardware will enable the macro whenever LVTMA is active), 1 to turn the bandgap macro off regardless of LVTMA status.Note that LVTMA shares the bandgap macro with DACB. For the shared macro either DACB or LVTMA can turn the macro on. Set to 0 in LVTM mode, 1 in TMDS mode. 0=Normal operation 1=Disable bandgap
LVTMA_PLL_RESET_HPD_MASK	7	0x0	Set this bit to automatically reset TMDS macro PLL (channel B, when LVTMA_SPLIT=1) on HPD connect event, reset pulse= 10us 0=LVTM macro pll is not reset on hot plug detect connect 1=LVTM macro pll reset on hot plug detect for 10 us
LVTMA_TMCLK	8	0x0	(only bit0 is used in LVTM macro)For macro debug only
LVTMA_TMCLK_FROM_PADS	13	0x0	0=Input to ITMCLK pins on macro come from LVTMA_TMCLK field 1=Input to ITMCLK pins on macro come from pads
LVTMA_TDCLK	14	0x0	For macro debug only
LVTMA_TDCLK_FROM_PADS	15	0x0	0=Input to ITDCLK pin on macro comes from LVTMA_TDCLK field 1=Input to ITDCLK pin on macro comes from pads
LVTMA_CLK_PATTERN	25:16	0x63	Data to be serialized to generate a clock when LVTMA_USE_CLK_DATA=0 If LVTMA_USE_CLK_DATA=1, LVTMA_CLK_PATTERN is 'don't care' and can be set to 0 In LVTM mode the 7 LSBs are used, in TMDS mode all 10 bits are used. For National Compatible mode, set LVTMA_CLK_PATTERN to XXX1100011 For VESA FPDI-2 Compatible mode, set LVTMA_CLK_PATTERN to XXX0001111 For TMDS, set LVTMA_CLK_PATTERN to 0000011111

LVTMA_BYPASS_PLL	28	0x1	Controls ICHCSEL1 pin on LVTM macro 0: Coherent mode: transmitted clock is PLL output 1: Incoherent mode: transmitted clock is PLL input 0=0: Coherent mode: transmitted clock is PLL output 1=1: Incoherent mode: transmitted clock is PLL input
LVTMA_USE_CLK_DATA	29	0x0	Controls ICHCSEL2 pin on LVTM macro Use to determine whether clock comes from PLL output or serialized LVTMA_CLK_PATTERN See macro spec for recommended settings in TMDS and LVTM modes 0=0: Use serialized data (LVTMA_CLK_PATTERN) as clock 1=1: Use clock selected by LVTMA_BYPASS_PLL (ICHCSEL1)
LVTMA_INPUT_TEST_CLK_SEL	31	0x0	Controls ITCLKSEL pin on LVTM macro

LVTMA_REG_TEST_OUTPUT - RW - 32 bits - [GpuF0MMReg:0x7B18]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_REG_TEST_OUTPUT (R)	9:0	0x0	From LVTMA macro OTDAT(9:0) outputs
LVTMA_TEST_CNTL	21:16	0x0	Drives LVTMA macro ITEST(5:0) control bits (for debug only)

LVTMA_TRANSMITTER_DEBUG - RW - 32 bits - [GpuF0MMReg:0x7B1C]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PLL_DEBUG	7:0	0x0	Drives ITPL pins on LVTMA macro (this functionality doesn't exist - pins are unconnected)
LVTMA_TX_DEBUG	11:8	0x0	Drives ITX pins on LVTMA macro (this functionality doesn't exist ? pins are unconnected)

Reserved for debugging purposes

LVTMA_DITHER_RAND_SEED - RW - 32 bits - [GpuF0MMReg:0x7B20]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_RAND_R_SEED	7:0	0x0	Seed for random red, the random seed is 1'b1, LVTMA_RAND_R_SEED[2:0], 3LVTMA_RAND_R_SEED = 28 bits
LVTMA_RAND_G_SEED	15:8	0x99	Seed for random green, the random seed is 1'b1, LVTMA_RAND_G_SEED[2:0], 3LVTMA_RAND_G_SEED = 28 bits
LVTMA_RAND_B_SEED	23:16	0xdd	Seed for random bleu, the random seed is 1'b1, LVTMA_RAND_B_SEED[2:0], 3LVTMA_RAND_B_SEED = 28 bits

programmable seed for random dithering

LVTMA_TRANSMITTER_ADJUST - RW - 32 bits - [GpuF0MMReg:0x7B24]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_TXOP	4:0	0x0	transmitter opam adjustment. LVTM mode: LVTMA_TXOP = 7. TMDS mode: LVTMA_TXOP = 0
LVTMA_NTXVS	12:8	0x0	bias current control for output driver for n current source. LVTM mode: LVTMA_NTXVS = 11. TMDS mode: LVTMA_NTXVS = 24
LVTMA_PTXVS	20:16	0x0	bias current control for output driver for p current source. LVTM mode: LVTMA_PTXVS = 11. TMDS mode: LVTMA_PTXVS = 0
LVTMA_TXT	28:24	0x0	reserved for transmitter. LVTMA_TXT[0]: 1 in LVTM mode, 0 in TMDS mode. LVTMA_TXT[1]: 1 in LVTM mode, 0 in TMDS mode.LVTMA_TXT[2] = 0. LVTMA_TXT[3] = 0. LVTMA_TXT[4] = 0
LVTMA_PUDSEL	30	0x0	output driver pull-down selection
LVTMA_REFSEL	31	0x0	Current source self bias selection

LVTM macro transmitter adjustment control in split mode

LVTMA_PREEMPHASIS_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7B28]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_PREMPHEN	0	0x0	pre-emphasi enable
LVTMA_PREMCHSEL	2	0x0	pre-emphasi clock channel selection
LVTMA_PREMPH_DV	7:4	0x0	pre-emphasi pulse height control
LVTMA_PREMPH_DT	15:12	0x0	pre-emphasi pulse width control
LVTMA_NTXSPREM	24:20	0x0	pre-emphasi n-bias control

LVTM macro pre-emphasis control

LVTMA_SPLIT_TRANSMITTER_ENABLE - RW - 32 bits - [GpuF0MMReg:0x792C]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SPLIT_TX_ENABLE	0	0x0	enable transmitter when split mode is enable 0=Disable 1=Enable
LVTMA_SPLIT_LNKCEN	1	0x0	enable transmitter clock channel when split mode is enable
LVTMA_SPLIT_LNKD0EN	2	0x0	enable transmitter data channel 0 when split mode is enable
LVTMA_SPLIT_LNKD1EN	3	0x0	enable transmitter data channel 1 when split mode is enable
LVTMA_SPLIT_LNKD2EN	4	0x0	enable transmitter data channel 2 when split mode is enable
LVTMA_SPLIT_TX_ENABLE_HPD_MASK	16	0x0	0:Disallow 1:Allow override of LVTMA_SPLIT_TX_ENABLE by HPD on disconnect 0=Result from HPD circuit can not override LVTMA_SPLIT_TX_ENABLE 1=Result from HPD circuit can override LVTMA_SPLIT_TX_ENABLE on disconnect

LVTMA_SPLIT_LNKCEN_HPD_MASK	17	0x0	0:Disallow 1:Allow override of LVTMA_SPLIT_LNKCEN by HPD on disconnect 0=Result from HPD circuit can not override LVTMA_SPLIT_LNKCEN 1=Result from HPD circuit overrides TMDSA_LNKCBEN on disconnect
LVTMA_SPLIT_LNKDEN_HPD_MASK	18	0x0	0:Disallow 1:Allow override of LVTMA_SPLIT_LNKDxEN by HPD on disconnect 0=Result from HPD circuit can not override LVTMA_SPLIT_LNKDxEN 1=Result from HPD circuit overrides LVTMA_SPLIT_LNKDxEN on disconnect

tmda macro transmitter control in split mode

LVTMA_SPLIT_LOAD_DETECT - RW - 32 bits - [GpuF0MMReg:0x7930]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SPLIT_LOAD_DETECT_ENABLE	0	0x1	0: Disable 1: Enable TMDSA macro load detect functionDrives IMSEN macro input when split mode is enable
LVTMA_SPLIT_LOAD_DETECT (R)	4	0x0	From TMDSA macro OMSEN output 0: No load detected 1: Load detected when split mode is enable

LVTMA_SPLIT_PLL_ADJUST - RW - 32 bits - [GpuF0MMReg:0x7934]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SPLIT_PLL_CP_GAIN	5:0	0xb	tmds macro channel B charge pump gain control in split mode
LVTMA_SPLIT_PLL_VCO_GAIN	13:8	0x7	tmds macro channel B vco control in split mode
LVTMA_SPLIT_PLL_DUTY_CYCLE	17:16	0x0	tmds macro channel B clock duty cycle control in split mode

LVTMA_SPLIT_TRANSMITTER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7938]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_SPLIT_PLL_ENABLE	0	0x0	TMDSA transmitter's PLLB enable in split mode. This can power down the PLL. 0=Disable 1=Enable
LVTMA_SPLIT_PLL_RESET	1	0x1	TMDSA transmitter's PLLB reset in split mode. PLL will start the locking acquisition process once this becomes low.

LVTMA_SPLIT_PLL_ENABLE_HPD_MASK	3:2	0x0	Determines whether result from HPD circuit can override TMDSA_PLL_ENABLE and TMDSA_PLL_RESET Bit 0: Set to 1 to enable override on disconnect Bit 1: Set to 1 to enable override on connect. 0=Result from HPD circuit can not override LVTMA_SPLIT_PLL_ENABLE 1=Result from HPD circuit overrides LVTMA_SPLIT_PLL_ENABLE on disconnect 2=Result from HPD circuit overrides LVTMA_SPLIT_PLL_ENABLE on connect 3=Result from HPD circuit overrides LVTMA_SPLIT_PLL_ENABLE
LVTMA_SPLIT_IDSCKSEL	4	0x1	picclk or pllclk select in plit mode 0=0: TMDS split Transmitter uses pclk_tmdsa (IPIXCLK) 1=1: TMDS split Transmitter uses pclk_LVTMa_direct (IDCLK)
LVTMA_SPLIT_PLL_PWRUP_SEQ_EN	6	0x0	Enable hardware delay of PLL enable / reset on power up / down to match macro timing requirements. When LVTMA_SPLIT_PLL_PWRUP_SEQ_EN=1, PLL will be reset 1 us before PLL enable is deasserted, and PLL reset will be asserted for 10 us after PLL enable is asserted. This timing is provided to match the TMDS macro timing specification. 0=Disabled 1=Delay Enable/ Reset for clean PLL power up/down
LVTMA_SPLIT_PLL_RESET_HPD_MASK	7	0x0	Set this bit to automatically reset TMDS macro PLL on HPD connect event, reset pulse= 10us 0=TMDS macro pll is not reset on hot plug detect connect 1=TMDS macro pll reset on hot plug detect connect for 10 us, apply only when LVTM_SPLIT=1
LVTMA_SPLIT_BYPASS_PLL	28	0x1	Controls ICHCSEL pin in plit mode on TMDSA macro 0: Coherent mode: transmitted clock is PLL output 1: Incoherent mode: transmitted clock is PLL input 0=0: TMDS split channel transmitter is in coherent mode 1=1: Tmds split channel transmitter is in incoherent mode

LVTMA_SPLIT_TRANSMITTER_ADJUST - RW - 32 bits - [GpuF0MMReg:0x793C]			
Field Name	**Bits**	**Default**	**Description**
LVTMA_TX_VOLTAGE_SWING	3:0	0xa	tmds macro transmitter b, voltage swing control, in split mode
LVTMA_SPLIT_TXPC	9:8	0x0	tmds macro transmitter b, pulse current control, in split mode
LVTMA_SPLIT_TXPW	13:12	0x0	tmds macro transmitter b, pulse width control, in split mode
LVTMA_TX_VS_COMP	21:20	0x0	tmds macro transmitter b, voltage swing compensation control

Hot Plug Detection Registers

DC_HOT_PLUG_DETECT1_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D00]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT1_EN	0	0x0	Enable 1st HPD circuit When disabled, HPD interrupts will not happen and DC_HOT_PLUG_DETECT1_SENSE will not change 0=Disable 1=Enable

DC_HOT_PLUG_DETECT1_INT_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D04]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT1_INT_STATUS (R)	0	0x0	Interrupt generated by 1st HPD circuit - connect or disconnect has taken place
DC_HOT_PLUG_DETECT1_SENSE (R)	1	0x0	Connection status of panel being monitored by the 1st HPD circuit 0=nothing connected to HPD1 1=panel connected to HPD1

DC_HOT_PLUG_DETECT1_INT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D08]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT1_INT_ACK (W)	0	0x0	Interrupt acknowledge for the 1st HPD circuit
DC_HOT_PLUG_DETECT1_INT_POLARITY	8	0x0	Polarity of 1st HPD circuit 0=generate interrupt on disconnect 1=generate interrupt on connect
DC_HOT_PLUG_DETECT1_INT_EN	16	0x0	Enable Interrupts on the 1st HPD circuit 0=Disable 1=Enable

DC_HOT_PLUG_DETECT2_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D10]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT2_EN	0	0x0	Enable 2nd HPD circuit When disabled, HPD interrupts will not happen and DC_HOT_PLUG_DETECT2_SENSE will not change 0=Disable 1=Enable

DC_HOT_PLUG_DETECT2_INT_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D14]

Field Name	Bits	Default	Description
DC_HOT_PLUG_DETECT2_INT_STATUS (R)	0	0x0	Interrupt generated by 2nd HPD circuit - connect or disconnect has taken place
DC_HOT_PLUG_DETECT2_SENSE (R)	1	0x0	Connection status of panel being monitored by the 2nd HPD circuit 0=nothing connected to HPD2 1=panel connected to HPD2

DC_HOT_PLUG_DETECT2_INT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D18]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT2_INT_ACK (W)	0	0x0	Interrupt acknowledge for the 2nd HPD circuit
DC_HOT_PLUG_DETECT2_INT_POLARITY	8	0x0	Polarity of 2nd HPD circuit. 0=generate interrupt on disconnect 1=generate interrupt on connect
DC_HOT_PLUG_DETECT2_INT_EN	16	0x0	Enable Interrupts on the 2nd HPD circuit 0=Disable 1=Enable

DC_HOT_PLUG_DETECT_CLOCK_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D20]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT_CLOCK_ENABLE	0	0x1	Enable HPD clock 0=Disable 1=Enable
DC_HOT_PLUG_DETECT_CLOCK_SEL	17:16	0x0	Select HPD reference frequency 0=Period = 8192 us 1=512 us 2=32 us 3=2 us

DC_HOT_PLUG_DETECT3_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D24]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT3_EN	0	0x0	0=Disable 1=Enable

DC_HOT_PLUG_DETECT3_INT_STATUS - RW - 32 bits - [GpuF0MMReg:0x7D28]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT3_INT_STATUS (R)	0	0x0	

DC_HOT_PLUG_DETECT3_SENSE (R)	1	0x0	0=nothing connected to HPD3 1=panel connected to HPD3

DC_HOT_PLUG_DETECT3_INT_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7D2C]			
Field Name	**Bits**	**Default**	**Description**
DC_HOT_PLUG_DETECT3_INT_ACK (W)	0	0x0	
DC_HOT_PLUG_DETECT3_INT_POLARITY	8	0x0	0=generate interrupt on disconnect 1=generate interrupt on connect
DC_HOT_PLUG_DETECT3_INT_EN	16	0x0	0=Disable 1=Enable

2.9.2 Display Output Control Registers

Registers for controlling general DISPOUT functionality

DC_GENERICA - RW - 32 bits - [GpuF0MMReg:0x7DC0]			
Field Name	**Bits**	**Default**	**Description**
GENERICA_EN	0	0x0	Enable signal for GENERICA pad
GENERICA_SEL	11:8	0x0	Select signals for GENERICA pad 0=DACA Stereosync 1=DACB Stereosync 2=DACA Pixclk 3=DACB Pixclk 4=DVOA CTL3 5=P1 PLLCLK 6=P2 PLLCLK 7=DVOA Stereosync 8=DACA Field Number 9=DACB Field Number 10=GENERICA test debug clock from DCCG 11=SYNCEN 12=GENERICA test debug clock from SCG 13=Reserved 14=Reserved 15=Reserved

DC_GENERICB - RW - 32 bits - [GpuF0MMReg:0x7DC4]			
Field Name	**Bits**	**Default**	**Description**
GENERICB_EN	0	0x0	Enable signals for GENERICB pad
GENERICB_SEL	11:8	0x0	Select signal for GENERICB pad 0=DACA Stereosync 1=DACB Stereosync 2=DACA PIXCLK 3=DACB PIXCLK 4=DVOA CTL3 5=P1 PLLCLK 6=P2 PLLCLK 7=DVOA Stereosync 8=DACA Field Number 9=DACB Field Number 10=GENERICB test debug clock from DCCG 11=SYNCEN 12=GENERICA test debug clock from SCG 13=Reserved 14=Reserved 15=Reserved

DC_PAD_EXTERN_SIG - RW - 32 bits - [GpuF0MMReg:0x7DCC]			
Field Name	**Bits**	**Default**	**Description**

DC_PAD_EXTERN_SIG_SEL	3:0	0x0	Select pin PAD_EXTERN_SIGNAL is connected to 0=PAD_EXTERN_SIGNAL is connected to HSYNCA pin 1=PAD_EXTERN_SIGNAL is connected to VSYNCA pin 2=PAD_EXTERN_SIGNAL is connected to HSYNCB pin 3=PAD_EXTERN_SIGNAL is connected to VSYNCB pin 4=PAD_EXTERN_SIGNAL is connected to GENERICA pin 5=PAD_EXTERN_SIGNAL is connected to GENERICB pin 6=PAD_EXTERN_SIGNAL is connected to GENERICC pin 7=PAD_EXTERN_SIGNAL is connected to HPD1 pin 8=PAD_EXTERN_SIGNAL is connected to HPD2 pin 9=PAD_EXTERN_SIGNAL is connected to DDC1CLK pin 10=PAD_EXTERN_SIGNAL is connected to DDC1DATA pin 11=PAD_EXTERN_SIGNAL is connected to DDC2CLK pin 12=PAD_EXTERN_SIGNAL is connected to DDC2DATA pin 13=PAD_EXTERN_SIGNAL is connected to VHAD(1) pin 14=PAD_EXTERN_SIGNAL is connected to VHAD(0) pin 15=PAD_EXTERN_SIGNAL is connected to VPHCTL pin

Select for PAD_EXTERN_SIGNAL

DC_REF_CLK_CNTL - RW - 32 bits - [GpuF0MMReg:0x7DD4]			
Field Name	**Bits**	**Default**	**Description**
HSYNCA_OUTPUT_SEL	1:0	0x0	0=Reference Clock Output disabled 1=PPLL1 Reference Clock Output 2=PPLL2 Reference Clock Output 3=Reserved
HSYNCB_OUTPUT_SEL	9:8	0x0	0=Reference Clock Output disabled 1=PPLL1 Reference Clock Output 2=PPLL2 Reference Clock Output 3=Reserved

Control output of external reference clocks

DISP_INTERRUPT_STATUS - RW - 32 bits - [GpuF0MMReg:0x7EDC]			
Field Name	**Bits**	**Default**	**Description**
SCL_DISP1_MODE_CHANGE_INTERRUPT (R)	0	0x0	Interrupt that can be generated by the primary display controller's scaler when it detects any change in the scale ratio or number of taps the scaling filter is using. In automatic mode, the scale ratio can change whenever the source size (i.e. viewport) changes or the destination size (i.e. active display of the CRTC output timing).

SCL_DISP2_MODE_CHANGE_INTERRUPT (R)	1	0x0	Interrupt that can be generated by the secondary display controller's scaler when it detects any change in the scale ratio or number of taps the scaling filter is using. In automatic mode, the scale ratio can change whenever the source size (i.e. viewport) changes or the destination size (i.e. active display of the CRTC output timing).
LB_D1_VLINE_INTERRUPT (R)	2	0x0	Interrupt that can be generated by the primary display controller's line buffer logic when the source image line counter falls within a programmed range of line numbers.
LB_D2_VLINE_INTERRUPT (R)	3	0x0	Interrupt that can be generated by the secondary display controller's line buffer logic when the source image line counter falls within a programmed range of line numbers.
LB_D1_VBLANK_INTERRUPT (R)	4	0x0	Interrupt that can be programmed to be generated by the primary display controller's line buffer logic either when the source image line counter is not requesting any active display data (i.e. in the vertical blank) or the output CRTC timing generator is within the vertical blanking region.
LB_D2_VBLANK_INTERRUPT (R)	5	0x0	Interrupt that can be programmed to be generated by the secondary display controller's line buffer logic either when the source image line counter is not requesting any active display data (i.e. in the vertical blank) or the output CRTC timing generator is within the vertical blanking region.
CRTC1_SNAPSHOT_INTERRUPT (R)	6	0x0	Interrupt that can be programmed to be generated by the primary display controller's snapshot logic when either manually forced to trigger by writing a register, or by either a primary CRTC TRIG_A or TRIG_B event occurring.
CRTC1_FORCE_VSYNC_NEXT_LINE_INTERRUPT (R)	7	0x0	Interrupt that can be programmed to be generated by the primary display controller's force VSYNC next line logic when a force VSYNC next line event occurs, caused by either manually writing a register, or by either a primary CRTC TRIG_A or TRIG_B event occurring.
CRTC1_FORCE_COUNT_NOW_INTERRUPT (R)	8	0x0	Interrupt that can be programmed to be generated by the primary display controller's force count now logic when either a primary CRTC TRIG_A or TRIG_B event occur and the horizontal and/or vertical primary CRTC output timing counters reach the H_TOTAL and/or V_TOTAL position selected by the D1CRTC_FORCE_COUNT_NOW_MODE.
CRTC1_TRIGA_INTERRUPT (R)	9	0x0	Interrupt that can be generated by the primary display controller when it detects a primary TRIGA event has occurred.
CRTC1_TRIGB_INTERRUPT (R)	10	0x0	Interrupt that can be generated by the primary display controller when it detects a primary TRIGB event has occurred.
CRTC2_SNAPSHOT_INTERRUPT (R)	11	0x0	Interrupt that can be programmed to be generated by the secondary display controller's snapshot logic when either manually forced to trigger by writing a register, or by either a CRTC TRIG_A or TRIG_B event from the secondary display controller occurring.
CRTC2_FORCE_VSYNC_NEXT_LINE_INTERRUPT (R)	12	0x0	Interrupt that can be programmed to be generated by the secondary display controller's force VSYNC next line logic when a force VSYNC next line event occurs, caused by either manually writing a register, or by either a secondary CRTC TRIG_A or TRIG_B event occurring.
CRTC2_FORCE_COUNT_NOW_INTERRUPT (R)	13	0x0	Interrupt that can be programmed to be generated by the secondary display controller's force count now logic when either a primary CRTC TRIG_A or TRIG_B event occur and the horizontal and/or vertical secondary CRTC output timing counters reach the H_TOTAL and/or V_TOTAL position selected by the D2CRTC_FORCE_COUNT_NOW_MODE.
CRTC2_TRIGA_INTERRUPT (R)	14	0x0	Interrupt that can be generated by the secondary display controller when it detects a secondary TRIGA event has occurred.

CRTC2_TRIGB_INTERRUPT (R)	15	0x0	Interrupt that can be generated by the secondary display controller when it detects a secondary TRIGB event has occurred.
DACA_AUTODETECT_INTERRUPT (R)	16	0x0	Interrupt that can be programmed to be generated when the Autodetect device connected to DACA output detects either a display being first connected or, once connected, first detects the display being disconnected.
DACB_AUTODETECT_INTERRUPT (R)	17	0x0	Interrupt that can be programmed to be generated when the Autodetect device connected to DACA output detects either a display being first connected or, once connected, first detects the display being disconnected.
DC_HOT_PLUG_DETECT1_INTERRUPT (R)	18	0x0	Interrupt that can be programmed to be generated when a Flat Panel (supporting the hot plug feature) is detected to be first connected to the HPD1 pin or, once connected, is detected to have disconnected from the HPD1 pin.
DC_HOT_PLUG_DETECT2_INTERRUPT (R)	19	0x0	Interrupt that can be programmed to be generated when a Flat Panel (supporting the hot plug feature) is detected to be first connected to the HPD2 pin or, once connected, is detected to have disconnected from the HPD2 pin.
DC_I2C_SW_DONE_INTERRUPT (R)	20	0x0	Interrupt that can be generated when the current I2C read or write operation done by the DISPOUT hardware assisted I2C finished execution.
DC_I2C_HW_DONE_INTERRUPT (R)	21	0x0	Interrupt that can be generated when the current I2C read or write operation done by the DISPOUT hardware assisted I2C finishes execution.
DISP_TIMER_INTERRUPT (R)	22	0x0	Interrupt that can be generated when the display Timer Control logic has generated a hardware interrupt.
DACA_CAPTURE_START_INTERRUPT (R)	23	0x0	Interrupt that can be generated each time a start of frame pulse arrives at the DACA output.
DACB_CAPTURE_START_INTERRUPT (R)	24	0x0	Interrupt that can be generated each time a start of frame pulse arrives at the DACB output.
TMDSA_CAPTURE_START_INTERRUPT (R)	25	0x0	Interrupt that can be generated each time a start of frame pulse arrives at the integrated TMDS transmitter output.
LVTMA_CAPTURE_START_INTERRUPT (R)	26	0x0	Interrupt that can be generated each time a start of frame pulse arrives at the integrated LVTM transmitter output.
DVOA_CAPTURE_START_INTERRUPT (R)	27	0x0	Interrupt that can be generated each time a start of frame pulse arrives at the DVOA port.
DISP_INTERRUPT_STATUS_CONTINUE (R)	31	0x0	when this bit is set, continue reading DISP_INTERRUPT_STATUS_CONTINUE

Status of all display block interrupts

DISP_INTERRUPT_STATUS_CONTINUE - RW - 32 bits - [GpuF0MMReg:0x7EE8]			
Field Name	**Bits**	**Default**	**Description**
D1MODE_DATA_UNDERFLOW_INTERRUPT (R)	16	0x0	Debug interrupt for Display 1 Line Buffer data underflow
D1MODE_REQUEST_UNDERFLOW_INTERRUPT (R)	17	0x0	Debug interrupt for Display 1 Line Buffer request underflow
D1SCL_DATA_UNDERFLOW_INTERRUPT (R)	18	0x0	Debug interrupt for Display 1 Scaler data underflow
D1SCL_HOST_CONFLICT_INTERRUPT (R)	19	0x0	Debug interrupt for Display 1 Scaler host conflict
D2MODE_DATA_UNDERFLOW_INTERRUPT (R)	20	0x0	Debug interrupt for Display 2 Line Buffer data underflow
D2MODE_REQUEST_UNDERFLOW_INTERRUPT (R)	21	0x0	Debug interrupt for Display 2 Line Buffer request underflow
D2SCL_DATA_UNDERFLOW_INTERRUPT (R)	22	0x0	Debug interrupt for Display 2 Scaler data underflow

D2SCL_HOST_CONFLICT_INTERRUPT (R)	23	0x0	Debug interrupt for Display 2 Scaler host conflict
MVP_FIFO_ERROR_INTERRUPT (R)	24	0x0	Debug interrupt for multi-vpu fifo error (underflow or overflow)
HDMI0_ERROR_INTERRUPT (R)	26	0x0	Debug interrupt for HDMI0 error (audio fifo overflow, acr tx overflow, audio packet error or vbi packet error)
HDMI1_ERROR_INTERRUPT (R)	27	0x0	Debug interrupt for HDMI1 error (audio fifo overflow, acr tx overflow, audio packet error or vbi packet error)

Status of all display block interrupts

DOUT_POWER_MANAGEMENT_CNTL - RW - 32 bits - [GpuF0MMReg:0x7EE0]			
Field Name	**Bits**	**Default**	**Description**
PWRDN_WAIT_BUSY_OFF	0	0x1	Control whether power management waits for signal indicating all block busy signals =0 from DCCG during powerdown 0=When in WAIT_BUSY_OFF, don't wait for all busy=0 1=When in WAIT_BUSY_OFF, wait for all busy=0
PWRDN_WAIT_PWRSEQ_OFF	4	0x1	Control whether power management waits for signal indicating power sequencer is off during powerdown 0=When in WAIT_BUSY_OFF, don't wait for pwrseq off 1=When in WAIT_BUSY_OFF, wait pwrseq off
PWRDN_WAIT_PPLL_OFF	8	0x0	Control whether power management waits for DCCG to report pixel PLLs are off during powerdown 0=When in WAIT_PPLL_OFF, proceed to next state 1=When in WAIT_PPLL_OFF, wait for pixel pll off indicator
PWRUP_WAIT_PPLL_ON	12	0x0	Control whether power management waits for 1 ms to allow pixel PLLs to lock during powerup 0=When in WAIT_PPLL_ON, proceed to next state 1=When in WAIT_PPLL_ON, wait for 1 ms proceeding to next state
PWRUP_WAIT_MEM_INIT_DONE	16	0x1	Control whether power management mem_init_done indicator 0=When in WAIT_MEM_INIT_DONE, proceed to next state 1=When in WAIT_MEM_INIT_DONE, wait for mem_init_done indicator
PM_ASSERT_RESET	20	0x1	Control whether power management asserts DOUT_CRTC_pwr_down_reset on powerdown 0=Don't assert pm_reset when in 'OFF' state 1=Assert pm_reset when in 'OFF' state
PM_PWRDN_PPLL	24	0x0	Control whether power management asserts pixel PLL reset on powerdown 0=Don't reset pixel PLLs when in 'OFF' state 1=Reset pixel PLLs when in 'OFF' state
PM_CURRENT_STATE (R)	30:28	0x0	Current power management state 0=PM_OFF 1=PM_WAIT_PPLL_ON 2=PM_WAIT_MEM_INIT_DONE 3=PM_ON 4=PM_WAIT_BUSY_OFF 5=PM_WAIT_PPLL_OFF 6=Reserved 7=Reserved

DISP_TIMER_CONTROL - RW - 32 bits - [GpuF0MMReg:0x7EF0]			
Field Name	**Bits**	**Default**	**Description**
DISP_TIMER_INT_COUNT	24:0	0x0	Desired value for Display Timer Counter to count to before generating event that can cause a hardware interrupt to occur. The counter value is decremented each clock pulse of the CG_xtal_ref_sclk signal. CG_xtal_ref_sclk = a one clock wide pulse on core clock (SCLK) that occurs (Crystal Oscillator Frequency (i.e. 27 MHz)) / (CG_RT_CNTL2_DIV) times per second.
DISP_TIMER_INT_ENABLE (W)	25	0x0	0=No effect 1=Start timer interrupt if TIMER_INT_CNT > 0.
DISP_TIMER_INT_RUNNING (R)	26	0x0	0=Timer interrupt counter not running 1=Timer interrupt counter running
DISP_TIMER_INT_MSK	27	0x0	0=Display Countdown Timer cannot generate hardware interrupt. 1=Display Countdown Timer can generate hardware interrupt when count reached.
DISP_TIMER_INT_STAT (R)	28	0x0	Status of the Display Timer Counter logic. When this bit is high, it does not indicate that a hardware interrupt has occurred.
DISP_TIMER_INT_STAT_AK (W)	29	0x0	Write 1 to acknowledge and clear interrupt 0=No effect 1=Interrupt Acknowledged and will be cleared.
DISP_TIMER_INT (R)	30	0x0	When this bit is high, it indicates that the Display Timer Control logic has generated a hardware interrupt. This bit equals the display timer status (DISP_TIMER_INT_STAT) logically 'AND'ed with the display timer interrupt mask (DISP_TIMER_INT_MSK).

Display Countdown Timer capable of generating a hardware interrupt

CAPTURE_START_STATUS - RW - 32 bits - [GpuF0MMReg:0x7ED0]			
Field Name	**Bits**	**Default**	**Description**
DACA_CAPTURE_START (R)	0	0x0	Extended DACA Capture Start that is used for test & debug purposes. This Capture Start is de-asserted by DACA_CAPTURE_START_AK. 0=No event 1=Capture_start has occurred
DACB_CAPTURE_START (R)	1	0x0	Extended DACB Capture Start that is used for test & debug purposes. This Capture Start is de-asserted by DACB_CAPTURE_START_AK. 0=No event 1=Capture_start has occurred
TMDSA_CAPTURE_START (R)	2	0x0	Extended TMDSA Capture Start that is used for test & debug purposes. This Capture Start is de-asserted by TMDSA_CAPTURE_START_AK. 0=No event 1=Capture_start has occurred
LVTMA_CAPTURE_START (R)	3	0x0	0=No event 1=Capture_start has occurred

DVOA_CAPTURE_START (R)	4	0x0	Extended DVOA Capture Start that is used for test & debug purposes. This Capture Start is de-asserted by DVOA_CAPTURE_START_AK. 0=No event 1=Capture_start has occurred
DACA_CAPTURE_START_AK (W)	6	0x0	Acknowledge bit for DACA Capture Start. This bit will clear DACA_CAPTURE_START and DISP_INTERRUPT_STATUS_x.DACA_CAPTURE_START_INTERRUPT. 0=No effect 1=Clear Capture_start
DACB_CAPTURE_START_AK (W)	7	0x0	Acknowledge bit for DACB Capture Start. This bit will clear DACB_CAPTURE_START and DISP_INTERRUPT_STATUS_x.DACB_CAPTURE_START_INTERRUPT. 0=No effect 1=Clear Capture_start
TMDSA_CAPTURE_START_AK (W)	8	0x0	Acknowledge bit for TMDSA Capture Start. This bit will clear TMDSA_CAPTURE_START and DISP_INTERRUPT_STATUS_x.TMDSA_CAPTURE_START_INTERRUPT. 0=No effect 1=Clear Capture_start
LVTMA_CAPTURE_START_AK (W)	9	0x0	0=No effect 1=Clear Capture_start
DVOA_CAPTURE_START_AK (W)	10	0x0	Acknowledge bit for DVOA Capture Start. This bit will clear DVOA_CAPTURE_START and DISP_INTERRUPT_STATUS_x.DVOA_CAPTURE_START_INTERRUPT. 0=No effect 1=Clear Capture_start
DACA_CAPTURE_START_INT_EN	12	0x0	Enable interrupts on DACA Capture Start. Interrupt can be monitored by polling DISP_INTERRUPT_STATUS_x.DACA_CAPTURE_START_INTERRUPT. 0=Disable 1=Enable
DACB_CAPTURE_START_INT_EN	13	0x0	Enable interrupts on DACB Capture Start. Interrupt can be monitored by polling DISP_INTERRUPT_STATUS_x.DACB_CAPTURE_START_INTERRUPT. 0=Disable 1=Enable
TMDSA_CAPTURE_START_INT_EN	14	0x0	Enable interrupts on TMDSA Capture Start. Interrupt can be monitored by polling DISP_INTERRUPT_STATUS_x.TMDSA_CAPTURE_START_INTERRUPT. 0=Disable 1=Enable
LVTMA_CAPTURE_START_INT_EN	15	0x0	0=Disable 1=Enable
DVOA_CAPTURE_START_INT_EN	16	0x0	Enable interrupts on DVOA Capture Start. Interrupt can be monitored by polling DISP_INTERRUPT_STATUS_x.DVOA_CAPTURE_START_INTERRUPT. 0=Disable 1=Enable

Capture Start Control

General Purpose I/O Registers

DC_GPIO_GENERIC_MASK - RW - 32 bits - [GpuF0MMReg:0x7DE0]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_GENERICA_MASK	0	0x0	0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_GENERICB_MASK	8	0x0	0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_GENERICC_MASK	16	0x0	0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.

Control GPIO functionality of GENERIC pads - active high.

DC_GPIO_GENERIC_A - RW - 32 bits - [GpuF0MMReg:0x7DE4]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_GENERICA_A	0	0x0	Asynchronous input for GENERICA pad when DC_GPIO_GENERICA_MASK = 1.
DC_GPIO_GENERICB_A	8	0x0	Asynchronous input for GENERICB pad when DC_GPIO_GENERICB_MASK = 1.
DC_GPIO_GENERICC_A	16	0x0	Asynchronous input for GENERICC pad when DC_GPIO_GENERICC_MASK = 1.

Asynchronous inputs for GENERIC pads when GPIO functionality enabled by DC_GPIO_GENERIC_MASK register.

DC_GPIO_GENERIC_EN - RW - 32 bits - [GpuF0MMReg:0x7DE8]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_GENERICA_EN	0	0x0	Output enable used for GENERICA when DC_GPIO_GENERICA_MASK = 1.
DC_GPIO_GENERICB_EN	8	0x0	Output enable used for GENERICB when DC_GPIO_GENERICB_MASK = 1.
DC_GPIO_GENERICC_EN	16	0x0	Output enable used for GENERICC when DC_GPIO_GENERICC_MASK = 1.

Ouput enable for GENERIC pads when GPIO functionality enabled by DC_GPIO_GENERIC_MASK register.

DC_GPIO_GENERIC_Y - RW - 32 bits - [GpuF0MMReg:0x7DEC]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_GENERICA_Y (R)	0	0x0	Value on GENERICA pad.
DC_GPIO_GENERICB_Y (R)	8	0x0	Value on GENERICB pad.
DC_GPIO_GENERICC_Y (R)	16	0x0	Value on GENERICC pad.

Output values of GENERIC pads.

DC_GPIO_DDC4_MASK - RW - 32 bits - [GpuF0MMReg:0x7E00]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC4CLK_MASK	0	0x0	Enable/Disable GPIO functionality on DDC4CLK pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC4DATA_MASK	8	0x0	Enable/Disable GPIO functionality on DDC4DATA pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.

Control GPIO functionality of the DDC4 pads - all fields are active high.

DC_GPIO_DDC4_A - RW - 32 bits - [GpuF0MMReg:0x7E04]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC4CLK_A	0	0x0	Asynchronous input for DDC4CLK when DC_GPIO_DDC4CLK_MASK = 1.
DC_GPIO_DDC4DATA_A	8	0x0	Asynchronous input for DDC4DATA when DC_GPIO_DDC4DATA_MASK = 1.

Asynchronous inputs for the DDC4 pads when the GPIO functionality is enabled by the DC_GPIO_DDC4_MASK register.

DC_GPIO_DDC4_EN - RW - 32 bits - [GpuF0MMReg:0x7E08]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC4CLK_EN	0	0x0	Output enable for DDC4CLK when DC_GPIO_DDC4CLK_MASK = 1.
DC_GPIO_DDC4DATA_EN	8	0x0	Output enable for DDC4DATA when DC_GPIO_DDC4DATA_MASK = 1.

Output enable values for the DDC4 pads when the GPIO functionality is enabled by the DC_GPIO_DDC4_MASK register.

DC_GPIO_DDC4_Y - RW - 32 bits - [GpuF0MMReg:0x7E0C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC4CLK_Y (R)	0	0x0	Value on DDC4CLK pad.
DC_GPIO_DDC4DATA_Y (R)	8	0x0	Value on DDC4DATA pad.

Output values for the DDC4 pads.

DC_GPIO_DEBUG - RW - 32 bits - [GpuF0MMReg:0x7E2C]

Field Name	Bits	Default	Description
DC_GPIO_VIPGPIO_DEBUG	0	0x1	0. Normal Mode 1. CG/BIF Debug on GPIO[34:18] 0=Normal 1=CG/BIF Debug on GPIO[34:18]
DC_GPIO_MACRO_DEBUG	9:8	0x1	0. Normal Mode 1. Mux Chip/BIF debug bus on DVODATA[23:0] and DVOCNTL[0] 2.Mux TMDS on DVODATA[15:6] 3. Mux LVTM on DVODATA[15:6] 0=Normal 1=Chip/BIF Debug on dvo[23:0] and dvoctrl[0] 2=TMDSA debug output on dvo[15:6] 3=LVTMA debug output on dvo[15:6]

Mux control to allow VIP and DCO Debug to share DVO Pads.

DC_GPIO_DVODATA_MASK - RW - 32 bits - [GpuF0MMReg:0x7E30]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DVODATA_MASK	23:0	0x0	Enable/Disable GPIO functionality on DVODATA pads. Bits can be set individually.
DC_GPIO_DVOCNTL_MASK	26:24	0x0	Enable/Disable GPIO functionality on DVOCNTL pads. Bits can be set individually.
DC_GPIO_DVOCLK_MASK	28	0x0	Enable/Disable GPIO functionality on DVOCLK pads. Bits can be set individually.
DC_GPIO_MVP_DVOCNTL_MASK	31:30	0x0	Enable/Disable GPIO functionality on DVO_MVP_CNTL pads. Bits can be set individually.

Control GPIO functionality of the DVO pads - all fields are active high.

DC_GPIO_DVODATA_A - RW - 32 bits - [GpuF0MMReg:0x7E34]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DVODATA_A	23:0	0x0	Asynchronous inputs for DVODATA pads when associated DC_GPIO_DVODATA_MASK = 1.
DC_GPIO_DVOCNTL_A	26:24	0x0	Asynchronous inputs for DVOCNTL pads when associated DC_GPIO_DVOCNTL_MASK = 1.
DC_GPIO_DVOCLK_A	28	0x0	Asynchronous inputs for DVOCLK pads when associated DC_GPIO_DVOCLK_MASK = 1.
DC_GPIO_MVP_DVOCNTL_A	31:30	0x0	Asynchronous inputs for DVO_MVP_CNTL pads when associated DC_GPIO_MVP_DVOCNTL_MASK = 1.

Asynchronous inputs for the DVO pads when the GPIO functionality is enabled by the DC_GPIO_DVODATA_MASK register.

DC_GPIO_DVODATA_EN - RW - 32 bits - [GpuF0MMReg:0x7E38]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DVODATA_EN	23:0	0x0	Output enables for DVODATA pads when associated DC_GPIO_DVODATA_MASK = 1.
DC_GPIO_DVOCNTL_EN	26:24	0x0	Output enables for DVOCNTL pads when associated DC_GPIO_DVOCNTL_MASK = 1.
DC_GPIO_DVOCLK_EN	28	0x0	Output enables for DVOCLK pads when associated DC_GPIO_DVOCLK_MASK = 1.

DC_GPIO_MVP_DVOCNTL_EN	31:30	0x0	Output enables for DVO_MVP_CNTL pads when associated DC_GPIO_MVP_DVOCNTL_MASK = 1.

Output enable values for the DVO pads when the GPIO functionality is enabled by the DC_GPIO_DVODATA_MASK register.

DC_GPIO_DVODATA_Y - RW - 32 bits - [GpuF0MMReg:0x7E3C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DVODATA_Y (R)	23:0	0x0	Values on DVODATA pads.
DC_GPIO_DVOCNTL_Y (R)	26:24	0x0	Values on DVOCNTL pads.
DC_GPIO_DVOCLK_Y (R)	28	0x0	Values on DVOCLK pads.
DC_GPIO_MVP_DVOCNTL_Y (R)	31:30	0x0	Values on DVO_MVP_CNTL pads.

Output values of the DVO pads.

DC_GPIO_DDC1_MASK - RW - 32 bits - [GpuF0MMReg:0x7E40]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC1CLK_MASK	0	0x0	Enable/Disable GPIO functionality on DDC1CLK pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC1CLK_PD_EN	4	0x0	Set to 1 to enable pulldown on DDC1CLK pad 0=Disable 1=Enable
DC_GPIO_DDC1CLK_PU_EN	6	0x0	Set to 1 to enable pullup on DDC1CLK pad 0=Disable 1=Enable
DC_GPIO_DDC1DATA_MASK	8	0x0	Enable/Disable GPIO functionality on DDC1DATA pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC1DATA_PD_EN	12	0x0	Set to 1 to enable pulldown on DDC1DATA pad 0=Disable 1=Enable
DC_GPIO_DDC1DATA_PU_EN	14	0x0	Set to 1 to enable pullup on DDC1DATA pad 0=Disable 1=Enable

Control GPIO functionality of the DDC1 pads - all fields are active high.

DC_GPIO_DDC1_A - RW - 32 bits - [GpuF0MMReg:0x7E44]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC1CLK_A	0	0x0	Asynchronous input for DDC1CLK when DC_GPIO_DDC1CLK_MASK = 1.
DC_GPIO_DDC1DATA_A	8	0x0	Asynchronous input for DDC1DATA when DC_GPIO_DDC1DATA_MASK = 1.

Asynchronous inputs for the DDC1 pads when the GPIO functionality is enabled by the DC_GPIO_DDC1_MASK register.

DC_GPIO_DDC1_EN - RW - 32 bits - [GpuF0MMReg:0x7E48]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC1CLK_EN	0	0x0	Output enable for DDC1CLK when DC_GPIO_DDC1CLK_MASK = 1.
DC_GPIO_DDC1DATA_EN	8	0x0	Output enable for DDC1DATA when DC_GPIO_DDC1DATA_MASK = 1.

Output enable values for the DDC1 pads when the GPIO functionality is enabled by the DC_GPIO_DDC1_MASK register.

DC_GPIO_DDC1_Y - RW - 32 bits - [GpuF0MMReg:0x7E4C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC1CLK_Y (R)	0	0x0	Value on DDC1CLK pad.
DC_GPIO_DDC1DATA_Y (R)	8	0x0	Value on DDC1DATA pad.

Output values of the DDC1 pads.

DC_GPIO_DDC2_MASK - RW - 32 bits - [GpuF0MMReg:0x7E50]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC2CLK_MASK	0	0x0	Enable/Disable GPIO functionality on DDC2CLK pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC2DATA_MASK	8	0x0	Enable/Disable GPIO functionality on DDC2DATA pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC2DATA_PD_EN	12	0x0	Set to 1 to enable pulldown on DDC2DATA pad 0=Disable 1=Enable
DC_GPIO_DDC2DATA_PU_EN	14	0x0	Set to 1 to enable pullup on DDC2DATA pad 0=Disable 1=Enable

Control GPIO functionality of the DDC2 pads - all fields are active high.

DC_GPIO_DDC2_A - RW - 32 bits - [GpuF0MMReg:0x7E54]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC2CLK_A	0	0x0	Asynchronous input for DDC2CLK when DC_GPIO_DDC2CLK_MASK = 1.
DC_GPIO_DDC2DATA_A	8	0x0	Asynchronous input for DDC2DATA when DC_GPIO_DDC2DATA_MASK = 1.

Asynchronous inputs for the DDC2 pads when the GPIO functionality is enabled by the DC_GPIO_DDC2_MASK register.

DC_GPIO_DDC2_EN - RW - 32 bits - [GpuF0MMReg:0x7E58]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC2CLK_EN	0	0x0	Output enable for DDC2CLK when DC_GPIO_DDC2CLK_MASK = 1.
DC_GPIO_DDC2DATA_EN	8	0x0	Output enable for DDC2DATA when DC_GPIO_DDC2DATA_MASK = 1.

Output enable values for the DDC2 pads when the GPIO functionality is enabled by the DC_GPIO_DDC2_MASK register.

DC_GPIO_DDC2_Y - RW - 32 bits - [GpuF0MMReg:0x7E5C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC2CLK_Y (R)	0	0x0	Value on DDC2CLK pad.
DC_GPIO_DDC2DATA_Y (R)	8	0x0	Value on DDC2DATA pad.

Output values of the DDC2 pads.

DC_GPIO_DDC3_MASK - RW - 32 bits - [GpuF0MMReg:0x7E60]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC3CLK_MASK	0	0x0	Enable/Disable GPIO functionality on DDC3CLK pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC3DATA_MASK	8	0x0	Enable/Disable GPIO functionality on DDC3DATA pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DDC3DATA_PD_EN	12	0x0	Set to 1 to enable pulldown on DDC3DATA pad 0=Disable 1=Enable
DC_GPIO_DDC3DATA_PU_EN	14	0x0	Set to 1 to enable pullup on DDC3DATA pad 0=Disable 1=Enable

Control GPIO functionality of the DDC3 pads - all fields are active high.

DC_GPIO_DDC3_A - RW - 32 bits - [GpuF0MMReg:0x7E64]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC3CLK_A	0	0x0	Asynchronous input for DDC3CLK when DC_GPIO_DDC3CLK_MASK = 1.
DC_GPIO_DDC3DATA_A	8	0x0	Asynchronous input for DDC3DATA when DC_GPIO_DDC3DATA_MASK = 1.

Asynchronous inputs for the DDC3 pads when the GPIO functionality is enabled by the DC_GPIO_DDC3_MASK register.

DC_GPIO_DDC3_EN - RW - 32 bits - [GpuF0MMReg:0x7E68]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC3CLK_EN	0	0x0	Output enable for DDC3CLK when DC_GPIO_DDC3CLK_MASK = 1.
DC_GPIO_DDC3DATA_EN	8	0x0	Output enable for DDC3DATA when DC_GPIO_DDC3DATA_MASK = 1.

Output enable values for the DDC3 pads when the GPIO functionality is enabled by the DC_GPIO_DDC3_MASK register.

DC_GPIO_DDC3_Y - RW - 32 bits - [GpuF0MMReg:0x7E6C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_DDC3CLK_Y (R)	0	0x0	Value on DDC3CLK pad.
DC_GPIO_DDC3DATA_Y (R)	8	0x0	Value on DDC3DATA pad.

Output values for the DDC3 pads.

DC_GPIO_SYNCA_MASK - RW - 32 bits - [GpuF0MMReg:0x7E70]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCA_MASK	0	0x0	Enable/Disable GPIO functionality on HSYNCA pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_HSYNCA_PD_EN	4	0x0	Set to 1 to enable pulldown on HSYNCA pad 0=Disable 1=Enable
DC_GPIO_HSYNCA_PU_EN	6	0x0	Set to 1 to enable pullup on HSYNCA pad 0=Disable 1=Enable
DC_GPIO_VSYNCA_MASK	8	0x0	Enable/Disable GPIO functionality on VSYNCA pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_VSYNCA_PD_EN	12	0x0	Set to 1 to enable pulldown on VSYNCA pad 0=Disable 1=Enable
DC_GPIO_VSYNCA_PU_EN	14	0x0	Set to 1 to enable pullup on VSYNCA pad 0=Disable 1=Enable

Control GPIO functionality of the HSYNCA & VSYNCA pads - all fields are active high.

DC_GPIO_SYNCA_A - RW - 32 bits - [GpuF0MMReg:0x7E74]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCA_A	0	0x0	Asynchronous input for HSYNCA when DC_GPIO_HSYNCA_MASK = 1.
DC_GPIO_VSYNCA_A	8	0x0	Asynchronous input for VSYNCA when DC_GPIO_VSYNCA_MASK = 1.

Asynchronous inputs for the HSYNCA & VSYNCA pads when the GPIO functionality is enabled by the DC_GPIO_SYNCA_MASK register.

DC_GPIO_SYNCA_EN - RW - 32 bits - [GpuF0MMReg:0x7E78]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCA_EN	0	0x0	Output enable for HSYNCA when DC_GPIO_HSYNCA_MASK = 1.
DC_GPIO_VSYNCA_EN	8	0x0	Output enable for VSYNCA when DC_GPIO_VSYNCA_MASK = 1.

Output enable values for the HSYNCA & VSYNCA pads when the GPIO functionality is enabled by the DC_GPIO_SYNCA_MASK register.

DC_GPIO_SYNCA_Y - RW - 32 bits - [GpuF0MMReg:0x7E7C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCA_Y (R)	0	0x0	Value on HSYNCA pad.
DC_GPIO_VSYNCA_Y (R)	8	0x0	Value on VSYNCA pad.

Output values of the HSYNCA & VSYNCA pads.

DC_GPIO_SYNCB_MASK - RW - 32 bits - [GpuF0MMReg:0x7E80]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCB_MASK	0	0x0	Enable/Disable GPIO functionality on HSYNCB pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_VSYNCB_MASK	8	0x0	Enable/Disable GPIO functionality on VSYNCB pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.

Control GPIO functionality of the HSYNCB & VSYNCB pads - all fields are active high.

DC_GPIO_SYNCB_A - RW - 32 bits - [GpuF0MMReg:0x7E84]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCB_A	0	0x0	Asynchronous input for HSYNCB when DC_GPIO_HSYNCB_MASK = 1.
DC_GPIO_VSYNCB_A	8	0x0	Asynchronous input for VSYNCB when DC_GPIO_VSYNCB_MASK = 1.

Asynchronous inputs for the HSYNCB & VSYNCB pads when the GPIO functionality is enabled by the DC_GPIO_SYNCB_MASK register.

DC_GPIO_SYNCB_EN - RW - 32 bits - [GpuF0MMReg:0x7E88]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCB_EN	0	0x0	Output enable for HSYNCB when DC_GPIO_HSYNCB_MASK = 1.
DC_GPIO_VSYNCB_EN	8	0x0	Output enable for VSYNCB when DC_GPIO_VSYNCB_MASK = 1.

Output enable values for the HSYNCB & VSYNCB pads when the GPIO functionality is enabled by the DC_GPIO_SYNCB_MASK register.

DC_GPIO_SYNCB_Y - RW - 32 bits - [GpuF0MMReg:0x7E8C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HSYNCB_Y (R)	0	0x0	Value on HSYNCB pad.
DC_GPIO_VSYNCB_Y (R)	8	0x0	Value on VSYNCB pad.

Output values of the HSYNCB & VSYNCB pads.

DC_GPIO_HPD_MASK - RW - 32 bits - [GpuF0MMReg:0x7E90]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HPD1_MASK	0	0x0	Enable/Disable GPIO functionality on HPD1 pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_HPD1_PD_EN	4	0x0	Set to 1 to enable pulldown on HPD1 pad 0=Disable 1=Enable
DC_GPIO_HPD1_PU_EN	6	0x0	Set to 1 to enable pullup on HPD1 pad 0=Disable 1=Enable
DC_GPIO_HPD2_MASK	8	0x0	Enable/Disable GPIO functionality on HPD2 pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_HPD3_MASK	16	0x0	0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.

Control GPIO functionality of the Hot Plug Detect pads - all fields are active high.

DC_GPIO_HPD_A - RW - 32 bits - [GpuF0MMReg:0x7E94]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HPD1_A	0	0x0	Asynchronous input for HPD1 when DC_GPIO_HPD1_MASK = 1.
DC_GPIO_HPD2_A	8	0x0	Asynchronous input for HPD2 when DC_GPIO_HPD2_MASK = 1.
DC_GPIO_HPD3_A	16	0x0	

Asynchronous inputs for the HPD pads when the GPIO functionality is enabled by the DC_GPIO_HPD_MASK register.

DC_GPIO_HPD_EN - RW - 32 bits - [GpuF0MMReg:0x7E98]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HPD1_EN	0	0x0	Output enable for HPD1 when DC_GPIO_HPD1_MASK = 1.
DC_GPIO_HPD2_EN	8	0x0	Output enable for HPD2 when DC_GPIO_HPD2_MASK = 1.
DC_GPIO_HPD3_EN	16	0x0	

Output enable values for the HPD pads when the GPIO functionality is enabled by the DC_GPIO_HPD_MASK register.

DC_GPIO_HPD_Y - RW - 32 bits - [GpuF0MMReg:0x7E9C]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_HPD1_Y (R)	0	0x0	Value on HPD1 pad.
DC_GPIO_HPD2_Y (R)	8	0x0	Value on HPD2 pad.
DC_GPIO_HPD3_Y (R)	16	0x0	

Output values of the HPD pads.

DC_GPIO_PWRSEQ_MASK - RW - 32 bits - [GpuF0MMReg:0x7EA0]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_BLON_MASK	0	0x0	Enable/Disable GPIO functionality on BLON pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_BLON_PD_EN	4	0x0	Set to 1 to enable pulldown on BLON pad 0=Disable 1=Enable
DC_GPIO_BLON_PU_EN	6	0x0	Set to 1 to enable pullup on BLON pad 0=Disable 1=Enable
DC_GPIO_DIGON_MASK	8	0x0	Enable/Disable GPIO functionality on DIGON pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.
DC_GPIO_DIGON_PD_EN	12	0x0	Set to 1 to enable pulldown on DIGON pad 0=Disable 1=Enable
DC_GPIO_DIGON_PU_EN	14	0x0	Set to 1 to enable pullup on DIGON pad 0=Disable 1=Enable
DC_GPIO_ENA_BL_MASK	16	0x0	Enable/Disable GPIO functionality on ENA_BL pad 0=Pad Driven by Hardware - Normal Operation 1=Pad Controlled by Software through associated GPIO registers. Pad values generated by hardware are overridden.

DC_GPIO_ENA_BL_PD_EN	20	0x0	Set to 1 to enable pulldown on ENA_BL pad 0=Disable 1=Enable
DC_GPIO_ENA_BL_PU_EN	22	0x0	Set to 1 to enable pullup on ENA_BL pad 0=Disable 1=Enable

Control GPIO functionality of the BLON, DIGON & ENA_BL pads - all fields are active high.

DC_GPIO_PWRSEQ_A - RW - 32 bits - [GpuF0MMReg:0x7EA4]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_BLON_A	0	0x0	Asynchronous input for BLON when DC_GPIO_BLON_MASK = 1.
DC_GPIO_DIGON_A	8	0x0	Asynchronous input for BIGON when DC_GPIO_DIGON_MASK = 1.
DC_GPIO_ENA_BL_A	16	0x0	Asynchronous input for BIGON when DC_GPIO_ENA_BL_MASK = 1.

Asynchronous inputs for the BLON, DIGON & ENA_BL pads when the GPIO functionality is enabled by the DC_GPIO_PWRSEQ_MASK register.

DC_GPIO_PWRSEQ_EN - RW - 32 bits - [GpuF0MMReg:0x7EA8]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_BLON_EN	0	0x0	Output enable for BLON when DC_GPIO_BLON_MASK = 1.
DC_GPIO_DIGON_EN	8	0x0	Output enable for DIGON when DC_GPIO_DIGON_MASK = 1.
DC_GPIO_ENA_BL_EN	16	0x0	Output enable for ENA_BL when DC_GPIO_ENA_BL_MASK = 1.

Output enable values for the BLON, DIGON & ENA_BL pads when the GPIO functionality is enabled by the DC_GPIO_PWRSEQ_MASK register.

DC_GPIO_PWRSEQ_Y - RW - 32 bits - [GpuF0MMReg:0x7EAC]			
Field Name	**Bits**	**Default**	**Description**
DC_GPIO_BLON_Y (R)	0	0x0	Value on BLON pad.
DC_GPIO_DIGON_Y (R)	8	0x0	Value on DIGON pad.
DC_GPIO_ENA_BL_Y (R)	16	0x0	Value on ENA_BL pad.

Output values of the BLON, DIGON & ENA_BL pads.

DC_GPIO_PAD_STRENGTH_1 - RW - 32 bits - [GpuF0MMReg:0x7ED4]			
Field Name	**Bits**	**Default**	**Description**
SYNC_STRENGTH_SN	27:24	0x7	Control SN strengths for HSYNCA, HSYNCB, VSYNCA & VSYNCB
SYNC_STRENGTH_SP	31:28	0x4	Control SP strengths for HSYNCA, HSYNCB, VSYNCA & VSYNCB

DC_GPIO_PAD_STRENGTH_2 - RW - 32 bits - [GpuF0MMReg:0x7ED8]			
Field Name	**Bits**	**Default**	**Description**
STRENGTH_SN	3:0	0x7	Control SN strengths for DDC1, DDC2, DDC3, GENERICA, GENERICB, GENERICC, HPD1 & HPD2 pads
STRENGTH_SP	7:4	0x4	Control SP strengths for DDC1, DDC2, DDC3, GENERICA, GENERICB, GENERICC, HPD1 & HPD2 pads
PWRSEQ_STRENGTH_SN	19:16	0x7	Control SN strengths for BLON & DIGON pads
PWRSEQ_STRENGTH_SP	23:20	0x4	Control SP strengths for BLON & DIGON pads

Appendix A
Cross Referenced Index

A.1 Quick Cross-Reference Index

"Configuration Registers Sorted by Name" on page A-2

"Configuration Registers Sorted by Address" on page A-5

"Clock Registers Sorted by Name" on page A-8

"Clock Registers Sorted by Address" on page A-9

"Display Controller Registers Stored by Name" on page A-10

"Display Controller Registers Stored by Address" on page A-26

"Host Interface Decode Space Registers Sorted by Name" on page A-42

"Memory Controller Registers Sorted By Name" on page A-43

"Memory Controller Registers Sorted By Address" on page A-47

"PCIE Registers Sorted By Name" on page A-51

"PCIE Registers Sorted By Address" on page A-53

"VIP Registers Sorted By Name" on page A-55

"VIP Registers Sorted By Address" on page A-58

"VGA ATTR Registers Sorted By Name" on page A-61

"VGA CRT Registers Sorted By Name" on page A-62

"VGA GRPH Registers Sorted By Name" on page A-63

"VGA SEQ Registers Sorted By Name" on page A-64

"All Registers Sorted by Name" on page A-65

A.2 Configuration Registers Sorted by Name

Table A-1 Configuration Registers Sorted by Name

Register Name	Address	Secondary Address	Page
ADAPTER_ID	*AudioPcie\:0x2C*	*GpuF0Pcie\:0x2C* *GpuF1Pcie\:0x2C*	*2-77*
ADAPTER_ID_W	*AudioPcie\:0x4C*	*GpuF0Pcie\:0x4C* *GpuF1Pcie\:0x4C*	*2-77*
BIST	*AudioPcie\:0xF*	*GpuF0Pcie\:0xF* *GpuF1Pcie\:0xF*	*2-76*
CACHE_LINE	*AudioPcie\:0xC*	*GpuF0Pcie\:0xC* *GpuF1Pcie\:0xC*	*2-76*
COMMAND	*AudioPcie\:0x4*	*GpuF0Pcie\:0x4* *GpuF1Pcie\:0x4*	*2-73*
DEVICE_CAP	*AudioPcie\:0x5C*	*GpuF0Pcie\:0x5C* *GpuF1Pcie\:0x5C*	*2-79*
DEVICE_CAP2	*AudioPcie\:0x7C*	*GpuF0Pcie\:0x7C* *GpuF1Pcie\:0x7C*	*2-83*
DEVICE_CNTL	*AudioPcie\:0x60*	*GpuF0Pcie\:0x60* *GpuF1Pcie\:0x60*	*2-80*
DEVICE_CNTL2	*AudioPcie\:0x80*	*GpuF0Pcie\:0x80* *GpuF1Pcie\:0x80*	*2-83*
DEVICE_ID	*AudioPcie\:0x2*	*GpuF0Pcie\:0x2* *GpuF1Pcie\:0x2*	*2-73*
DEVICE_STATUS	*AudioPcie\:0x62*	*GpuF0Pcie\:0x62* *GpuF1Pcie\:0x62*	*2-81*
DEVICE_STATUS2	*AudioPcie\:0x82*	*GpuF0Pcie\:0x82* *GpuF1Pcie\:0x82*	*2-83*
HEADER	*AudioPcie\:0xE*	*GpuF0Pcie\:0xE* *GpuF1Pcie\:0xE*	*2-76*
INTERRUPT_LINE	*AudioPcie\:0x3C*	*GpuF0Pcie\:0x3C* *GpuF1Pcie\:0x3C*	*2-77*
INTERRUPT_PIN	*AudioPcie\:0x3D*	*GpuF0Pcie\:0x3D* *GpuF1Pcie\:0x3D*	*2-77*
LATENCY	*AudioPcie\:0xD*	*GpuF0Pcie\:0xD* *GpuF1Pcie\:0xD*	*2-76*
LINK_CAP	*AudioPcie\:0x64*	*GpuF0Pcie\:0x64* *GpuF1Pcie\:0x64*	*2-81*
LINK_CAP2	*AudioPcie\:0x84*	*GpuF0Pcie\:0x84* *GpuF1Pcie\:0x84*	*2-83*
LINK_CNTL	*AudioPcie\:0x68*	*GpuF0Pcie\:0x68* *GpuF1Pcie\:0x68*	*2-82*
LINK_CNTL2	*AudioPcie\:0x88*	*GpuF0Pcie\:0x88* *GpuF1Pcie\:0x88*	*2-84*
LINK_STATUS	*AudioPcie\:0x6A*	*GpuF0Pcie\:0x6A* *GpuF1Pcie\:0x6A*	*2-82*
LINK_STATUS2	*AudioPcie\:0x8A*	*GpuF0Pcie\:0x8A* *GpuF1Pcie\:0x8A*	*2-84*

Table A-1 Configuration Registers Sorted by Name (Continued)

Register Name	Address	Secondary Address	Page
MAX_LATENCY	*AudioPcie\:0x3F*	*GpuF0Pcie\:0x3F* *GpuF1Pcie\:0x3F*	*2-77*
MIN_GRANT	*AudioPcie\:0x3E*	*GpuF0Pcie\:0x3E* *GpuF1Pcie\:0x3E*	*2-77*
MSI_CAP_LIST	*AudioPcie\:0xA0*	*GpuF0Pcie\:0xA0* *GpuF1Pcie\:0xA0*	*2-84*
MSI_MSG_ADDR_HI	*AudioPcie\:0xA8*	*GpuF0Pcie\:0xA8* *GpuF1Pcie\:0xA8*	*2-85*
MSI_MSG_ADDR_LO	*AudioPcie\:0xA4*	*GpuF0Pcie\:0xA4* *GpuF1Pcie\:0xA4*	*2-85*
MSI_MSG_CNTL	*AudioPcie\:0xA2*	*GpuF0Pcie\:0xA2* *GpuF1Pcie\:0xA2*	*2-84*
MSI_MSG_DATA	*AudioPcie\:0xA8*	*GpuF0Pcie\:0xA8* *GpuF1Pcie\:0xA8*	*2-85*
MSI_MSG_DATA_64	*AudioPcie\:0xAC*	*GpuF0Pcie\:0xAC* *GpuF1Pcie\:0xAC*	*2-85*
PCIE_ADV_ERR_CAP_CNTL	*AudioPcie\:0x168*	*GpuF0Pcie\:0x168* *GpuF1Pcie\:0x168*	*2-87*
PCIE_ADV_ERR_RPT_ENH_CAP_LIST	*AudioPcie\:0x150*	*GpuF0Pcie\:0x150* *GpuF1Pcie\:0x150*	*2-85*
PCIE_CAP	*AudioPcie\:0x5A*	*GpuF0Pcie\:0x5A* *GpuF1Pcie\:0x5A*	*2-79*
PCIE_CAP_LIST	*AudioPcie\:0x58*	*GpuF0Pcie\:0x58* *GpuF1Pcie\:0x58*	*2-78*
PCIE_CORR_ERR_MASK	*AudioPcie\:0x164*	*GpuF0Pcie\:0x164* *GpuF1Pcie\:0x164*	*2-87*
PCIE_CORR_ERR_STATUS	*AudioPcie\:0x160*	*GpuF0Pcie\:0x160* *GpuF1Pcie\:0x160*	*2-87*
PCIE_HDR_LOG0	*AudioPcie\:0x16C*	*GpuF0Pcie\:0x16C* *GpuF1Pcie\:0x16C*	*2-87*
PCIE_HDR_LOG1	*AudioPcie\:0x170*	*GpuF0Pcie\:0x170* *GpuF1Pcie\:0x170*	*2-88*
PCIE_HDR_LOG2	*AudioPcie\:0x174*	*GpuF0Pcie\:0x174* *GpuF1Pcie\:0x174*	*2-88*
PCIE_HDR_LOG3	*AudioPcie\:0x178*	*GpuF0Pcie\:0x178* *GpuF1Pcie\:0x178*	*2-88*
PCIE_UNCORR_ERR_MASK	*AudioPcie\:0x158*	*GpuF0Pcie\:0x158* *GpuF1Pcie\:0x158*	*2-86*
PCIE_UNCORR_ERR_SEVERITY	*AudioPcie\:0x15C*	*GpuF0Pcie\:0x15C* *GpuF1Pcie\:0x15C*	*2-86*
PCIE_UNCORR_ERR_STATUS	*AudioPcie\:0x154*	*GpuF0Pcie\:0x154* *GpuF1Pcie\:0x154*	*2-86*
REVISION_ID	*AudioPcie\:0x8*	*GpuF0Pcie\:0x8* *GpuF1Pcie\:0x8*	*2-75*
STATUS	*AudioPcie\:0x6*	*GpuF0Pcie\:0x6* *GpuF1Pcie\:0x6*	*2-74*

Table A-1 Configuration Registers Sorted by Name (Continued)

Register Name	Address	Secondary Address	Page
SUB_CLASS	*AudioPcie\:0xA*	*GpuF0Pcie\:0xA* *GpuF1Pcie\:0xA*	*2-75*
VENDOR_ID	*AudioPcie\:0x0*	*GpuF0Pcie\:0x0* *GpuF1Pcie\:0x0*	*2-73*

A.3 Configuration Registers Sorted by Address

Table A-2 Configuration Registers Sorted by Address

Register Name	Address	Secondary Address	Page
VENDOR_ID	*AudioPcie\:0x0*	*GpuF0Pcie\:0x0* *GpuF1Pcie\:0x0*	*2-73*
PCIE_ADV_ERR_RPT_ENH_CAP_LIST	*AudioPcie\:0x150*	*GpuF0Pcie\:0x150* *GpuF1Pcie\:0x150*	*2-85*
PCIE_UNCORR_ERR_STATUS	*AudioPcie\:0x154*	*GpuF0Pcie\:0x154* *GpuF1Pcie\:0x154*	*2-86*
PCIE_UNCORR_ERR_MASK	*AudioPcie\:0x158*	*GpuF0Pcie\:0x158* *GpuF1Pcie\:0x158*	*2-86*
PCIE_UNCORR_ERR_SEVERITY	*AudioPcie\:0x15C*	*GpuF0Pcie\:0x15C* *GpuF1Pcie\:0x15C*	*2-86*
PCIE_CORR_ERR_STATUS	*AudioPcie\:0x160*	*GpuF0Pcie\:0x160* *GpuF1Pcie\:0x160*	*2-87*
PCIE_CORR_ERR_MASK	*AudioPcie\:0x164*	*GpuF0Pcie\:0x164* *GpuF1Pcie\:0x164*	*2-87*
PCIE_ADV_ERR_CAP_CNTL	*AudioPcie\:0x168*	*GpuF0Pcie\:0x168* *GpuF1Pcie\:0x168*	*2-87*
PCIE_HDR_LOG0	*AudioPcie\:0x16C*	*GpuF0Pcie\:0x16C* *GpuF1Pcie\:0x16C*	*2-87*
PCIE_HDR_LOG1	*AudioPcie\:0x170*	*GpuF0Pcie\:0x170* *GpuF1Pcie\:0x170*	*2-88*
PCIE_HDR_LOG2	*AudioPcie\:0x174*	*GpuF0Pcie\:0x174* *GpuF1Pcie\:0x174*	*2-88*
PCIE_HDR_LOG3	*AudioPcie\:0x178*	*GpuF0Pcie\:0x178* *GpuF1Pcie\:0x178*	*2-88*
DEVICE_ID	*AudioPcie\:0x2*	*GpuF0Pcie\:0x2* *GpuF1Pcie\:0x2*	*2-73*
ADAPTER_ID	*AudioPcie\:0x2C*	*GpuF0Pcie\:0x2C* *GpuF1Pcie\:0x2C*	*2-77*
INTERRUPT_LINE	*AudioPcie\:0x3C*	*GpuF0Pcie\:0x3C* *GpuF1Pcie\:0x3C*	*2-77*
INTERRUPT_PIN	*AudioPcie\:0x3D*	*GpuF0Pcie\:0x3D* *GpuF1Pcie\:0x3D*	*2-77*
MIN_GRANT	*AudioPcie\:0x3E*	*GpuF0Pcie\:0x3E* *GpuF1Pcie\:0x3E*	*2-77*
MAX_LATENCY	*AudioPcie\:0x3F*	*GpuF0Pcie\:0x3F* *GpuF1Pcie\:0x3F*	*2-77*
COMMAND	*AudioPcie\:0x4*	*GpuF0Pcie\:0x4* *GpuF1Pcie\:0x4*	*2-73*
ADAPTER_ID_W	*AudioPcie\:0x4C*	*GpuF0Pcie\:0x4C* *GpuF1Pcie\:0x4C*	*2-77*
PCIE_CAP_LIST	*AudioPcie\:0x58*	*GpuF0Pcie\:0x58* *GpuF1Pcie\:0x58*	*2-78*
PCIE_CAP	*AudioPcie\:0x5A*	*GpuF0Pcie\:0x5A* *GpuF1Pcie\:0x5A*	*2-79*

Table A-2 Configuration Registers Sorted by Address (Continued)

Register Name	Address	Secondary Address	Page
DEVICE_CAP	AudioPcie\:0x5C	GpuF0Pcie\:0x5C GpuF1Pcie\:0x5C	2-79
STATUS	AudioPcie\:0x6	GpuF0Pcie\:0x6 GpuF1Pcie\:0x6	2-74
DEVICE_CNTL	AudioPcie\:0x60	GpuF0Pcie\:0x60 GpuF1Pcie\:0x60	2-80
DEVICE_STATUS	AudioPcie\:0x62	GpuF0Pcie\:0x62 GpuF1Pcie\:0x62	2-81
LINK_CAP	AudioPcie\:0x64	GpuF0Pcie\:0x64 GpuF1Pcie\:0x64	2-81
LINK_CNTL	AudioPcie\:0x68	GpuF0Pcie\:0x68 GpuF1Pcie\:0x68	2-82
LINK_STATUS	AudioPcie\:0x6A	GpuF0Pcie\:0x6A GpuF1Pcie\:0x6A	2-82
DEVICE_CAP2	AudioPcie\:0x7C	GpuF0Pcie\:0x7C GpuF1Pcie\:0x7C	2-83
REVISION_ID	AudioPcie\:0x8	GpuF0Pcie\:0x8 GpuF1Pcie\:0x8	2-75
DEVICE_CNTL2	AudioPcie\:0x80	GpuF0Pcie\:0x80 GpuF1Pcie\:0x80	2-83
DEVICE_STATUS2	AudioPcie\:0x82	GpuF0Pcie\:0x82 GpuF1Pcie\:0x82	2-83
LINK_CAP2	AudioPcie\:0x84	GpuF0Pcie\:0x84 GpuF1Pcie\:0x84	2-83
LINK_CNTL2	AudioPcie\:0x88	GpuF0Pcie\:0x88 GpuF1Pcie\:0x88	2-84
LINK_STATUS2	AudioPcie\:0x8A	GpuF0Pcie\:0x8A GpuF1Pcie\:0x8A	2-84
SUB_CLASS	AudioPcie\:0xA	GpuF0Pcie\:0xA GpuF1Pcie\:0xA	2-75
MSI_CAP_LIST	AudioPcie\:0xA0	GpuF0Pcie\:0xA0 GpuF1Pcie\:0xA0	2-84
MSI_MSG_CNTL	AudioPcie\:0xA2	GpuF0Pcie\:0xA2 GpuF1Pcie\:0xA2	2-84
MSI_MSG_ADDR_LO	AudioPcie\:0xA4	GpuF0Pcie\:0xA4 GpuF1Pcie\:0xA4	2-85
MSI_MSG_ADDR_HI	AudioPcie\:0xA8	GpuF0Pcie\:0xA8 GpuF1Pcie\:0xA8	2-85
MSI_MSG_DATA	AudioPcie\:0xA8	GpuF0Pcie\:0xA8 GpuF1Pcie\:0xA8	2-85
MSI_MSG_DATA_64	AudioPcie\:0xAC	GpuF0Pcie\:0xAC GpuF1Pcie\:0xAC	2-85
CACHE_LINE	AudioPcie\:0xC	GpuF0Pcie\:0xC GpuF1Pcie\:0xC	2-76
LATENCY	AudioPcie\:0xD	GpuF0Pcie\:0xD GpuF1Pcie\:0xD	2-76

Table A-2 Configuration Registers Sorted by Address (Continued)

Register Name	Address	Secondary Address	Page
HEADER	*AudioPcie\:0xE*	*GpuF0Pcie\:0xE* *GpuF1Pcie\:0xE*	*2-76*
BIST	*AudioPcie\:0xF*	*GpuF0Pcie\:0xF* *GpuF1Pcie\:0xF*	*2-76*

A.4 Clock Registers Sorted by Name

Table A-3 Clock Registers Sorted by Name

Register Name	Address	Page
CG_CLKPIN_CNTL	*GpuF0MMReg\:0x644*	*2-94*
CG_MISC_REG	*GpuF0MMReg\:0x7C8*	*2-95*
CG_MPLL_SPREAD_SPECTRUM	*GpuF0MMReg\:0x830*	*2-96*
CG_SPLL_SPREAD_SPECTRUM_CTXSW	*GpuF0MMReg\:0x82C*	*2-96*
CG_SPLL_SPREAD_SPECTRUM_HIGH	*GpuF0MMReg\:0x828*	*2-95*
CG_SPLL_SPREAD_SPECTRUM_LOW	*GpuF0MMReg\:0x820*	*2-95*
CG_SPLL_SPREAD_SPECTRUM_MED	*GpuF0MMReg\:0x824*	*2-95*
CG_TC_JTAG_0	*GpuF0MMReg\:0x7A0*	*2-94*
CG_TC_JTAG_1	*GpuF0MMReg\:0x7A4*	*2-95*
CG_UPLL_SPREAD_SPECTRUM	*GpuF0MMReg\:0x834*	*2-96*
DLL_CNTL	*GpuF0MMReg\:0x62C*	*2-92*
ERROR_STATUS	*GpuF0MMReg\:0x640*	*2-93*
GENERAL_PWRMGT	*GpuF0MMReg\:0x618*	*2-90*
MCLK_PWRMGT_CNTL	*GpuF0MMReg\:0x624*	*2-91*
MPLL_CNTL_MODE	*GpuF0MMReg\:0x614*	*2-90*
MPLL_FUNC_CNTL	*GpuF0MMReg\:0x610*	*2-89*
MPLL_TIME	*GpuF0MMReg\:0x634*	*2-93*
PLL_BYPASSCLK_SEL	*GpuF0MMReg\:0x608*	*2-89*
PLL_TEST_CNTL	*GpuF0MMReg\:0x79C*	*2-94*
SCLK_PWRMGT_CNTL	*GpuF0MMReg\:0x620*	*2-91*
SPLL_CNTL_MODE	*GpuF0MMReg\:0x60C*	*2-89*
SPLL_TIME	*GpuF0MMReg\:0x630*	*2-93*

A.5 Clock Registers Sorted by Address

Table A-4 Clock Registers Sorted by Address

Register Name	Address	Page
PLL_BYPASSCLK_SEL	*GpuF0MMReg\:0x608*	*2-89*
SPLL_CNTL_MODE	*GpuF0MMReg\:0x60C*	*2-89*
MPLL_FUNC_CNTL	*GpuF0MMReg\:0x610*	*2-89*
MPLL_CNTL_MODE	*GpuF0MMReg\:0x614*	*2-90*
GENERAL_PWRMGT	*GpuF0MMReg\:0x618*	*2-90*
SCLK_PWRMGT_CNTL	*GpuF0MMReg\:0x620*	*2-91*
MCLK_PWRMGT_CNTL	*GpuF0MMReg\:0x624*	*2-91*
DLL_CNTL	*GpuF0MMReg\:0x62C*	*2-92*
SPLL_TIME	*GpuF0MMReg\:0x630*	*2-93*
MPLL_TIME	*GpuF0MMReg\:0x634*	*2-93*
ERROR_STATUS	*GpuF0MMReg\:0x640*	*2-93*
CG_CLKPIN_CNTL	*GpuF0MMReg\:0x644*	*2-94*
PLL_TEST_CNTL	*GpuF0MMReg\:0x79C*	*2-94*
CG_TC_JTAG_0	*GpuF0MMReg\:0x7A0*	*2-94*
CG_TC_JTAG_1	*GpuF0MMReg\:0x7A4*	*2-95*
CG_MISC_REG	*GpuF0MMReg\:0x7C8*	*2-95*
CG_SPLL_SPREAD_SPECTRUM_LOW	*GpuF0MMReg\:0x820*	*2-95*
CG_SPLL_SPREAD_SPECTRUM_MED	*GpuF0MMReg\:0x824*	*2-95*
CG_SPLL_SPREAD_SPECTRUM_HIGH	*GpuF0MMReg\:0x828*	*2-95*
CG_SPLL_SPREAD_SPECTRUM_CTXSW	*GpuF0MMReg\:0x82C*	*2-96*
CG_MPLL_SPREAD_SPECTRUM	*GpuF0MMReg\:0x830*	*2-96*
CG_UPLL_SPREAD_SPECTRUM	*GpuF0MMReg\:0x834*	*2-96*

A.6 Display Controller Registers Stored by Name

Table A-5 Display Controller Registers Sorted by Name

Register Name	Address	Page
ATTRDR	*GpuF0MMReg\:0x3C1* *VGA_IO\:0x3C1*	*2-154*
ATTRDW	*GpuF0MMReg\:0x3C0* *VGA_IO\:0x3C0*	*2-154*
ATTRX	*GpuF0MMReg\:0x3C0* *VGA_IO\:0x3C0*	*2-153*
CAPTURE_START_STATUS	*GpuF0MMReg\:0x7ED0*	*2-347*
CRTC8_DATA	*GpuF0MMReg\:0x3B5* *GpuF0MMReg\:0x3D5* *VGA_IO\:0x3B5* *VGA_IO\:0x3D5*	*2-144*
CRTC8_IDX	*GpuF0MMReg\:0x3B4* *GpuF0MMReg\:0x3D4* *VGA_IO\:0x3B4* *VGA_IO\:0x3D4*	*2-144*
D1_MVP_AFR_FLIP_FIFO_CNTL	*GpuF0MMReg\:0x6518*	*2-207*
D1_MVP_AFR_FLIP_MODE	*GpuF0MMReg\:0x6514*	*2-207*
D1_MVP_FLIP_LINE_NUM_INSERT	*GpuF0MMReg\:0x651C*	*2-208*
D1COLOR_MATRIX_COEF_1_1	*GpuF0MMReg\:0x6384*	*2-192*
D1COLOR_MATRIX_COEF_1_2	*GpuF0MMReg\:0x6388*	*2-193*
D1COLOR_MATRIX_COEF_1_3	*GpuF0MMReg\:0x638C*	*2-193*
D1COLOR_MATRIX_COEF_1_4	*GpuF0MMReg\:0x6390*	*2-193*
D1COLOR_MATRIX_COEF_2_1	*GpuF0MMReg\:0x6394*	*2-193*
D1COLOR_MATRIX_COEF_2_2	*GpuF0MMReg\:0x6398*	*2-194*
D1COLOR_MATRIX_COEF_2_3	*GpuF0MMReg\:0x639C*	*2-194*
D1COLOR_MATRIX_COEF_2_4	*GpuF0MMReg\:0x63A0*	*2-194*
D1COLOR_MATRIX_COEF_3_1	*GpuF0MMReg\:0x63A4*	*2-194*
D1COLOR_MATRIX_COEF_3_2	*GpuF0MMReg\:0x63A8*	*2-195*
D1COLOR_MATRIX_COEF_3_3	*GpuF0MMReg\:0x63AC*	*2-195*
D1COLOR_MATRIX_COEF_3_4	*GpuF0MMReg\:0x63B0*	*2-195*
D1COLOR_SPACE_CONVERT	*GpuF0MMReg\:0x613C*	*2-195*
D1CRTC_BLACK_COLOR	*GpuF0MMReg\:0x6098*	*2-265*
D1CRTC_BLANK_CONTROL	*GpuF0MMReg\:0x6084*	*2-263*
D1CRTC_BLANK_DATA_COLOR	*GpuF0MMReg\:0x6090*	*2-264*
D1CRTC_CONTROL	*GpuF0MMReg\:0x6080*	*2-263*
D1CRTC_COUNT_CONTROL	*GpuF0MMReg\:0x60B4*	*2-266*
D1CRTC_COUNT_RESET	*GpuF0MMReg\:0x60B0*	*2-266*
D1CRTC_DOUBLE_BUFFER_CONTROL	*GpuF0MMReg\:0x60EC*	*2-270*
D1CRTC_FLOW_CONTROL	*GpuF0MMReg\:0x6074*	*2-261*
D1CRTC_FORCE_COUNT_NOW_CNTL	*GpuF0MMReg\:0x6070*	*2-261*
D1CRTC_H_BLANK_START_END	*GpuF0MMReg\:0x6004*	*2-255*
D1CRTC_H_SYNC_A	*GpuF0MMReg\:0x6008*	*2-256*
D1CRTC_H_SYNC_A_CNTL	*GpuF0MMReg\:0x600C*	*2-256*

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D1CRTC_H_SYNC_B	GpuF0MMReg\:0x6010	2-256
D1CRTC_H_SYNC_B_CNTL	GpuF0MMReg\:0x6014	2-257
D1CRTC_H_TOTAL	GpuF0MMReg\:0x6000	2-255
D1CRTC_INTERLACE_CONTROL	GpuF0MMReg\:0x6088	2-264
D1CRTC_INTERLACE_STATUS	GpuF0MMReg\:0x608C	2-264
D1CRTC_INTERRUPT_CONTROL	GpuF0MMReg\:0x60DC	2-267
D1CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE	GpuF0MMReg\:0x60B8	2-267
D1CRTC_MVP_BLACK_KEYER	GpuF0MMReg\:0x6058	2-205
D1CRTC_MVP_CONTROL1	GpuF0MMReg\:0x6038	2-202
D1CRTC_MVP_CONTROL2	GpuF0MMReg\:0x603C	2-203
D1CRTC_MVP_CONTROL3	GpuF0MMReg\:0x6850	2-206
D1CRTC_MVP_CRC_CNTL	GpuF0MMReg\:0x6840	2-205
D1CRTC_MVP_CRC_RESULT	GpuF0MMReg\:0x6844	2-206
D1CRTC_MVP_CRC2_CNTL	GpuF0MMReg\:0x6848	2-206
D1CRTC_MVP_CRC2_RESULT	GpuF0MMReg\:0x684C	2-206
D1CRTC_MVP_FIFO_CONTROL	GpuF0MMReg\:0x6040	2-203
D1CRTC_MVP_FIFO_STATUS	GpuF0MMReg\:0x6044	2-203
D1CRTC_MVP_INBAND_CNTL_CAP	GpuF0MMReg\:0x604C	2-204
D1CRTC_MVP_INBAND_CNTL_INSERT	GpuF0MMReg\:0x6050	2-204
D1CRTC_MVP_INBAND_CNTL_INSERT_TIMER	GpuF0MMReg\:0x6054	2-204
D1CRTC_MVP_RECEIVE_CNT_CNTL1	GpuF0MMReg\:0x6854	2-207
D1CRTC_MVP_RECEIVE_CNT_CNTL2	GpuF0MMReg\:0x6858	2-207
D1CRTC_MVP_SLAVE_STATUS	GpuF0MMReg\:0x6048	2-204
D1CRTC_MVP_STATUS	GpuF0MMReg\:0x605C	2-205
D1CRTC_OVERSCAN_COLOR	GpuF0MMReg\:0x6094	2-265
D1CRTC_PIXEL_DATA_READBACK	GpuF0MMReg\:0x6078	2-262
D1CRTC_SNAPSHOT_CONTROL	GpuF0MMReg\:0x60CC	2-268
D1CRTC_SNAPSHOT_FRAME	GpuF0MMReg\:0x60D4	2-268
D1CRTC_SNAPSHOT_POSITION	GpuF0MMReg\:0x60D0	2-268
D1CRTC_SNAPSHOT_STATUS	GpuF0MMReg\:0x60C8	2-268
D1CRTC_START_LINE_CONTROL	GpuF0MMReg\:0x60D8	2-269
D1CRTC_STATUS	GpuF0MMReg\:0x609C	2-265
D1CRTC_STATUS_FRAME_COUNT	GpuF0MMReg\:0x60A4	2-266
D1CRTC_STATUS_HV_COUNT	GpuF0MMReg\:0x60AC	2-266
D1CRTC_STATUS_POSITION	GpuF0MMReg\:0x60A0	2-265
D1CRTC_STATUS_VF_COUNT	GpuF0MMReg\:0x60A8	2-266
D1CRTC_STEREO_CONTROL	GpuF0MMReg\:0x60C4	2-267
D1CRTC_STEREO_FORCE_NEXT_EYE	GpuF0MMReg\:0x607C	2-262
D1CRTC_STEREO_STATUS	GpuF0MMReg\:0x60C0	2-267
D1CRTC_TRIGA_CNTL	GpuF0MMReg\:0x6060	2-258
D1CRTC_TRIGA_MANUAL_TRIG	GpuF0MMReg\:0x6064	2-260
D1CRTC_TRIGB_CNTL	GpuF0MMReg\:0x6068	2-260
D1CRTC_TRIGB_MANUAL_TRIG	GpuF0MMReg\:0x606C	2-261

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D1CRTC_UPDATE_LOCK	GpuF0MMReg\:0x60E8	2-270
D1CRTC_V_BLANK_START_END	GpuF0MMReg\:0x6024	2-257
D1CRTC_V_SYNC_A	GpuF0MMReg\:0x6028	2-257
D1CRTC_V_SYNC_A_CNTL	GpuF0MMReg\:0x602C	2-258
D1CRTC_V_SYNC_B	GpuF0MMReg\:0x6030	2-258
D1CRTC_V_SYNC_B_CNTL	GpuF0MMReg\:0x6034	2-258
D1CRTC_V_TOTAL	GpuF0MMReg\:0x6020	2-257
D1CRTC_VERT_SYNC_CONTROL	GpuF0MMReg\:0x60BC	2-267
D1CRTC_VGA_PARAMETER_CAPTURE_MODE	GpuF0MMReg\:0x60F0	2-271
D1CUR_COLOR1	GpuF0MMReg\:0x641C	2-199
D1CUR_COLOR2	GpuF0MMReg\:0x6420	2-199
D1CUR_CONTROL	GpuF0MMReg\:0x6400	2-197
D1CUR_HOT_SPOT	GpuF0MMReg\:0x6418	2-199
D1CUR_POSITION	GpuF0MMReg\:0x6414	2-198
D1CUR_SIZE	GpuF0MMReg\:0x6410	2-198
D1CUR_SURFACE_ADDRESS	GpuF0MMReg\:0x6408	2-198
D1CUR_UPDATE	GpuF0MMReg\:0x6424	2-199
D1GRPH_ALPHA	GpuF0MMReg\:0x6304	2-189
D1GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6380	2-192
D1GRPH_CONTROL	GpuF0MMReg\:0x6104	2-170
D1GRPH_ENABLE	GpuF0MMReg\:0x6100	2-170
D1GRPH_FLIP_CONTROL	GpuF0MMReg\:0x6148	2-176
D1GRPH_KEY_RANGE_ALPHA	GpuF0MMReg\:0x631C	2-191
D1GRPH_KEY_RANGE_BLUE	GpuF0MMReg\:0x6318	2-190
D1GRPH_KEY_RANGE_GREEN	GpuF0MMReg\:0x6314	2-190
D1GRPH_KEY_RANGE_RED	GpuF0MMReg\:0x6310	2-190
D1GRPH_LUT_SEL	GpuF0MMReg\:0x6108	2-171
D1GRPH_PITCH	GpuF0MMReg\:0x6120	2-173
D1GRPH_PRIMARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6110	2-172
D1GRPH_SECONDARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6118	2-173
D1GRPH_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x614C	2-176
D1GRPH_SURFACE_OFFSET_X	GpuF0MMReg\:0x6124	2-173
D1GRPH_SURFACE_OFFSET_Y	GpuF0MMReg\:0x6128	2-173
D1GRPH_SWAP_CNTL	GpuF0MMReg\:0x610C	2-172
D1GRPH_UPDATE	GpuF0MMReg\:0x6144	2-174
D1GRPH_X_END	GpuF0MMReg\:0x6134	2-174
D1GRPH_X_START	GpuF0MMReg\:0x612C	2-173
D1GRPH_Y_END	GpuF0MMReg\:0x6138	2-174
D1GRPH_Y_START	GpuF0MMReg\:0x6130	2-174
D1ICON_COLOR1	GpuF0MMReg\:0x6458	2-201
D1ICON_COLOR2	GpuF0MMReg\:0x645C	2-201
D1ICON_CONTROL	GpuF0MMReg\:0x6440	2-200
D1ICON_SIZE	GpuF0MMReg\:0x6450	2-201

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D1ICON_START_POSITION	GpuF0MMReg\:0x6454	2-201
D1ICON_SURFACE_ADDRESS	GpuF0MMReg\:0x6448	2-200
D1ICON_UPDATE	GpuF0MMReg\:0x6460	2-202
D1MODE_MASTER_UPDATE_LOCK	GpuF0MMReg\:0x60E0	2-269
D1MODE_MASTER_UPDATE_MODE	GpuF0MMReg\:0x60E4	2-270
D1OVL_ALPHA	GpuF0MMReg\:0x6308	2-189
D1OVL_ALPHA_CONTROL	GpuF0MMReg\:0x630C	2-189
D1OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6140	2-181
D1OVL_CONTROL1	GpuF0MMReg\:0x6184	2-176
D1OVL_CONTROL2	GpuF0MMReg\:0x6188	2-177
D1OVL_DFQ_CONTROL	GpuF0MMReg\:0x61B4	2-180
D1OVL_DFQ_STATUS	GpuF0MMReg\:0x61B8	2-181
D1OVL_ENABLE	GpuF0MMReg\:0x6180	2-176
D1OVL_END	GpuF0MMReg\:0x61A8	2-179
D1OVL_KEY_ALPHA	GpuF0MMReg\:0x632C	2-192
D1OVL_KEY_CONTROL	GpuF0MMReg\:0x6300	2-188
D1OVL_KEY_RANGE_BLUE_CB	GpuF0MMReg\:0x6328	2-191
D1OVL_KEY_RANGE_GREEN_Y	GpuF0MMReg\:0x6324	2-191
D1OVL_KEY_RANGE_RED_CR	GpuF0MMReg\:0x6320	2-191
D1OVL_MATRIX_COEF_1_1	GpuF0MMReg\:0x6204	2-181
D1OVL_MATRIX_COEF_1_2	GpuF0MMReg\:0x6208	2-182
D1OVL_MATRIX_COEF_1_3	GpuF0MMReg\:0x620C	2-182
D1OVL_MATRIX_COEF_1_4	GpuF0MMReg\:0x6210	2-182
D1OVL_MATRIX_COEF_2_1	GpuF0MMReg\:0x6214	2-182
D1OVL_MATRIX_COEF_2_2	GpuF0MMReg\:0x6218	2-183
D1OVL_MATRIX_COEF_2_3	GpuF0MMReg\:0x621C	2-183
D1OVL_MATRIX_COEF_2_4	GpuF0MMReg\:0x6220	2-183
D1OVL_MATRIX_COEF_3_1	GpuF0MMReg\:0x6224	2-183
D1OVL_MATRIX_COEF_3_2	GpuF0MMReg\:0x6228	2-184
D1OVL_MATRIX_COEF_3_3	GpuF0MMReg\:0x622C	2-184
D1OVL_MATRIX_COEF_3_4	GpuF0MMReg\:0x6230	2-184
D1OVL_MATRIX_TRANSFORM_EN	GpuF0MMReg\:0x6200	2-181
D1OVL_PITCH	GpuF0MMReg\:0x6198	2-178
D1OVL_PWL_0TOF	GpuF0MMReg\:0x6284	2-184
D1OVL_PWL_100TO13F	GpuF0MMReg\:0x629C	2-186
D1OVL_PWL_10TO1F	GpuF0MMReg\:0x6288	2-185
D1OVL_PWL_140TO17F	GpuF0MMReg\:0x62A0	2-186
D1OVL_PWL_180TO1BF	GpuF0MMReg\:0x62A4	2-186
D1OVL_PWL_1C0TO1FF	GpuF0MMReg\:0x62A8	2-186
D1OVL_PWL_200TO23F	GpuF0MMReg\:0x62AC	2-187
D1OVL_PWL_20TO3F	GpuF0MMReg\:0x628C	2-185
D1OVL_PWL_240TO27F	GpuF0MMReg\:0x62B0	2-187
D1OVL_PWL_280TO2BF	GpuF0MMReg\:0x62B4	2-187

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D1OVL_PWL_2C0TO2FF	GpuF0MMReg\:0x62B8	2-187
D1OVL_PWL_300TO33F	GpuF0MMReg\:0x62BC	2-187
D1OVL_PWL_340TO37F	GpuF0MMReg\:0x62C0	2-188
D1OVL_PWL_380TO3BF	GpuF0MMReg\:0x62C4	2-188
D1OVL_PWL_3C0TO3FF	GpuF0MMReg\:0x62C8	2-188
D1OVL_PWL_40TO7F	GpuF0MMReg\:0x6290	2-185
D1OVL_PWL_80TOBF	GpuF0MMReg\:0x6294	2-185
D1OVL_PWL_C0TOFF	GpuF0MMReg\:0x6298	2-186
D1OVL_PWL_TRANSFORM_EN	GpuF0MMReg\:0x6280	2-184
D1OVL_RT_BAND_POSITION	GpuF0MMReg\:0x6508	2-196
D1OVL_RT_PROCEED_COND	GpuF0MMReg\:0x650C	2-196
D1OVL_RT_SKEWCOMMAND	GpuF0MMReg\:0x6500	2-196
D1OVL_RT_SKEWCONTROL	GpuF0MMReg\:0x6504	2-196
D1OVL_RT_STAT	GpuF0MMReg\:0x6510	2-197
D1OVL_START	GpuF0MMReg\:0x61A4	2-179
D1OVL_SURFACE_ADDRESS	GpuF0MMReg\:0x6190	2-178
D1OVL_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x61B0	2-180
D1OVL_SURFACE_OFFSET_X	GpuF0MMReg\:0x619C	2-179
D1OVL_SURFACE_OFFSET_Y	GpuF0MMReg\:0x61A0	2-179
D1OVL_SWAP_CNTL	GpuF0MMReg\:0x618C	2-178
D1OVL_UPDATE	GpuF0MMReg\:0x61AC	2-179
D1VGA_CONTROL	GpuF0MMReg\:0x330	2-163
D2_MVP_AFR_FLIP_FIFO_CNTL	GpuF0MMReg\:0x65EC	2-241
D2_MVP_AFR_FLIP_MODE	GpuF0MMReg\:0x65E8	2-241
D2_MVP_FLIP_LINE_NUM_INSERT	GpuF0MMReg\:0x65F0	2-241
D2COLOR_MATRIX_COEF_1_1	GpuF0MMReg\:0x6B84	2-231
D2COLOR_MATRIX_COEF_1_2	GpuF0MMReg\:0x6B88	2-231
D2COLOR_MATRIX_COEF_1_3	GpuF0MMReg\:0x6B8C	2-231
D2COLOR_MATRIX_COEF_1_4	GpuF0MMReg\:0x6B90	2-231
D2COLOR_MATRIX_COEF_2_1	GpuF0MMReg\:0x6B94	2-232
D2COLOR_MATRIX_COEF_2_2	GpuF0MMReg\:0x6B98	2-232
D2COLOR_MATRIX_COEF_2_3	GpuF0MMReg\:0x6B9C	2-232
D2COLOR_MATRIX_COEF_2_4	GpuF0MMReg\:0x6BA0	2-232
D2COLOR_MATRIX_COEF_3_1	GpuF0MMReg\:0x6BA4	2-232
D2COLOR_MATRIX_COEF_3_2	GpuF0MMReg\:0x6BA8	2-233
D2COLOR_MATRIX_COEF_3_3	GpuF0MMReg\:0x6BAC	2-233
D2COLOR_MATRIX_COEF_3_4	GpuF0MMReg\:0x6BB0	2-233
D2COLOR_SPACE_CONVERT	GpuF0MMReg\:0x693C	2-233
D2CRTC_BLACK_COLOR	GpuF0MMReg\:0x6898	2-281
D2CRTC_BLANK_CONTROL	GpuF0MMReg\:0x6884	2-279
D2CRTC_BLANK_DATA_COLOR	GpuF0MMReg\:0x6890	2-280
D2CRTC_CONTROL	GpuF0MMReg\:0x6880	2-279
D2CRTC_COUNT_CONTROL	GpuF0MMReg\:0x68B4	2-282

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D2CRTC_COUNT_RESET	GpuF0MMReg\:0x68B0	2-282
D2CRTC_DOUBLE_BUFFER_CONTROL	GpuF0MMReg\:0x68EC	2-286
D2CRTC_FLOW_CONTROL	GpuF0MMReg\:0x6874	2-277
D2CRTC_FORCE_COUNT_NOW_CNTL	GpuF0MMReg\:0x6870	2-277
D2CRTC_H_BLANK_START_END	GpuF0MMReg\:0x6804	2-271
D2CRTC_H_SYNC_A	GpuF0MMReg\:0x6808	2-271
D2CRTC_H_SYNC_A_CNTL	GpuF0MMReg\:0x680C	2-272
D2CRTC_H_SYNC_B	GpuF0MMReg\:0x6810	2-272
D2CRTC_H_SYNC_B_CNTL	GpuF0MMReg\:0x6814	2-272
D2CRTC_H_TOTAL	GpuF0MMReg\:0x6800	2-271
D2CRTC_INTERLACE_CONTROL	GpuF0MMReg\:0x6888	2-280
D2CRTC_INTERLACE_STATUS	GpuF0MMReg\:0x688C	2-280
D2CRTC_INTERRUPT_CONTROL	GpuF0MMReg\:0x68DC	2-285
D2CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE	GpuF0MMReg\:0x68B8	2-283
D2CRTC_MVP_INBAND_CNTL_INSERT	GpuF0MMReg\:0x6838	2-205
D2CRTC_MVP_INBAND_CNTL_INSERT_TIMER	GpuF0MMReg\:0x683C	2-205
D2CRTC_MVP_STATUS	GpuF0MMReg\:0x685C	2-240
D2CRTC_OVERSCAN_COLOR	GpuF0MMReg\:0x6894	2-281
D2CRTC_PIXEL_DATA_READBACK	GpuF0MMReg\:0x6878	2-278
D2CRTC_SNAPSHOT_CONTROL	GpuF0MMReg\:0x68CC	2-284
D2CRTC_SNAPSHOT_FRAME	GpuF0MMReg\:0x68D4	2-284
D2CRTC_SNAPSHOT_POSITION	GpuF0MMReg\:0x68D0	2-284
D2CRTC_SNAPSHOT_STATUS	GpuF0MMReg\:0x68C8	2-284
D2CRTC_START_LINE_CONTROL	GpuF0MMReg\:0x68D8	2-285
D2CRTC_STATUS	GpuF0MMReg\:0x689C	2-281
D2CRTC_STATUS_FRAME_COUNT	GpuF0MMReg\:0x68A4	2-282
D2CRTC_STATUS_HV_COUNT	GpuF0MMReg\:0x68AC	2-282
D2CRTC_STATUS_POSITION	GpuF0MMReg\:0x68A0	2-281
D2CRTC_STATUS_VF_COUNT	GpuF0MMReg\:0x68A8	2-282
D2CRTC_STEREO_CONTROL	GpuF0MMReg\:0x68C4	2-283
D2CRTC_STEREO_FORCE_NEXT_EYE	GpuF0MMReg\:0x687C	2-278
D2CRTC_STEREO_STATUS	GpuF0MMReg\:0x68C0	2-283
D2CRTC_TRIGA_CNTL	GpuF0MMReg\:0x6860	2-274
D2CRTC_TRIGA_MANUAL_TRIG	GpuF0MMReg\:0x6864	2-276
D2CRTC_TRIGB_CNTL	GpuF0MMReg\:0x6868	2-276
D2CRTC_TRIGB_MANUAL_TRIG	GpuF0MMReg\:0x686C	2-277
D2CRTC_UPDATE_LOCK	GpuF0MMReg\:0x68E8	2-286
D2CRTC_V_BLANK_START_END	GpuF0MMReg\:0x6824	2-273
D2CRTC_V_SYNC_A	GpuF0MMReg\:0x6828	2-273
D2CRTC_V_SYNC_A_CNTL	GpuF0MMReg\:0x682C	2-274
D2CRTC_V_SYNC_B	GpuF0MMReg\:0x6830	2-274
D2CRTC_V_SYNC_B_CNTL	GpuF0MMReg\:0x6834	2-274
D2CRTC_V_TOTAL	GpuF0MMReg\:0x6820	2-273

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D2CRTC_VBI_END	GpuF0MMReg\:0x6818	2-273
D2CRTC_VERT_SYNC_CONTROL	GpuF0MMReg\:0x68BC	2-283
D2CRTC_VGA_PARAMETER_CAPTURE_MODE	GpuF0MMReg\:0x68F0	2-287
D2CUR_COLOR1	GpuF0MMReg\:0x6C1C	2-237
D2CUR_COLOR2	GpuF0MMReg\:0x6C20	2-237
D2CUR_CONTROL	GpuF0MMReg\:0x6C00	2-236
D2CUR_HOT_SPOT	GpuF0MMReg\:0x6C18	2-237
D2CUR_POSITION	GpuF0MMReg\:0x6C14	2-236
D2CUR_SIZE	GpuF0MMReg\:0x6C10	2-236
D2CUR_SURFACE_ADDRESS	GpuF0MMReg\:0x6C08	2-236
D2CUR_UPDATE	GpuF0MMReg\:0x6C24	2-237
D2GRPH_ALPHA	GpuF0MMReg\:0x6B04	2-227
D2GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6B80	2-230
D2GRPH_CONTROL	GpuF0MMReg\:0x6904	2-208
D2GRPH_ENABLE	GpuF0MMReg\:0x6900	2-208
D2GRPH_FLIP_CONTROL	GpuF0MMReg\:0x6948	2-214
D2GRPH_KEY_RANGE_ALPHA	GpuF0MMReg\:0x6B1C	2-229
D2GRPH_KEY_RANGE_BLUE	GpuF0MMReg\:0x6B18	2-229
D2GRPH_KEY_RANGE_GREEN	GpuF0MMReg\:0x6B14	2-228
D2GRPH_KEY_RANGE_RED	GpuF0MMReg\:0x6B10	2-228
D2GRPH_LUT_SEL	GpuF0MMReg\:0x6908	2-210
D2GRPH_PITCH	GpuF0MMReg\:0x6920	2-211
D2GRPH_PRIMARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6910	2-211
D2GRPH_SECONDARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6918	2-211
D2GRPH_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x694C	2-214
D2GRPH_SURFACE_OFFSET_X	GpuF0MMReg\:0x6924	2-212
D2GRPH_SURFACE_OFFSET_Y	GpuF0MMReg\:0x6928	2-212
D2GRPH_SWAP_CNTL	GpuF0MMReg\:0x690C	2-210
D2GRPH_UPDATE	GpuF0MMReg\:0x6944	2-213
D2GRPH_X_END	GpuF0MMReg\:0x6934	2-213
D2GRPH_X_START	GpuF0MMReg\:0x692C	2-212
D2GRPH_Y_END	GpuF0MMReg\:0x6938	2-213
D2GRPH_Y_START	GpuF0MMReg\:0x6930	2-212
D2ICON_COLOR1	GpuF0MMReg\:0x6C58	2-239
D2ICON_COLOR2	GpuF0MMReg\:0x6C5C	2-239
D2ICON_CONTROL	GpuF0MMReg\:0x6C40	2-238
D2ICON_SIZE	GpuF0MMReg\:0x6C50	2-239
D2ICON_START_POSITION	GpuF0MMReg\:0x6C54	2-239
D2ICON_SURFACE_ADDRESS	GpuF0MMReg\:0x6C48	2-238
D2ICON_UPDATE	GpuF0MMReg\:0x6C60	2-240
D2MODE_MASTER_UPDATE_LOCK	GpuF0MMReg\:0x68E0	2-285
D2MODE_MASTER_UPDATE_MODE	GpuF0MMReg\:0x68E4	2-289
D2OVL_ALPHA	GpuF0MMReg\:0x6B08	2-227

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D2OVL_ALPHA_CONTROL	GpuF0MMReg\:0x6B0C	2-228
D2OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6940	2-222
D2OVL_CONTROL1	GpuF0MMReg\:0x6984	2-215
D2OVL_CONTROL2	GpuF0MMReg\:0x6988	2-216
D2OVL_DFQ_CONTROL	GpuF0MMReg\:0x69B4	2-219
D2OVL_DFQ_STATUS	GpuF0MMReg\:0x69B8	2-219
D2OVL_ENABLE	GpuF0MMReg\:0x6980	2-215
D2OVL_END	GpuF0MMReg\:0x69A8	2-218
D2OVL_KEY_ALPHA	GpuF0MMReg\:0x6B2C	2-230
D2OVL_KEY_CONTROL	GpuF0MMReg\:0x6B00	2-227
D2OVL_KEY_RANGE_BLUE_CB	GpuF0MMReg\:0x6B28	2-230
D2OVL_KEY_RANGE_GREEN_Y	GpuF0MMReg\:0x6B24	2-230
D2OVL_KEY_RANGE_RED_CR	GpuF0MMReg\:0x6B20	2-229
D2OVL_MATRIX_COEF_1_1	GpuF0MMReg\:0x6A04	2-220
D2OVL_MATRIX_COEF_1_2	GpuF0MMReg\:0x6A08	2-220
D2OVL_MATRIX_COEF_1_3	GpuF0MMReg\:0x6A0C	2-220
D2OVL_MATRIX_COEF_1_4	GpuF0MMReg\:0x6A10	2-220
D2OVL_MATRIX_COEF_2_1	GpuF0MMReg\:0x6A14	2-220
D2OVL_MATRIX_COEF_2_2	GpuF0MMReg\:0x6A18	2-221
D2OVL_MATRIX_COEF_2_3	GpuF0MMReg\:0x6A1C	2-221
D2OVL_MATRIX_COEF_2_4	GpuF0MMReg\:0x6A20	2-221
D2OVL_MATRIX_COEF_3_1	GpuF0MMReg\:0x6A24	2-221
D2OVL_MATRIX_COEF_3_2	GpuF0MMReg\:0x6A28	2-222
D2OVL_MATRIX_COEF_3_3	GpuF0MMReg\:0x6A2C	2-222
D2OVL_MATRIX_COEF_3_4	GpuF0MMReg\:0x6A30	2-222
D2OVL_MATRIX_TRANSFORM_EN	GpuF0MMReg\:0x6A00	2-219
D2OVL_PITCH	GpuF0MMReg\:0x6998	2-217
D2OVL_PWL_0TOF	GpuF0MMReg\:0x6A84	2-223
D2OVL_PWL_100TO13F	GpuF0MMReg\:0x6A9C	2-224
D2OVL_PWL_10TO1F	GpuF0MMReg\:0x6A88	2-223
D2OVL_PWL_140TO17F	GpuF0MMReg\:0x6AA0	2-224
D2OVL_PWL_180TO1BF	GpuF0MMReg\:0x6AA4	2-225
D2OVL_PWL_1C0TO1FF	GpuF0MMReg\:0x6AA8	2-225
D2OVL_PWL_200TO23F	GpuF0MMReg\:0x6AAC	2-225
D2OVL_PWL_20TO3F	GpuF0MMReg\:0x6A8C	2-223
D2OVL_PWL_240TO27F	GpuF0MMReg\:0x6AB0	2-225
D2OVL_PWL_280TO2BF	GpuF0MMReg\:0x6AB4	2-225
D2OVL_PWL_2C0TO2FF	GpuF0MMReg\:0x6AB8	2-226
D2OVL_PWL_300TO33F	GpuF0MMReg\:0x6ABC	2-226
D2OVL_PWL_340TO37F	GpuF0MMReg\:0x6AC0	2-226
D2OVL_PWL_380TO3BF	GpuF0MMReg\:0x6AC4	2-226
D2OVL_PWL_3C0TO3FF	GpuF0MMReg\:0x6AC8	2-226
D2OVL_PWL_40TO7F	GpuF0MMReg\:0x6A90	2-223

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
D2OVL_PWL_80TOBF	GpuF0MMReg\:0x6A94	2-224
D2OVL_PWL_C0TOFF	GpuF0MMReg\:0x6A98	2-224
D2OVL_PWL_TRANSFORM_EN	GpuF0MMReg\:0x6A80	2-223
D2OVL_RT_BAND_POSITION	GpuF0MMReg\:0x6D08	2-234
D2OVL_RT_PROCEED_COND	GpuF0MMReg\:0x6D0C	2-235
D2OVL_RT_SKEWCOMMAND	GpuF0MMReg\:0x6D00	2-234
D2OVL_RT_SKEWCONTROL	GpuF0MMReg\:0x6D04	2-234
D2OVL_RT_STAT	GpuF0MMReg\:0x6D10	2-235
D2OVL_START	GpuF0MMReg\:0x69A4	2-218
D2OVL_SURFACE_ADDRESS	GpuF0MMReg\:0x6990	2-217
D2OVL_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x69B0	2-219
D2OVL_SURFACE_OFFSET_X	GpuF0MMReg\:0x699C	2-217
D2OVL_SURFACE_OFFSET_Y	GpuF0MMReg\:0x69A0	2-217
D2OVL_SWAP_CNTL	GpuF0MMReg\:0x698C	2-216
D2OVL_UPDATE	GpuF0MMReg\:0x69AC	2-218
D2VGA_CONTROL	GpuF0MMReg\:0x338	2-164
DAC_DATA	GpuF0MMReg\:0x3C9 VGA_IO\:0x3C9	2-141
DAC_MASK	GpuF0MMReg\:0x3C6 VGA_IO\:0x3C6	2-141
DAC_R_INDEX	GpuF0MMReg\:0x3C7 VGA_IO\:0x3C7	2-141
DAC_W_INDEX	GpuF0MMReg\:0x3C8 VGA_IO\:0x3C8	2-141
DACA_AUTODETECT_CONTROL	GpuF0MMReg\:0x7828	2-290
DACA_AUTODETECT_CONTROL2	GpuF0MMReg\:0x782C	2-290
DACA_AUTODETECT_CONTROL3	GpuF0MMReg\:0x7830	2-290
DACA_AUTODETECT_INT_CONTROL	GpuF0MMReg\:0x7838	2-291
DACA_AUTODETECT_STATUS	GpuF0MMReg\:0x7834	2-291
DACA_COMPARATOR_ENABLE	GpuF0MMReg\:0x785C	2-293
DACA_COMPARATOR_OUTPUT	GpuF0MMReg\:0x7860	2-294
DACA_CONTROL1	GpuF0MMReg\:0x7854	2-292
DACA_CONTROL2	GpuF0MMReg\:0x7858	2-292
DACA_CRC_CONTROL	GpuF0MMReg\:0x780C	2-288
DACA_CRC_EN	GpuF0MMReg\:0x7808	2-288
DACA_CRC_SIG_CONTROL	GpuF0MMReg\:0x781C	2-289
DACA_CRC_SIG_CONTROL_MASK	GpuF0MMReg\:0x7814	2-289
DACA_CRC_SIG_RGB	GpuF0MMReg\:0x7818	2-289
DACA_CRC_SIG_RGB_MASK	GpuF0MMReg\:0x7810	2-289
DACA_DFT_CONFIG	GpuF0MMReg\:0x786C	2-294
DACA_ENABLE	GpuF0MMReg\:0x7800	2-288
DACA_FORCE_DATA	GpuF0MMReg\:0x7840	2-292
DACA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x783C	2-291
DACA_POWERDOWN	GpuF0MMReg\:0x7850	2-292

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
DACA_PWR_CNTL	GpuF0MMReg\:0x7868	2-294
DACA_SOURCE_SELECT	GpuF0MMReg\:0x7804	2-288
DACA_SYNC_SELECT	GpuF0MMReg\:0x7824	2-290
DACA_SYNC_TRISTATE_CONTROL	GpuF0MMReg\:0x7820	2-289
DACA_TEST_ENABLE	GpuF0MMReg\:0x7864	2-294
DACB_AUTODETECT_CONTROL	GpuF0MMReg\:0x7A28	2-296
DACB_AUTODETECT_CONTROL2	GpuF0MMReg\:0x7A2C	2-297
DACB_AUTODETECT_CONTROL3	GpuF0MMReg\:0x7A30	2-297
DACB_AUTODETECT_INT_CONTROL	GpuF0MMReg\:0x7A38	2-298
DACB_AUTODETECT_STATUS	GpuF0MMReg\:0x7A34	2-297
DACB_COMPARATOR_ENABLE	GpuF0MMReg\:0x7A5C	2-299
DACB_COMPARATOR_OUTPUT	GpuF0MMReg\:0x7A60	2-300
DACB_CONTROL1	GpuF0MMReg\:0x7A54	2-299
DACB_CONTROL2	GpuF0MMReg\:0x7A58	2-299
DACB_CRC_CONTROL	GpuF0MMReg\:0x7A0C	2-295
DACB_CRC_EN	GpuF0MMReg\:0x7A08	2-295
DACB_CRC_SIG_CONTROL	GpuF0MMReg\:0x7A1C	2-296
DACB_CRC_SIG_CONTROL_MASK	GpuF0MMReg\:0x7A14	2-296
DACB_CRC_SIG_RGB	GpuF0MMReg\:0x7A18	2-296
DACB_CRC_SIG_RGB_MASK	GpuF0MMReg\:0x7A10	2-295
DACB_ENABLE	GpuF0MMReg\:0x7A00	2-294
DACB_FORCE_DATA	GpuF0MMReg\:0x7A40	2-298
DACB_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x7A3C	2-298
DACB_POWERDOWN	GpuF0MMReg\:0x7A50	2-298
DACB_PWR_CNTL	GpuF0MMReg\:0x7A68	2-301
DACB_SOURCE_SELECT	GpuF0MMReg\:0x7A04	2-295
DACB_SYNC_SELECT	GpuF0MMReg\:0x7A24	2-296
DACB_SYNC_TRISTATE_CONTROL	GpuF0MMReg\:0x7A20	2-296
DACB_TEST_ENABLE	GpuF0MMReg\:0x7A64	2-300
DC_CRTC_MASTER_EN	GpuF0MMReg\:0x60F8	2-255
DC_CRTC_TV_CONTROL	GpuF0MMReg\:0x60FC	2-255
DC_GENERICA	GpuF0MMReg\:0x7DC0	2-342
DC_GENERICB	GpuF0MMReg\:0x7DC4	2-342
DC_GPIO_DDC1_A	GpuF0MMReg\:0x7E44	2-352
DC_GPIO_DDC1_EN	GpuF0MMReg\:0x7E48	2-353
DC_GPIO_DDC1_MASK	GpuF0MMReg\:0x7E40	2-352
DC_GPIO_DDC1_Y	GpuF0MMReg\:0x7E4C	2-353
DC_GPIO_DDC2_A	GpuF0MMReg\:0x7E54	2-353
DC_GPIO_DDC2_EN	GpuF0MMReg\:0x7E58	2-354
DC_GPIO_DDC2_MASK	GpuF0MMReg\:0x7E50	2-353
DC_GPIO_DDC2_Y	GpuF0MMReg\:0x7E5C	2-354
DC_GPIO_DDC3_A	GpuF0MMReg\:0x7E64	2-354
DC_GPIO_DDC3_EN	GpuF0MMReg\:0x7E68	2-355

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
DC_GPIO_DDC3_MASK	GpuF0MMReg\:0x7E60	2-354
DC_GPIO_DDC3_Y	GpuF0MMReg\:0x7E6C	2-355
DC_GPIO_DDC4_A	GpuF0MMReg\:0x7E04	2-350
DC_GPIO_DDC4_EN	GpuF0MMReg\:0x7E08	2-350
DC_GPIO_DDC4_MASK	GpuF0MMReg\:0x7E00	2-350
DC_GPIO_DDC4_Y	GpuF0MMReg\:0x7E0C	2-350
DC_GPIO_DEBUG	GpuF0MMReg\:0x7E2C	2-350
DC_GPIO_DVODATA_A	GpuF0MMReg\:0x7E34	2-351
DC_GPIO_DVODATA_EN	GpuF0MMReg\:0x7E38	2-351
DC_GPIO_DVODATA_MASK	GpuF0MMReg\:0x7E30	2-351
DC_GPIO_DVODATA_Y	GpuF0MMReg\:0x7E3C	2-352
DC_GPIO_GENERIC_A	GpuF0MMReg\:0x7DE4	2-349
DC_GPIO_GENERIC_EN	GpuF0MMReg\:0x7DE8	2-349
DC_GPIO_GENERIC_MASK	GpuF0MMReg\:0x7DE0	2-349
DC_GPIO_GENERIC_Y	GpuF0MMReg\:0x7DEC	2-349
DC_GPIO_HPD_A	GpuF0MMReg\:0x7E94	2-357
DC_GPIO_HPD_EN	GpuF0MMReg\:0x7E98	2-358
DC_GPIO_HPD_MASK	GpuF0MMReg\:0x7E90	2-357
DC_GPIO_HPD_Y	GpuF0MMReg\:0x7E9C	2-358
DC_GPIO_PAD_STRENGTH_1	GpuF0MMReg\:0x7ED4	2-359
DC_GPIO_PAD_STRENGTH_2	GpuF0MMReg\:0x7ED8	2-360
DC_GPIO_PWRSEQ_A	GpuF0MMReg\:0x7EA4	2-359
DC_GPIO_PWRSEQ_EN	GpuF0MMReg\:0x7EA8	2-359
DC_GPIO_PWRSEQ_MASK	GpuF0MMReg\:0x7EA0	2-358
DC_GPIO_PWRSEQ_Y	GpuF0MMReg\:0x7EAC	2-359
DC_GPIO_SYNCA_A	GpuF0MMReg\:0x7E74	2-355
DC_GPIO_SYNCA_EN	GpuF0MMReg\:0x7E78	2-356
DC_GPIO_SYNCA_MASK	GpuF0MMReg\:0x7E70	2-355
DC_GPIO_SYNCA_Y	GpuF0MMReg\:0x7E7C	2-356
DC_GPIO_SYNCB_A	GpuF0MMReg\:0x7E84	2-356
DC_GPIO_SYNCB_EN	GpuF0MMReg\:0x7E88	2-357
DC_GPIO_SYNCB_MASK	GpuF0MMReg\:0x7E80	2-356
DC_GPIO_SYNCB_Y	GpuF0MMReg\:0x7E8C	2-357
DC_HOT_PLUG_DETECT_CLOCK_CONTROL	GpuF0MMReg\:0x7D20	2-340
DC_HOT_PLUG_DETECT1_CONTROL	GpuF0MMReg\:0x7D00	2-339
DC_HOT_PLUG_DETECT1_INT_CONTROL	GpuF0MMReg\:0x7D08	2-339
DC_HOT_PLUG_DETECT1_INT_STATUS	GpuF0MMReg\:0x7D04	2-339
DC_HOT_PLUG_DETECT2_CONTROL	GpuF0MMReg\:0x7D10	2-339
DC_HOT_PLUG_DETECT2_INT_CONTROL	GpuF0MMReg\:0x7D18	2-340
DC_HOT_PLUG_DETECT2_INT_STATUS	GpuF0MMReg\:0x7D14	2-340
DC_HOT_PLUG_DETECT3_CONTROL	GpuF0MMReg\:0x7D24	2-340
DC_HOT_PLUG_DETECT3_INT_CONTROL	GpuF0MMReg\:0x7D2C	2-341
DC_HOT_PLUG_DETECT3_INT_STATUS	GpuF0MMReg\:0x7D28	2-341

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
DC_LUT_30_COLOR	GpuF0MMReg\:0x6494	2-243
DC_LUT_AUTOFILL	GpuF0MMReg\:0x64A0	2-243
DC_LUT_PWL_DATA	GpuF0MMReg\:0x6490	2-242
DC_LUT_READ_PIPE_SELECT	GpuF0MMReg\:0x6498	2-243
DC_LUT_RW_INDEX	GpuF0MMReg\:0x6488	2-242
DC_LUT_RW_MODE	GpuF0MMReg\:0x6484	2-242
DC_LUT_RW_SELECT	GpuF0MMReg\:0x6480	2-241
DC_LUT_SEQ_COLOR	GpuF0MMReg\:0x648C	2-242
DC_LUT_WRITE_EN_MASK	GpuF0MMReg\:0x649C	2-243
DC_LUTA_BLACK_OFFSET_BLUE	GpuF0MMReg\:0x64C4	2-246
DC_LUTA_BLACK_OFFSET_GREEN	GpuF0MMReg\:0x64C8	2-246
DC_LUTA_BLACK_OFFSET_RED	GpuF0MMReg\:0x64CC	2-246
DC_LUTA_CONTROL	GpuF0MMReg\:0x64C0	2-244
DC_LUTA_WHITE_OFFSET_BLUE	GpuF0MMReg\:0x64D0	2-246
DC_LUTA_WHITE_OFFSET_GREEN	GpuF0MMReg\:0x64D4	2-246
DC_LUTA_WHITE_OFFSET_RED	GpuF0MMReg\:0x64D8	2-246
DC_LUTB_BLACK_OFFSET_BLUE	GpuF0MMReg\:0x6CC4	2-249
DC_LUTB_BLACK_OFFSET_GREEN	GpuF0MMReg\:0x6CC8	2-249
DC_LUTB_BLACK_OFFSET_RED	GpuF0MMReg\:0x6CCC	2-249
DC_LUTB_CONTROL	GpuF0MMReg\:0x6CC0	2-247
DC_LUTB_WHITE_OFFSET_BLUE	GpuF0MMReg\:0x6CD0	2-249
DC_LUTB_WHITE_OFFSET_GREEN	GpuF0MMReg\:0x6CD4	2-249
DC_LUTB_WHITE_OFFSET_RED	GpuF0MMReg\:0x6CD8	2-249
DC_MVP_LB_CONTROL	GpuF0MMReg\:0x65F4	2-254
DC_PAD_EXTERN_SIG	GpuF0MMReg\:0x7DCC	2-342
DC_REF_CLK_CNTL	GpuF0MMReg\:0x7DD4	2-343
DCP_CRC_CONTROL	GpuF0MMReg\:0x6C80	2-250
DCP_CRC_MASK	GpuF0MMReg\:0x6C84	2-250
DCP_CRC_P0_CURRENT	GpuF0MMReg\:0x6C88	2-250
DCP_CRC_P0_LAST	GpuF0MMReg\:0x6C90	2-250
DCP_CRC_P1_CURRENT	GpuF0MMReg\:0x6C8C	2-250
DCP_CRC_P1_LAST	GpuF0MMReg\:0x6C94	2-251
DCP_LB_DATA_GAP_BETWEEN_CHUNK	GpuF0MMReg\:0x6CBC	2-254
DCP_MULTI_CHIP_CNTL	GpuF0MMReg\:0x6CA4	2-252
DCP_TILING_CONFIG	GpuF0MMReg\:0x6CA0	2-251
DISP_INTERRUPT_STATUS	GpuF0MMReg\:0x7EDC	2-343
DISP_INTERRUPT_STATUS_CONTINUE	GpuF0MMReg\:0x7EE8	2-345
DISP_TIMER_CONTROL	GpuF0MMReg\:0x7EF0	2-347
DMIF_CONTROL	GpuF0MMReg\:0x6CB0	2-252
DMIF_STATUS	GpuF0MMReg\:0x6CB4	2-253
DOUT_POWER_MANAGEMENT_CNTL	GpuF0MMReg\:0x7EE0	2-346
DVOA_BIT_DEPTH_CONTROL	GpuF0MMReg\:0x7988	2-315

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
DVOA_CONTROL	GpuF0MMReg\:0x7990	2-316
DVOA_CRC_CONTROL	GpuF0MMReg\:0x7998	2-317
DVOA_CRC_EN	GpuF0MMReg\:0x7994	2-316
DVOA_CRC_SIG_MASK1	GpuF0MMReg\:0x799C	2-317
DVOA_CRC_SIG_MASK2	GpuF0MMReg\:0x79A0	2-317
DVOA_CRC_SIG_RESULT1	GpuF0MMReg\:0x79A4	2-317
DVOA_CRC_SIG_RESULT2	GpuF0MMReg\:0x79A8	2-318
DVOA_CRC2_SIG_MASK	GpuF0MMReg\:0x79AC	2-318
DVOA_CRC2_SIG_RESULT	GpuF0MMReg\:0x79B0	2-318
DVOA_ENABLE	GpuF0MMReg\:0x7980	2-314
DVOA_FORCE_DATA	GpuF0MMReg\:0x79BC	2-319
DVOA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x79B8	2-319
DVOA_OUTPUT	GpuF0MMReg\:0x798C	2-316
DVOA_SOURCE_SELECT	GpuF0MMReg\:0x7984	2-315
DVOA_STRENGTH_CONTROL	GpuF0MMReg\:0x79B4	2-318
GENFC_RD	GpuF0MMReg\:0x3CA	2-140
GENFC_WT	GpuF0MMReg\:0x3BA GpuF0MMReg\:0x3DA VGA_IO\:0x3BA VGA_IO\:0x3DA	2-140
GENS0	GpuF0MMReg\:0x3C2 VGA_IO\:0x3C2	2-140
GENS1	GpuF0MMReg\:0x3BA GpuF0MMReg\:0x3DA VGA_IO\:0x3BA VGA_IO\:0x3DA	2-140
GRPH8_DATA	GpuF0MMReg\:0x3CF VGA_IO\:0x3CF	2-151
GRPH8_IDX	GpuF0MMReg\:0x3CE VGA_IO\:0x3CE	2-151
LVTMA_2ND_CRC_RESULT	GpuF0MMReg\:0x7ABC	2-324
LVTMA_BIT_DEPTH_CONTROL	GpuF0MMReg\:0x7A94	2-321
LVTMA_BL_MOD_CNTL	GpuF0MMReg\:0x7AFC	2-332
LVTMA_CNTL	GpuF0MMReg\:0x7A80	2-319
LVTMA_COLOR_FORMAT	GpuF0MMReg\:0x7A88	2-320
LVTMA_CONTROL_CHAR	GpuF0MMReg\:0x7A98	2-322
LVTMA_CONTROL0_FEEDBACK	GpuF0MMReg\:0x7A9C	2-322
LVTMA_CRC_CNTL	GpuF0MMReg\:0x7AB0	2-323
LVTMA_CRC_SIG_MASK	GpuF0MMReg\:0x7AB4	2-324
LVTMA_CRC_SIG_RGB	GpuF0MMReg\:0x7AB8	2-324
LVTMA_CTL_BITS	GpuF0MMReg\:0x7ACC	2-326
LVTMA_CTL0_1_GEN_CNTL	GpuF0MMReg\:0x7AE0	2-327
LVTMA_CTL2_3_GEN_CNTL	GpuF0MMReg\:0x7AE4	2-328
LVTMA_DATA_SYNCHRONIZATION	GpuF0MMReg\:0x7ADC	2-326
LVTMA_DCBALANCER_CONTROL	GpuF0MMReg\:0x7AD0	2-326

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
LVTMA_DEBUG	*GpuF0MMReg\:0x7AC8*	*2-325*
LVTMA_DITHER_RAND_SEED	*GpuF0MMReg\:0x7B20*	*2-336*
LVTMA_FORCE_DATA	*GpuF0MMReg\:0x7A90*	*2-321*
LVTMA_FORCE_OUTPUT_CNTL	*GpuF0MMReg\:0x7A8C*	*2-321*
LVTMA_LOAD_DETECT	*GpuF0MMReg\:0x7B0C*	*2-333*
LVTMA_LVTM_DATA_CNTL	*GpuF0MMReg\:0x7B00*	*2-332*
LVTMA_MACRO_CONTROL	*GpuF0MMReg\:0x7B10*	*2-334*
LVTMA_MODE	*GpuF0MMReg\:0x7B04*	*2-332*
LVTMA_PREEMPHASIS_CONTROL	*GpuF0MMReg\:0x7B28*	*2-336*
LVTMA_PWRSEQ_CNTL	*GpuF0MMReg\:0x7AF4*	*2-330*
LVTMA_PWRSEQ_DELAY1	*GpuF0MMReg\:0x7AEC*	*2-329*
LVTMA_PWRSEQ_DELAY2	*GpuF0MMReg\:0x7AF0*	*2-330*
LVTMA_PWRSEQ_REF_DIV	*GpuF0MMReg\:0x7AE8*	*2-329*
LVTMA_PWRSEQ_STATE	*GpuF0MMReg\:0x7AF8*	*2-331*
LVTMA_RANDOM_PATTERN_SEED	*GpuF0MMReg\:0x7AC4*	*2-325*
LVTMA_RED_BLUE_SWITCH	*GpuF0MMReg\:0x7AD4*	*2-326*
LVTMA_REG_TEST_OUTPUT	*GpuF0MMReg\:0x7B18*	*2-335*
LVTMA_SOURCE_SELECT	*GpuF0MMReg\:0x7A84*	*2-320*
LVTMA_SPLIT_LOAD_DETECT	*GpuF0MMReg\:0x7930*	*2-337*
LVTMA_SPLIT_PLL_ADJUST	*GpuF0MMReg\:0x7934*	*2-337*
LVTMA_SPLIT_TRANSMITTER_ADJUST	*GpuF0MMReg\:0x793C*	*2-338*
LVTMA_SPLIT_TRANSMITTER_CONTROL	*GpuF0MMReg\:0x7938*	*2-337*
LVTMA_SPLIT_TRANSMITTER_ENABLE	*GpuF0MMReg\:0x792C*	*2-336*
LVTMA_STEREOSYNC_CTL_SEL	*GpuF0MMReg\:0x7AA0*	*2-322*
LVTMA_SYNC_CHAR_PATTERN_0_1	*GpuF0MMReg\:0x7AA8*	*2-323*
LVTMA_SYNC_CHAR_PATTERN_2_3	*GpuF0MMReg\:0x7AAC*	*2-323*
LVTMA_SYNC_CHAR_PATTERN_SEL	*GpuF0MMReg\:0x7AA4*	*2-323*
LVTMA_TEST_PATTERN	*GpuF0MMReg\:0x7AC0*	*2-324*
LVTMA_TRANSMITTER_ADJUST	*GpuF0MMReg\:0x7B24*	*2-336*
LVTMA_TRANSMITTER_CONTROL	*GpuF0MMReg\:0x7B14*	*2-334*
LVTMA_TRANSMITTER_DEBUG	*GpuF0MMReg\:0x7B1C*	*2-335*
LVTMA_TRANSMITTER_ENABLE	*GpuF0MMReg\:0x7B08*	*2-333*
MCIF_CONTROL	*GpuF0MMReg\:0x6CB8*	*2-253*
SEQ8_DATA	*GpuF0MMReg\:0x3C5* *VGA_IO\:0x3C5*	*2-142*
SEQ8_IDX	*GpuF0MMReg\:0x3C4* *VGA_IO\:0x3C4*	*2-142*
TMDSA_2ND_CRC_RESULT	*GpuF0MMReg\:0x78BC*	*2-306*
TMDSA_BIT_DEPTH_CONTROL	*GpuF0MMReg\:0x7894*	*2-303*
TMDSA_CNTL	*GpuF0MMReg\:0x7880*	*2-301*
TMDSA_COLOR_FORMAT	*GpuF0MMReg\:0x7888*	*2-302*
TMDSA_CONTROL_CHAR	*GpuF0MMReg\:0x7898*	*2-303*
TMDSA_CONTROL0_FEEDBACK	*GpuF0MMReg\:0x789C*	*2-304*

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
TMDSA_CRC_CNTL	GpuF0MMReg\:0x78B0	2-305
TMDSA_CRC_SIG_MASK	GpuF0MMReg\:0x78B4	2-305
TMDSA_CRC_SIG_RGB	GpuF0MMReg\:0x78B8	2-305
TMDSA_CTL_BITS	GpuF0MMReg\:0x78CC	2-307
TMDSA_CTL0_1_GEN_CNTL	GpuF0MMReg\:0x78E0	2-308
TMDSA_CTL2_3_GEN_CNTL	GpuF0MMReg\:0x78E4	2-309
TMDSA_DATA_SYNCHRONIZATION	GpuF0MMReg\:0x78DC	2-308
TMDSA_DCBALANCER_CONTROL	GpuF0MMReg\:0x78D0	2-307
TMDSA_DEBUG	GpuF0MMReg\:0x78C8	2-307
TMDSA_DITHER_RAND_SEED	GpuF0MMReg\:0x791C	2-313
TMDSA_FORCE_DATA	GpuF0MMReg\:0x7890	2-302
TMDSA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x788C	2-302
TMDSA_LOAD_DETECT	GpuF0MMReg\:0x7908	2-311
TMDSA_PLL_ADJUST	GpuF0MMReg\:0x790C	2-311
TMDSA_RANDOM_PATTERN_SEED	GpuF0MMReg\:0x78C4	2-307
TMDSA_RED_BLUE_SWITCH	GpuF0MMReg\:0x78D4	2-307
TMDSA_REG_TEST_OUTPUTA	GpuF0MMReg\:0x7914	2-313
TMDSA_REG_TEST_OUTPUTB	GpuF0MMReg\:0x7924	2-314
TMDSA_SOURCE_SELECT	GpuF0MMReg\:0x7884	2-302
TMDSA_STEREOSYNC_CTL_SEL	GpuF0MMReg\:0x78A0	2-304
TMDSA_SYNC_CHAR_PATTERN_0_1	GpuF0MMReg\:0x78A8	2-304
TMDSA_SYNC_CHAR_PATTERN_2_3	GpuF0MMReg\:0x78AC	2-304
TMDSA_SYNC_CHAR_PATTERN_SEL	GpuF0MMReg\:0x78A4	2-304
TMDSA_TEST_PATTERN	GpuF0MMReg\:0x78C0	2-306
TMDSA_TRANSMITTER_ADJUST	GpuF0MMReg\:0x7920	2-314
TMDSA_TRANSMITTER_CONTROL	GpuF0MMReg\:0x7910	2-312
TMDSA_TRANSMITTER_DEBUG	GpuF0MMReg\:0x7918	2-313
TMDSA_TRANSMITTER_ENABLE	GpuF0MMReg\:0x7904	2-310
VGA_CACHE_CONTROL	GpuF0MMReg\:0x32C	2-163
VGA_DEBUG_READBACK_DATA	GpuF0MMReg\:0x35C	2-169
VGA_DEBUG_READBACK_INDEX	GpuF0MMReg\:0x358	2-168
VGA_DISPBUF1_SURFACE_ADDR	GpuF0MMReg\:0x318	2-162
VGA_DISPBUF2_SURFACE_ADDR	GpuF0MMReg\:0x320	2-162
VGA_HDP_CONTROL	GpuF0MMReg\:0x328	2-162
VGA_HW_DEBUG	GpuF0MMReg\:0x33C	2-165
VGA_INTERRUPT_CONTROL	GpuF0MMReg\:0x344	2-165
VGA_INTERRUPT_STATUS	GpuF0MMReg\:0x34C	2-166
VGA_MAIN_CONTROL	GpuF0MMReg\:0x350	2-166
VGA_MEM_READ_PAGE_ADDR	GpuF0MMReg\:0x4C GpuIOReg\:0x4C	2-169
VGA_MEM_WRITE_PAGE_ADDR	GpuF0MMReg\:0x48 GpuIOReg\:0x48	2-169
VGA_MEMORY_BASE_ADDRESS	GpuF0MMReg\:0x310	2-162
VGA_MODE_CONTROL	GpuF0MMReg\:0x308	2-161

Table A-5 Display Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
VGA_RENDER_CONTROL	*GpuF0MMReg\:0x300*	*2-159*
VGA_SEQUENCER_RESET_CONTROL	*GpuF0MMReg\:0x304*	*2-160*
VGA_STATUS	*GpuF0MMReg\:0x340*	*2-165*
VGA_STATUS_CLEAR	*GpuF0MMReg\:0x348*	*2-165*
VGA_SURFACE_PITCH_SELECT	*GpuF0MMReg\:0x30C*	*2-161*
VGA_TEST_CONTROL	*GpuF0MMReg\:0x354*	*2-168*

A.7 Display Controller Registers Stored by Address

Table A-6 Display Controller Registers Sorted by Address

Register Name	Address	Page
VGA_RENDER_CONTROL	*GpuF0MMReg\:0x300*	*2-159*
VGA_SEQUENCER_RESET_CONTROL	*GpuF0MMReg\:0x304*	*2-160*
VGA_MODE_CONTROL	*GpuF0MMReg\:0x308*	*2-161*
VGA_SURFACE_PITCH_SELECT	*GpuF0MMReg\:0x30C*	*2-161*
VGA_MEMORY_BASE_ADDRESS	*GpuF0MMReg\:0x310*	*2-162*
VGA_DISPBUF1_SURFACE_ADDR	*GpuF0MMReg\:0x318*	*2-162*
VGA_DISPBUF2_SURFACE_ADDR	*GpuF0MMReg\:0x320*	*2-162*
VGA_HDP_CONTROL	*GpuF0MMReg\:0x328*	*2-162*
VGA_CACHE_CONTROL	*GpuF0MMReg\:0x32C*	*2-163*
D1VGA_CONTROL	*GpuF0MMReg\:0x330*	*2-163*
D2VGA_CONTROL	*GpuF0MMReg\:0x338*	*2-164*
VGA_HW_DEBUG	*GpuF0MMReg\:0x33C*	*2-165*
VGA_STATUS	*GpuF0MMReg\:0x340*	*2-165*
VGA_INTERRUPT_CONTROL	*GpuF0MMReg\:0x344*	*2-165*
VGA_STATUS_CLEAR	*GpuF0MMReg\:0x348*	*2-165*
VGA_INTERRUPT_STATUS	*GpuF0MMReg\:0x34C*	*2-166*
VGA_MAIN_CONTROL	*GpuF0MMReg\:0x350*	*2-166*
VGA_TEST_CONTROL	*GpuF0MMReg\:0x354*	*2-168*
VGA_DEBUG_READBACK_INDEX	*GpuF0MMReg\:0x358*	*2-168*
VGA_DEBUG_READBACK_DATA	*GpuF0MMReg\:0x35C*	*2-169*
CRTC8_IDX	*GpuF0MMReg\:0x3B4* *GpuF0MMReg\:0x3D4* *VGA_IO\:0x3B4* *VGA_IO\:0x3D4*	*2-144*
CRTC8_DATA	*GpuF0MMReg\:0x3B5* *GpuF0MMReg\:0x3D5* *VGA_IO\:0x3B5* *VGA_IO\:0x3D5*	*2-144*
GENFC_WT	*GpuF0MMReg\:0x3BA* *GpuF0MMReg\:0x3DA* *VGA_IO\:0x3BA* *VGA_IO\:0x3DA*	*2-140*
GENS1	*GpuF0MMReg\:0x3BA* *GpuF0MMReg\:0x3DA* *VGA_IO\:0x3BA* *VGA_IO\:0x3DA*	*2-140*
ATTRDW	*GpuF0MMReg\:0x3C0* *VGA_IO\:0x3C0*	*2-154*
ATTRX	*GpuF0MMReg\:0x3C0* *VGA_IO\:0x3C0*	*2-153*
ATTRDR	*GpuF0MMReg\:0x3C1* *VGA_IO\:0x3C1*	*2-154*
GENS0	*GpuF0MMReg\:0x3C2* *VGA_IO\:0x3C2*	*2-140*

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
SEQ8_IDX	GpuF0MMReg\:0x3C4 VGA_IO\:0x3C4	2-142
SEQ8_DATA	GpuF0MMReg\:0x3C5 VGA_IO\:0x3C5	2-142
DAC_MASK	GpuF0MMReg\:0x3C6 VGA_IO\:0x3C6	2-141
DAC_R_INDEX	GpuF0MMReg\:0x3C7 VGA_IO\:0x3C7	2-141
DAC_W_INDEX	GpuF0MMReg\:0x3C8 VGA_IO\:0x3C8	2-141
DAC_DATA	GpuF0MMReg\:0x3C9 VGA_IO\:0x3C9	2-141
GENFC_RD	GpuF0MMReg\:0x3CA	2-140
GRPH8_IDX	GpuF0MMReg\:0x3CE VGA_IO\:0x3CE	2-151
GRPH8_DATA	GpuF0MMReg\:0x3CF VGA_IO\:0x3CF	2-151
VGA_MEM_WRITE_PAGE_ADDR	GpuF0MMReg\:0x48 GpuIOReg\:0x48	2-169
VGA_MEM_READ_PAGE_ADDR	GpuF0MMReg\:0x4C GpuIOReg\:0x4C	2-169
D1CRTC_H_TOTAL	GpuF0MMReg\:0x6000	2-255
D1CRTC_H_BLANK_START_END	GpuF0MMReg\:0x6004	2-255
D1CRTC_H_SYNC_A	GpuF0MMReg\:0x6008	2-256
D1CRTC_H_SYNC_A_CNTL	GpuF0MMReg\:0x600C	2-256
D1CRTC_H_SYNC_B	GpuF0MMReg\:0x6010	2-256
D1CRTC_H_SYNC_B_CNTL	GpuF0MMReg\:0x6014	2-257
D1CRTC_V_TOTAL	GpuF0MMReg\:0x6020	2-257
D1CRTC_V_BLANK_START_END	GpuF0MMReg\:0x6024	2-257
D1CRTC_V_SYNC_A	GpuF0MMReg\:0x6028	2-257
D1CRTC_V_SYNC_A_CNTL	GpuF0MMReg\:0x602C	2-258
D1CRTC_V_SYNC_B	GpuF0MMReg\:0x6030	2-258
D1CRTC_V_SYNC_B_CNTL	GpuF0MMReg\:0x6034	2-258
D1CRTC_MVP_CONTROL1	GpuF0MMReg\:0x6038	2-202
D1CRTC_MVP_CONTROL2	GpuF0MMReg\:0x603C	2-203
D1CRTC_MVP_FIFO_CONTROL	GpuF0MMReg\:0x6040	2-203
D1CRTC_MVP_FIFO_STATUS	GpuF0MMReg\:0x6044	2-203
D1CRTC_MVP_SLAVE_STATUS	GpuF0MMReg\:0x6048	2-204
D1CRTC_MVP_INBAND_CNTL_CAP	GpuF0MMReg\:0x604C	2-204
D1CRTC_MVP_INBAND_CNTL_INSERT	GpuF0MMReg\:0x6050	2-204
D1CRTC_MVP_INBAND_CNTL_INSERT_TIMER	GpuF0MMReg\:0x6054	2-204
D1CRTC_MVP_BLACK_KEYER	GpuF0MMReg\:0x6058	2-205
D1CRTC_MVP_STATUS	GpuF0MMReg\:0x605C	2-205
D1CRTC_TRIGA_CNTL	GpuF0MMReg\:0x6060	2-258
D1CRTC_TRIGA_MANUAL_TRIG	GpuF0MMReg\:0x6064	2-260

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D1CRTC_TRIGB_CNTL	GpuF0MMReg\:0x6068	2-260
D1CRTC_TRIGB_MANUAL_TRIG	GpuF0MMReg\:0x606C	2-261
D1CRTC_FORCE_COUNT_NOW_CNTL	GpuF0MMReg\:0x6070	2-261
D1CRTC_FLOW_CONTROL	GpuF0MMReg\:0x6074	2-261
D1CRTC_PIXEL_DATA_READBACK	GpuF0MMReg\:0x6078	2-262
D1CRTC_STEREO_FORCE_NEXT_EYE	GpuF0MMReg\:0x607C	2-262
D1CRTC_CONTROL	GpuF0MMReg\:0x6080	2-263
D1CRTC_BLANK_CONTROL	GpuF0MMReg\:0x6084	2-263
D1CRTC_INTERLACE_CONTROL	GpuF0MMReg\:0x6088	2-264
D1CRTC_INTERLACE_STATUS	GpuF0MMReg\:0x608C	2-264
D1CRTC_BLANK_DATA_COLOR	GpuF0MMReg\:0x6090	2-264
D1CRTC_OVERSCAN_COLOR	GpuF0MMReg\:0x6094	2-265
D1CRTC_BLACK_COLOR	GpuF0MMReg\:0x6098	2-265
D1CRTC_STATUS	GpuF0MMReg\:0x609C	2-265
D1CRTC_STATUS_POSITION	GpuF0MMReg\:0x60A0	2-265
D1CRTC_STATUS_FRAME_COUNT	GpuF0MMReg\:0x60A4	2-266
D1CRTC_STATUS_VF_COUNT	GpuF0MMReg\:0x60A8	2-266
D1CRTC_STATUS_HV_COUNT	GpuF0MMReg\:0x60AC	2-266
D1CRTC_COUNT_RESET	GpuF0MMReg\:0x60B0	2-266
D1CRTC_COUNT_CONTROL	GpuF0MMReg\:0x60B4	2-266
D1CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE	GpuF0MMReg\:0x60B8	2-267
D1CRTC_VERT_SYNC_CONTROL	GpuF0MMReg\:0x60BC	2-267
D1CRTC_STEREO_STATUS	GpuF0MMReg\:0x60C0	2-267
D1CRTC_STEREO_CONTROL	GpuF0MMReg\:0x60C4	2-267
D1CRTC_SNAPSHOT_STATUS	GpuF0MMReg\:0x60C8	2-268
D1CRTC_SNAPSHOT_CONTROL	GpuF0MMReg\:0x60CC	2-268
D1CRTC_SNAPSHOT_POSITION	GpuF0MMReg\:0x60D0	2-268
D1CRTC_SNAPSHOT_FRAME	GpuF0MMReg\:0x60D4	2-268
D1CRTC_START_LINE_CONTROL	GpuF0MMReg\:0x60D8	2-269
D1CRTC_INTERRUPT_CONTROL	GpuF0MMReg\:0x60DC	2-267
D1MODE_MASTER_UPDATE_LOCK	GpuF0MMReg\:0x60E0	2-269
D1MODE_MASTER_UPDATE_MODE	GpuF0MMReg\:0x60E4	2-270
D1CRTC_UPDATE_LOCK	GpuF0MMReg\:0x60E8	2-270
D1CRTC_DOUBLE_BUFFER_CONTROL	GpuF0MMReg\:0x60EC	2-270
D1CRTC_VGA_PARAMETER_CAPTURE_MODE	GpuF0MMReg\:0x60F0	2-271
DC_CRTC_MASTER_EN	GpuF0MMReg\:0x60F8	2-255
DC_CRTC_TV_CONTROL	GpuF0MMReg\:0x60FC	2-255
D1GRPH_ENABLE	GpuF0MMReg\:0x6100	2-170
D1GRPH_CONTROL	GpuF0MMReg\:0x6104	2-170
D1GRPH_LUT_SEL	GpuF0MMReg\:0x6108	2-171
D1GRPH_SWAP_CNTL	GpuF0MMReg\:0x610C	2-172
D1GRPH_PRIMARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6110	2-172
D1GRPH_SECONDARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6118	2-173

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D1GRPH_PITCH	GpuF0MMReg\:0x6120	2-173
D1GRPH_SURFACE_OFFSET_X	GpuF0MMReg\:0x6124	2-173
D1GRPH_SURFACE_OFFSET_Y	GpuF0MMReg\:0x6128	2-173
D1GRPH_X_START	GpuF0MMReg\:0x612C	2-173
D1GRPH_Y_START	GpuF0MMReg\:0x6130	2-174
D1GRPH_X_END	GpuF0MMReg\:0x6134	2-174
D1GRPH_Y_END	GpuF0MMReg\:0x6138	2-174
D1COLOR_SPACE_CONVERT	GpuF0MMReg\:0x613C	2-195
D1OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6140	2-181
D1GRPH_UPDATE	GpuF0MMReg\:0x6144	2-174
D1GRPH_FLIP_CONTROL	GpuF0MMReg\:0x6148	2-176
D1GRPH_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x614C	2-176
D1OVL_ENABLE	GpuF0MMReg\:0x6180	2-176
D1OVL_CONTROL1	GpuF0MMReg\:0x6184	2-176
D1OVL_CONTROL2	GpuF0MMReg\:0x6188	2-177
D1OVL_SWAP_CNTL	GpuF0MMReg\:0x618C	2-178
D1OVL_SURFACE_ADDRESS	GpuF0MMReg\:0x6190	2-178
D1OVL_PITCH	GpuF0MMReg\:0x6198	2-178
D1OVL_SURFACE_OFFSET_X	GpuF0MMReg\:0x619C	2-179
D1OVL_SURFACE_OFFSET_Y	GpuF0MMReg\:0x61A0	2-179
D1OVL_START	GpuF0MMReg\:0x61A4	2-179
D1OVL_END	GpuF0MMReg\:0x61A8	2-179
D1OVL_UPDATE	GpuF0MMReg\:0x61AC	2-179
D1OVL_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x61B0	2-180
D1OVL_DFQ_CONTROL	GpuF0MMReg\:0x61B4	2-180
D1OVL_DFQ_STATUS	GpuF0MMReg\:0x61B8	2-181
D1OVL_MATRIX_TRANSFORM_EN	GpuF0MMReg\:0x6200	2-181
D1OVL_MATRIX_COEF_1_1	GpuF0MMReg\:0x6204	2-181
D1OVL_MATRIX_COEF_1_2	GpuF0MMReg\:0x6208	2-182
D1OVL_MATRIX_COEF_1_3	GpuF0MMReg\:0x620C	2-182
D1OVL_MATRIX_COEF_1_4	GpuF0MMReg\:0x6210	2-182
D1OVL_MATRIX_COEF_2_1	GpuF0MMReg\:0x6214	2-182
D1OVL_MATRIX_COEF_2_2	GpuF0MMReg\:0x6218	2-183
D1OVL_MATRIX_COEF_2_3	GpuF0MMReg\:0x621C	2-183
D1OVL_MATRIX_COEF_2_4	GpuF0MMReg\:0x6220	2-183
D1OVL_MATRIX_COEF_3_1	GpuF0MMReg\:0x6224	2-183
D1OVL_MATRIX_COEF_3_2	GpuF0MMReg\:0x6228	2-184
D1OVL_MATRIX_COEF_3_3	GpuF0MMReg\:0x622C	2-184
D1OVL_MATRIX_COEF_3_4	GpuF0MMReg\:0x6230	2-184
D1OVL_PWL_TRANSFORM_EN	GpuF0MMReg\:0x6280	2-184
D1OVL_PWL_0TOF	GpuF0MMReg\:0x6284	2-184
D1OVL_PWL_10TO1F	GpuF0MMReg\:0x6288	2-185
D1OVL_PWL_20TO3F	GpuF0MMReg\:0x628C	2-185

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D1OVL_PWL_40TO7F	GpuF0MMReg\:0x6290	2-185
D1OVL_PWL_80TOBF	GpuF0MMReg\:0x6294	2-185
D1OVL_PWL_C0TOFF	GpuF0MMReg\:0x6298	2-186
D1OVL_PWL_100TO13F	GpuF0MMReg\:0x629C	2-186
D1OVL_PWL_140TO17F	GpuF0MMReg\:0x62A0	2-186
D1OVL_PWL_180TO1BF	GpuF0MMReg\:0x62A4	2-186
D1OVL_PWL_1C0TO1FF	GpuF0MMReg\:0x62A8	2-186
D1OVL_PWL_200TO23F	GpuF0MMReg\:0x62AC	2-187
D1OVL_PWL_240TO27F	GpuF0MMReg\:0x62B0	2-187
D1OVL_PWL_280TO2BF	GpuF0MMReg\:0x62B4	2-187
D1OVL_PWL_2C0TO2FF	GpuF0MMReg\:0x62B8	2-187
D1OVL_PWL_300TO33F	GpuF0MMReg\:0x62BC	2-187
D1OVL_PWL_340TO37F	GpuF0MMReg\:0x62C0	2-188
D1OVL_PWL_380TO3BF	GpuF0MMReg\:0x62C4	2-188
D1OVL_PWL_3C0TO3FF	GpuF0MMReg\:0x62C8	2-188
D1OVL_KEY_CONTROL	GpuF0MMReg\:0x6300	2-188
D1GRPH_ALPHA	GpuF0MMReg\:0x6304	2-189
D1OVL_ALPHA	GpuF0MMReg\:0x6308	2-189
D1OVL_ALPHA_CONTROL	GpuF0MMReg\:0x630C	2-189
D1GRPH_KEY_RANGE_RED	GpuF0MMReg\:0x6310	2-190
D1GRPH_KEY_RANGE_GREEN	GpuF0MMReg\:0x6314	2-190
D1GRPH_KEY_RANGE_BLUE	GpuF0MMReg\:0x6318	2-190
D1GRPH_KEY_RANGE_ALPHA	GpuF0MMReg\:0x631C	2-191
D1OVL_KEY_RANGE_RED_CR	GpuF0MMReg\:0x6320	2-191
D1OVL_KEY_RANGE_GREEN_Y	GpuF0MMReg\:0x6324	2-191
D1OVL_KEY_RANGE_BLUE_CB	GpuF0MMReg\:0x6328	2-191
D1OVL_KEY_ALPHA	GpuF0MMReg\:0x632C	2-192
D1GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6380	2-192
D1COLOR_MATRIX_COEF_1_1	GpuF0MMReg\:0x6384	2-192
D1COLOR_MATRIX_COEF_1_2	GpuF0MMReg\:0x6388	2-193
D1COLOR_MATRIX_COEF_1_3	GpuF0MMReg\:0x638C	2-193
D1COLOR_MATRIX_COEF_1_4	GpuF0MMReg\:0x6390	2-193
D1COLOR_MATRIX_COEF_2_1	GpuF0MMReg\:0x6394	2-193
D1COLOR_MATRIX_COEF_2_2	GpuF0MMReg\:0x6398	2-194
D1COLOR_MATRIX_COEF_2_3	GpuF0MMReg\:0x639C	2-194
D1COLOR_MATRIX_COEF_2_4	GpuF0MMReg\:0x63A0	2-194
D1COLOR_MATRIX_COEF_3_1	GpuF0MMReg\:0x63A4	2-194
D1COLOR_MATRIX_COEF_3_2	GpuF0MMReg\:0x63A8	2-195
D1COLOR_MATRIX_COEF_3_3	GpuF0MMReg\:0x63AC	2-195
D1COLOR_MATRIX_COEF_3_4	GpuF0MMReg\:0x63B0	2-195
D1CUR_CONTROL	GpuF0MMReg\:0x6400	2-197
D1CUR_SURFACE_ADDRESS	GpuF0MMReg\:0x6408	2-198
D1CUR_SIZE	GpuF0MMReg\:0x6410	2-198

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D1CUR_POSITION	GpuF0MMReg\:0x6414	2-198
D1CUR_HOT_SPOT	GpuF0MMReg\:0x6418	2-199
D1CUR_COLOR1	GpuF0MMReg\:0x641C	2-199
D1CUR_COLOR2	GpuF0MMReg\:0x6420	2-199
D1CUR_UPDATE	GpuF0MMReg\:0x6424	2-199
D1ICON_CONTROL	GpuF0MMReg\:0x6440	2-200
D1ICON_SURFACE_ADDRESS	GpuF0MMReg\:0x6448	2-200
D1ICON_SIZE	GpuF0MMReg\:0x6450	2-201
D1ICON_START_POSITION	GpuF0MMReg\:0x6454	2-201
D1ICON_COLOR1	GpuF0MMReg\:0x6458	2-201
D1ICON_COLOR2	GpuF0MMReg\:0x645C	2-201
D1ICON_UPDATE	GpuF0MMReg\:0x6460	2-202
DC_LUT_RW_SELECT	GpuF0MMReg\:0x6480	2-241
DC_LUT_RW_MODE	GpuF0MMReg\:0x6484	2-242
DC_LUT_RW_INDEX	GpuF0MMReg\:0x6488	2-242
DC_LUT_SEQ_COLOR	GpuF0MMReg\:0x648C	2-242
DC_LUT_PWL_DATA	GpuF0MMReg\:0x6490	2-242
DC_LUT_30_COLOR	GpuF0MMReg\:0x6494	2-243
DC_LUT_READ_PIPE_SELECT	GpuF0MMReg\:0x6498	2-243
DC_LUT_WRITE_EN_MASK	GpuF0MMReg\:0x649C	2-243
DC_LUT_AUTOFILL	GpuF0MMReg\:0x64A0	2-243
DC_LUTA_CONTROL	GpuF0MMReg\:0x64C0	2-244
DC_LUTA_BLACK_OFFSET_BLUE	GpuF0MMReg\:0x64C4	2-246
DC_LUTA_BLACK_OFFSET_GREEN	GpuF0MMReg\:0x64C8	2-246
DC_LUTA_BLACK_OFFSET_RED	GpuF0MMReg\:0x64CC	2-246
DC_LUTA_WHITE_OFFSET_BLUE	GpuF0MMReg\:0x64D0	2-246
DC_LUTA_WHITE_OFFSET_GREEN	GpuF0MMReg\:0x64D4	2-246
DC_LUTA_WHITE_OFFSET_RED	GpuF0MMReg\:0x64D8	2-246
D1OVL_RT_SKEWCOMMAND	GpuF0MMReg\:0x6500	2-196
D1OVL_RT_SKEWCONTROL	GpuF0MMReg\:0x6504	2-196
D1OVL_RT_BAND_POSITION	GpuF0MMReg\:0x6508	2-196
D1OVL_RT_PROCEED_COND	GpuF0MMReg\:0x650C	2-196
D1OVL_RT_STAT	GpuF0MMReg\:0x6510	2-197
D1_MVP_AFR_FLIP_MODE	GpuF0MMReg\:0x6514	2-207
D1_MVP_AFR_FLIP_FIFO_CNTL	GpuF0MMReg\:0x6518	2-207
D1_MVP_FLIP_LINE_NUM_INSERT	GpuF0MMReg\:0x651C	2-208
D2_MVP_AFR_FLIP_MODE	GpuF0MMReg\:0x65E8	2-241
D2_MVP_AFR_FLIP_FIFO_CNTL	GpuF0MMReg\:0x65EC	2-241
D2_MVP_FLIP_LINE_NUM_INSERT	GpuF0MMReg\:0x65F0	2-241
DC_MVP_LB_CONTROL	GpuF0MMReg\:0x65F4	2-254
D2CRTC_H_TOTAL	GpuF0MMReg\:0x6800	2-271
D2CRTC_H_BLANK_START_END	GpuF0MMReg\:0x6804	2-271
D2CRTC_H_SYNC_A	GpuF0MMReg\:0x6808	2-271

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D2CRTC_H_SYNC_A_CNTL	GpuF0MMReg\:0x680C	2-272
D2CRTC_H_SYNC_B	GpuF0MMReg\:0x6810	2-272
D2CRTC_H_SYNC_B_CNTL	GpuF0MMReg\:0x6814	2-272
D2CRTC_VBI_END	GpuF0MMReg\:0x6818	2-273
D2CRTC_V_TOTAL	GpuF0MMReg\:0x6820	2-273
D2CRTC_V_BLANK_START_END	GpuF0MMReg\:0x6824	2-273
D2CRTC_V_SYNC_A	GpuF0MMReg\:0x6828	2-273
D2CRTC_V_SYNC_A_CNTL	GpuF0MMReg\:0x682C	2-274
D2CRTC_V_SYNC_B	GpuF0MMReg\:0x6830	2-274
D2CRTC_V_SYNC_B_CNTL	GpuF0MMReg\:0x6834	2-274
D2CRTC_MVP_INBAND_CNTL_INSERT	GpuF0MMReg\:0x6838	2-205
D2CRTC_MVP_INBAND_CNTL_INSERT_TIMER	GpuF0MMReg\:0x683C	2-205
D1CRTC_MVP_CRC_CNTL	GpuF0MMReg\:0x6840	2-205
D1CRTC_MVP_CRC_RESULT	GpuF0MMReg\:0x6844	2-206
D1CRTC_MVP_CRC2_CNTL	GpuF0MMReg\:0x6848	2-206
D1CRTC_MVP_CRC2_RESULT	GpuF0MMReg\:0x684C	2-206
D1CRTC_MVP_CONTROL3	GpuF0MMReg\:0x6850	2-206
D1CRTC_MVP_RECEIVE_CNT_CNTL1	GpuF0MMReg\:0x6854	2-207
D1CRTC_MVP_RECEIVE_CNT_CNTL2	GpuF0MMReg\:0x6858	2-207
D2CRTC_MVP_STATUS	GpuF0MMReg\:0x685C	2-240
D2CRTC_TRIGA_CNTL	GpuF0MMReg\:0x6860	2-274
D2CRTC_TRIGA_MANUAL_TRIG	GpuF0MMReg\:0x6864	2-276
D2CRTC_TRIGB_CNTL	GpuF0MMReg\:0x6868	2-276
D2CRTC_TRIGB_MANUAL_TRIG	GpuF0MMReg\:0x686C	2-277
D2CRTC_FORCE_COUNT_NOW_CNTL	GpuF0MMReg\:0x6870	2-277
D2CRTC_FLOW_CONTROL	GpuF0MMReg\:0x6874	2-277
D2CRTC_PIXEL_DATA_READBACK	GpuF0MMReg\:0x6878	2-278
D2CRTC_STEREO_FORCE_NEXT_EYE	GpuF0MMReg\:0x687C	2-278
D2CRTC_CONTROL	GpuF0MMReg\:0x6880	2-279
D2CRTC_BLANK_CONTROL	GpuF0MMReg\:0x6884	2-279
D2CRTC_INTERLACE_CONTROL	GpuF0MMReg\:0x6888	2-280
D2CRTC_INTERLACE_STATUS	GpuF0MMReg\:0x688C	2-280
D2CRTC_BLANK_DATA_COLOR	GpuF0MMReg\:0x6890	2-280
D2CRTC_OVERSCAN_COLOR	GpuF0MMReg\:0x6894	2-281
D2CRTC_BLACK_COLOR	GpuF0MMReg\:0x6898	2-281
D2CRTC_STATUS	GpuF0MMReg\:0x689C	2-281
D2CRTC_STATUS_POSITION	GpuF0MMReg\:0x68A0	2-281
D2CRTC_STATUS_FRAME_COUNT	GpuF0MMReg\:0x68A4	2-282
D2CRTC_STATUS_VF_COUNT	GpuF0MMReg\:0x68A8	2-282
D2CRTC_STATUS_HV_COUNT	GpuF0MMReg\:0x68AC	2-282
D2CRTC_COUNT_RESET	GpuF0MMReg\:0x68B0	2-282
D2CRTC_COUNT_CONTROL	GpuF0MMReg\:0x68B4	2-282
D2CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE	GpuF0MMReg\:0x68B8	2-283

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D2CRTC_VERT_SYNC_CONTROL	GpuF0MMReg\:0x68BC	2-283
D2CRTC_STEREO_STATUS	GpuF0MMReg\:0x68C0	2-283
D2CRTC_STEREO_CONTROL	GpuF0MMReg\:0x68C4	2-283
D2CRTC_SNAPSHOT_STATUS	GpuF0MMReg\:0x68C8	2-284
D2CRTC_SNAPSHOT_CONTROL	GpuF0MMReg\:0x68CC	2-284
D2CRTC_SNAPSHOT_POSITION	GpuF0MMReg\:0x68D0	2-284
D2CRTC_SNAPSHOT_FRAME	GpuF0MMReg\:0x68D4	2-284
D2CRTC_START_LINE_CONTROL	GpuF0MMReg\:0x68D8	2-285
D2CRTC_INTERRUPT_CONTROL	GpuF0MMReg\:0x68DC	2-285
D2MODE_MASTER_UPDATE_LOCK	GpuF0MMReg\:0x68E0	2-285
D2MODE_MASTER_UPDATE_MODE	GpuF0MMReg\:0x68E4	2-289
D2CRTC_UPDATE_LOCK	GpuF0MMReg\:0x68E8	2-286
D2CRTC_DOUBLE_BUFFER_CONTROL	GpuF0MMReg\:0x68EC	2-286
D2CRTC_VGA_PARAMETER_CAPTURE_MODE	GpuF0MMReg\:0x68F0	2-287
D2GRPH_ENABLE	GpuF0MMReg\:0x6900	2-208
D2GRPH_CONTROL	GpuF0MMReg\:0x6904	2-208
D2GRPH_LUT_SEL	GpuF0MMReg\:0x6908	2-210
D2GRPH_SWAP_CNTL	GpuF0MMReg\:0x690C	2-210
D2GRPH_PRIMARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6910	2-211
D2GRPH_SECONDARY_SURFACE_ADDRESS	GpuF0MMReg\:0x6918	2-211
D2GRPH_PITCH	GpuF0MMReg\:0x6920	2-211
D2GRPH_SURFACE_OFFSET_X	GpuF0MMReg\:0x6924	2-212
D2GRPH_SURFACE_OFFSET_Y	GpuF0MMReg\:0x6928	2-212
D2GRPH_X_START	GpuF0MMReg\:0x692C	2-212
D2GRPH_Y_START	GpuF0MMReg\:0x6930	2-212
D2GRPH_X_END	GpuF0MMReg\:0x6934	2-213
D2GRPH_Y_END	GpuF0MMReg\:0x6938	2-213
D2COLOR_SPACE_CONVERT	GpuF0MMReg\:0x693C	2-233
D2OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6940	2-222
D2GRPH_UPDATE	GpuF0MMReg\:0x6944	2-213
D2GRPH_FLIP_CONTROL	GpuF0MMReg\:0x6948	2-214
D2GRPH_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x694C	2-214
D2OVL_ENABLE	GpuF0MMReg\:0x6980	2-215
D2OVL_CONTROL1	GpuF0MMReg\:0x6984	2-215
D2OVL_CONTROL2	GpuF0MMReg\:0x6988	2-216
D2OVL_SWAP_CNTL	GpuF0MMReg\:0x698C	2-216
D2OVL_SURFACE_ADDRESS	GpuF0MMReg\:0x6990	2-217
D2OVL_PITCH	GpuF0MMReg\:0x6998	2-217
D2OVL_SURFACE_OFFSET_X	GpuF0MMReg\:0x699C	2-217
D2OVL_SURFACE_OFFSET_Y	GpuF0MMReg\:0x69A0	2-217
D2OVL_START	GpuF0MMReg\:0x69A4	2-218
D2OVL_END	GpuF0MMReg\:0x69A8	2-218
D2OVL_UPDATE	GpuF0MMReg\:0x69AC	2-218

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D2OVL_SURFACE_ADDRESS_INUSE	GpuF0MMReg\:0x69B0	2-219
D2OVL_DFQ_CONTROL	GpuF0MMReg\:0x69B4	2-219
D2OVL_DFQ_STATUS	GpuF0MMReg\:0x69B8	2-219
D2OVL_MATRIX_TRANSFORM_EN	GpuF0MMReg\:0x6A00	2-219
D2OVL_MATRIX_COEF_1_1	GpuF0MMReg\:0x6A04	2-220
D2OVL_MATRIX_COEF_1_2	GpuF0MMReg\:0x6A08	2-220
D2OVL_MATRIX_COEF_1_3	GpuF0MMReg\:0x6A0C	2-220
D2OVL_MATRIX_COEF_1_4	GpuF0MMReg\:0x6A10	2-220
D2OVL_MATRIX_COEF_2_1	GpuF0MMReg\:0x6A14	2-220
D2OVL_MATRIX_COEF_2_2	GpuF0MMReg\:0x6A18	2-221
D2OVL_MATRIX_COEF_2_3	GpuF0MMReg\:0x6A1C	2-221
D2OVL_MATRIX_COEF_2_4	GpuF0MMReg\:0x6A20	2-221
D2OVL_MATRIX_COEF_3_1	GpuF0MMReg\:0x6A24	2-221
D2OVL_MATRIX_COEF_3_2	GpuF0MMReg\:0x6A28	2-222
D2OVL_MATRIX_COEF_3_3	GpuF0MMReg\:0x6A2C	2-222
D2OVL_MATRIX_COEF_3_4	GpuF0MMReg\:0x6A30	2-222
D2OVL_PWL_TRANSFORM_EN	GpuF0MMReg\:0x6A80	2-223
D2OVL_PWL_0TOF	GpuF0MMReg\:0x6A84	2-223
D2OVL_PWL_10TO1F	GpuF0MMReg\:0x6A88	2-223
D2OVL_PWL_20TO3F	GpuF0MMReg\:0x6A8C	2-223
D2OVL_PWL_40TO7F	GpuF0MMReg\:0x6A90	2-223
D2OVL_PWL_80TOBF	GpuF0MMReg\:0x6A94	2-224
D2OVL_PWL_C0TOFF	GpuF0MMReg\:0x6A98	2-224
D2OVL_PWL_100TO13F	GpuF0MMReg\:0x6A9C	2-224
D2OVL_PWL_140TO17F	GpuF0MMReg\:0x6AA0	2-224
D2OVL_PWL_180TO1BF	GpuF0MMReg\:0x6AA4	2-225
D2OVL_PWL_1C0TO1FF	GpuF0MMReg\:0x6AA8	2-225
D2OVL_PWL_200TO23F	GpuF0MMReg\:0x6AAC	2-225
D2OVL_PWL_240TO27F	GpuF0MMReg\:0x6AB0	2-225
D2OVL_PWL_280TO2BF	GpuF0MMReg\:0x6AB4	2-225
D2OVL_PWL_2C0TO2FF	GpuF0MMReg\:0x6AB8	2-226
D2OVL_PWL_300TO33F	GpuF0MMReg\:0x6ABC	2-226
D2OVL_PWL_340TO37F	GpuF0MMReg\:0x6AC0	2-226
D2OVL_PWL_380TO3BF	GpuF0MMReg\:0x6AC4	2-226
D2OVL_PWL_3C0TO3FF	GpuF0MMReg\:0x6AC8	2-226
D2OVL_KEY_CONTROL	GpuF0MMReg\:0x6B00	2-227
D2GRPH_ALPHA	GpuF0MMReg\:0x6B04	2-227
D2OVL_ALPHA	GpuF0MMReg\:0x6B08	2-227
D2OVL_ALPHA_CONTROL	GpuF0MMReg\:0x6B0C	2-228
D2GRPH_KEY_RANGE_RED	GpuF0MMReg\:0x6B10	2-228
D2GRPH_KEY_RANGE_GREEN	GpuF0MMReg\:0x6B14	2-228
D2GRPH_KEY_RANGE_BLUE	GpuF0MMReg\:0x6B18	2-229
D2GRPH_KEY_RANGE_ALPHA	GpuF0MMReg\:0x6B1C	2-229

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
D2OVL_KEY_RANGE_RED_CR	GpuF0MMReg\:0x6B20	2-229
D2OVL_KEY_RANGE_GREEN_Y	GpuF0MMReg\:0x6B24	2-230
D2OVL_KEY_RANGE_BLUE_CB	GpuF0MMReg\:0x6B28	2-230
D2OVL_KEY_ALPHA	GpuF0MMReg\:0x6B2C	2-230
D2GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL	GpuF0MMReg\:0x6B80	2-230
D2COLOR_MATRIX_COEF_1_1	GpuF0MMReg\:0x6B84	2-231
D2COLOR_MATRIX_COEF_1_2	GpuF0MMReg\:0x6B88	2-231
D2COLOR_MATRIX_COEF_1_3	GpuF0MMReg\:0x6B8C	2-231
D2COLOR_MATRIX_COEF_1_4	GpuF0MMReg\:0x6B90	2-231
D2COLOR_MATRIX_COEF_2_1	GpuF0MMReg\:0x6B94	2-232
D2COLOR_MATRIX_COEF_2_2	GpuF0MMReg\:0x6B98	2-232
D2COLOR_MATRIX_COEF_2_3	GpuF0MMReg\:0x6B9C	2-232
D2COLOR_MATRIX_COEF_2_4	GpuF0MMReg\:0x6BA0	2-232
D2COLOR_MATRIX_COEF_3_1	GpuF0MMReg\:0x6BA4	2-232
D2COLOR_MATRIX_COEF_3_2	GpuF0MMReg\:0x6BA8	2-233
D2COLOR_MATRIX_COEF_3_3	GpuF0MMReg\:0x6BAC	2-233
D2COLOR_MATRIX_COEF_3_4	GpuF0MMReg\:0x6BB0	2-233
D2CUR_CONTROL	GpuF0MMReg\:0x6C00	2-236
D2CUR_SURFACE_ADDRESS	GpuF0MMReg\:0x6C08	2-236
D2CUR_SIZE	GpuF0MMReg\:0x6C10	2-236
D2CUR_POSITION	GpuF0MMReg\:0x6C14	2-236
D2CUR_HOT_SPOT	GpuF0MMReg\:0x6C18	2-237
D2CUR_COLOR1	GpuF0MMReg\:0x6C1C	2-237
D2CUR_COLOR2	GpuF0MMReg\:0x6C20	2-237
D2CUR_UPDATE	GpuF0MMReg\:0x6C24	2-237
D2ICON_CONTROL	GpuF0MMReg\:0x6C40	2-238
D2ICON_SURFACE_ADDRESS	GpuF0MMReg\:0x6C48	2-238
D2ICON_SIZE	GpuF0MMReg\:0x6C50	2-239
D2ICON_START_POSITION	GpuF0MMReg\:0x6C54	2-239
D2ICON_COLOR1	GpuF0MMReg\:0x6C58	2-239
D2ICON_COLOR2	GpuF0MMReg\:0x6C5C	2-239
D2ICON_UPDATE	GpuF0MMReg\:0x6C60	2-240
DCP_CRC_CONTROL	GpuF0MMReg\:0x6C80	2-250
DCP_CRC_MASK	GpuF0MMReg\:0x6C84	2-250
DCP_CRC_P0_CURRENT	GpuF0MMReg\:0x6C88	2-250
DCP_CRC_P1_CURRENT	GpuF0MMReg\:0x6C8C	2-250
DCP_CRC_P0_LAST	GpuF0MMReg\:0x6C90	2-250
DCP_CRC_P1_LAST	GpuF0MMReg\:0x6C94	2-251
DCP_TILING_CONFIG	GpuF0MMReg\:0x6CA0	2-251
DCP_MULTI_CHIP_CNTL	GpuF0MMReg\:0x6CA4	2-252
DMIF_CONTROL	GpuF0MMReg\:0x6CB0	2-252
DMIF_STATUS	GpuF0MMReg\:0x6CB4	2-253
MCIF_CONTROL	GpuF0MMReg\:0x6CB8	2-253

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
DCP_LB_DATA_GAP_BETWEEN_CHUNK	GpuF0MMReg\:0x6CBC	2-254
DC_LUTB_CONTROL	GpuF0MMReg\:0x6CC0	2-247
DC_LUTB_BLACK_OFFSET_BLUE	GpuF0MMReg\:0x6CC4	2-249
DC_LUTB_BLACK_OFFSET_GREEN	GpuF0MMReg\:0x6CC8	2-249
DC_LUTB_BLACK_OFFSET_RED	GpuF0MMReg\:0x6CCC	2-249
DC_LUTB_WHITE_OFFSET_BLUE	GpuF0MMReg\:0x6CD0	2-249
DC_LUTB_WHITE_OFFSET_GREEN	GpuF0MMReg\:0x6CD4	2-249
DC_LUTB_WHITE_OFFSET_RED	GpuF0MMReg\:0x6CD8	2-249
D2OVL_RT_SKEWCOMMAND	GpuF0MMReg\:0x6D00	2-234
D2OVL_RT_SKEWCONTROL	GpuF0MMReg\:0x6D04	2-234
D2OVL_RT_BAND_POSITION	GpuF0MMReg\:0x6D08	2-234
D2OVL_RT_PROCEED_COND	GpuF0MMReg\:0x6D0C	2-235
D2OVL_RT_STAT	GpuF0MMReg\:0x6D10	2-235
DACA_ENABLE	GpuF0MMReg\:0x7800	2-288
DACA_SOURCE_SELECT	GpuF0MMReg\:0x7804	2-288
DACA_CRC_EN	GpuF0MMReg\:0x7808	2-288
DACA_CRC_CONTROL	GpuF0MMReg\:0x780C	2-288
DACA_CRC_SIG_RGB_MASK	GpuF0MMReg\:0x7810	2-289
DACA_CRC_SIG_CONTROL_MASK	GpuF0MMReg\:0x7814	2-289
DACA_CRC_SIG_RGB	GpuF0MMReg\:0x7818	2-289
DACA_CRC_SIG_CONTROL	GpuF0MMReg\:0x781C	2-289
DACA_SYNC_TRISTATE_CONTROL	GpuF0MMReg\:0x7820	2-289
DACA_SYNC_SELECT	GpuF0MMReg\:0x7824	2-290
DACA_AUTODETECT_CONTROL	GpuF0MMReg\:0x7828	2-290
DACA_AUTODETECT_CONTROL2	GpuF0MMReg\:0x782C	2-290
DACA_AUTODETECT_CONTROL3	GpuF0MMReg\:0x7830	2-290
DACA_AUTODETECT_STATUS	GpuF0MMReg\:0x7834	2-291
DACA_AUTODETECT_INT_CONTROL	GpuF0MMReg\:0x7838	2-291
DACA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x783C	2-291
DACA_FORCE_DATA	GpuF0MMReg\:0x7840	2-292
DACA_POWERDOWN	GpuF0MMReg\:0x7850	2-292
DACA_CONTROL1	GpuF0MMReg\:0x7854	2-292
DACA_CONTROL2	GpuF0MMReg\:0x7858	2-292
DACA_COMPARATOR_ENABLE	GpuF0MMReg\:0x785C	2-293
DACA_COMPARATOR_OUTPUT	GpuF0MMReg\:0x7860	2-294
DACA_TEST_ENABLE	GpuF0MMReg\:0x7864	2-294
DACA_PWR_CNTL	GpuF0MMReg\:0x7868	2-294
DACA_DFT_CONFIG	GpuF0MMReg\:0x786C	2-294
TMDSA_CNTL	GpuF0MMReg\:0x7880	2-301
TMDSA_SOURCE_SELECT	GpuF0MMReg\:0x7884	2-302
TMDSA_COLOR_FORMAT	GpuF0MMReg\:0x7888	2-302
TMDSA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x788C	2-302
TMDSA_FORCE_DATA	GpuF0MMReg\:0x7890	2-302

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
TMDSA_BIT_DEPTH_CONTROL	GpuF0MMReg\:0x7894	2-303
TMDSA_CONTROL_CHAR	GpuF0MMReg\:0x7898	2-303
TMDSA_CONTROL0_FEEDBACK	GpuF0MMReg\:0x789C	2-304
TMDSA_STEREOSYNC_CTL_SEL	GpuF0MMReg\:0x78A0	2-304
TMDSA_SYNC_CHAR_PATTERN_SEL	GpuF0MMReg\:0x78A4	2-304
TMDSA_SYNC_CHAR_PATTERN_0_1	GpuF0MMReg\:0x78A8	2-304
TMDSA_SYNC_CHAR_PATTERN_2_3	GpuF0MMReg\:0x78AC	2-304
TMDSA_CRC_CNTL	GpuF0MMReg\:0x78B0	2-305
TMDSA_CRC_SIG_MASK	GpuF0MMReg\:0x78B4	2-305
TMDSA_CRC_SIG_RGB	GpuF0MMReg\:0x78B8	2-305
TMDSA_2ND_CRC_RESULT	GpuF0MMReg\:0x78BC	2-306
TMDSA_TEST_PATTERN	GpuF0MMReg\:0x78C0	2-306
TMDSA_RANDOM_PATTERN_SEED	GpuF0MMReg\:0x78C4	2-307
TMDSA_DEBUG	GpuF0MMReg\:0x78C8	2-307
TMDSA_CTL_BITS	GpuF0MMReg\:0x78CC	2-307
TMDSA_DCBALANCER_CONTROL	GpuF0MMReg\:0x78D0	2-307
TMDSA_RED_BLUE_SWITCH	GpuF0MMReg\:0x78D4	2-307
TMDSA_DATA_SYNCHRONIZATION	GpuF0MMReg\:0x78DC	2-308
TMDSA_CTL0_1_GEN_CNTL	GpuF0MMReg\:0x78E0	2-308
TMDSA_CTL2_3_GEN_CNTL	GpuF0MMReg\:0x78E4	2-309
TMDSA_TRANSMITTER_ENABLE	GpuF0MMReg\:0x7904	2-310
TMDSA_LOAD_DETECT	GpuF0MMReg\:0x7908	2-311
TMDSA_PLL_ADJUST	GpuF0MMReg\:0x790C	2-311
TMDSA_TRANSMITTER_CONTROL	GpuF0MMReg\:0x7910	2-312
TMDSA_REG_TEST_OUTPUTA	GpuF0MMReg\:0x7914	2-313
TMDSA_TRANSMITTER_DEBUG	GpuF0MMReg\:0x7918	2-313
TMDSA_DITHER_RAND_SEED	GpuF0MMReg\:0x791C	2-313
TMDSA_TRANSMITTER_ADJUST	GpuF0MMReg\:0x7920	2-314
TMDSA_REG_TEST_OUTPUTB	GpuF0MMReg\:0x7924	2-314
LVTMA_SPLIT_TRANSMITTER_ENABLE	GpuF0MMReg\:0x792C	2-336
LVTMA_SPLIT_LOAD_DETECT	GpuF0MMReg\:0x7930	2-337
LVTMA_SPLIT_PLL_ADJUST	GpuF0MMReg\:0x7934	2-337
LVTMA_SPLIT_TRANSMITTER_CONTROL	GpuF0MMReg\:0x7938	2-337
LVTMA_SPLIT_TRANSMITTER_ADJUST	GpuF0MMReg\:0x793C	2-338
DVOA_ENABLE	GpuF0MMReg\:0x7980	2-314
DVOA_SOURCE_SELECT	GpuF0MMReg\:0x7984	2-315
DVOA_BIT_DEPTH_CONTROL	GpuF0MMReg\:0x7988	2-315
DVOA_OUTPUT	GpuF0MMReg\:0x798C	2-316
DVOA_CONTROL	GpuF0MMReg\:0x7990	2-316
DVOA_CRC_EN	GpuF0MMReg\:0x7994	2-316
DVOA_CRC_CONTROL	GpuF0MMReg\:0x7998	2-317
DVOA_CRC_SIG_MASK1	GpuF0MMReg\:0x799C	2-317
DVOA_CRC_SIG_MASK2	GpuF0MMReg\:0x79A0	2-317

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
DVOA_CRC_SIG_RESULT1	GpuF0MMReg\:0x79A4	2-317
DVOA_CRC_SIG_RESULT2	GpuF0MMReg\:0x79A8	2-318
DVOA_CRC2_SIG_MASK	GpuF0MMReg\:0x79AC	2-318
DVOA_CRC2_SIG_RESULT	GpuF0MMReg\:0x79B0	2-318
DVOA_STRENGTH_CONTROL	GpuF0MMReg\:0x79B4	2-318
DVOA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x79B8	2-319
DVOA_FORCE_DATA	GpuF0MMReg\:0x79BC	2-319
DACB_ENABLE	GpuF0MMReg\:0x7A00	2-294
DACB_SOURCE_SELECT	GpuF0MMReg\:0x7A04	2-295
DACB_CRC_EN	GpuF0MMReg\:0x7A08	2-295
DACB_CRC_CONTROL	GpuF0MMReg\:0x7A0C	2-295
DACB_CRC_SIG_RGB_MASK	GpuF0MMReg\:0x7A10	2-295
DACB_CRC_SIG_CONTROL_MASK	GpuF0MMReg\:0x7A14	2-296
DACB_CRC_SIG_RGB	GpuF0MMReg\:0x7A18	2-296
DACB_CRC_SIG_CONTROL	GpuF0MMReg\:0x7A1C	2-296
DACB_SYNC_TRISTATE_CONTROL	GpuF0MMReg\:0x7A20	2-296
DACB_SYNC_SELECT	GpuF0MMReg\:0x7A24	2-296
DACB_AUTODETECT_CONTROL	GpuF0MMReg\:0x7A28	2-296
DACB_AUTODETECT_CONTROL2	GpuF0MMReg\:0x7A2C	2-297
DACB_AUTODETECT_CONTROL3	GpuF0MMReg\:0x7A30	2-297
DACB_AUTODETECT_STATUS	GpuF0MMReg\:0x7A34	2-297
DACB_AUTODETECT_INT_CONTROL	GpuF0MMReg\:0x7A38	2-298
DACB_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x7A3C	2-298
DACB_FORCE_DATA	GpuF0MMReg\:0x7A40	2-298
DACB_POWERDOWN	GpuF0MMReg\:0x7A50	2-298
DACB_CONTROL1	GpuF0MMReg\:0x7A54	2-299
DACB_CONTROL2	GpuF0MMReg\:0x7A58	2-299
DACB_COMPARATOR_ENABLE	GpuF0MMReg\:0x7A5C	2-299
DACB_COMPARATOR_OUTPUT	GpuF0MMReg\:0x7A60	2-300
DACB_TEST_ENABLE	GpuF0MMReg\:0x7A64	2-300
DACB_PWR_CNTL	GpuF0MMReg\:0x7A68	2-301
LVTMA_CNTL	GpuF0MMReg\:0x7A80	2-319
LVTMA_SOURCE_SELECT	GpuF0MMReg\:0x7A84	2-320
LVTMA_COLOR_FORMAT	GpuF0MMReg\:0x7A88	2-320
LVTMA_FORCE_OUTPUT_CNTL	GpuF0MMReg\:0x7A8C	2-321
LVTMA_FORCE_DATA	GpuF0MMReg\:0x7A90	2-321
LVTMA_BIT_DEPTH_CONTROL	GpuF0MMReg\:0x7A94	2-321
LVTMA_CONTROL_CHAR	GpuF0MMReg\:0x7A98	2-322
LVTMA_CONTROL0_FEEDBACK	GpuF0MMReg\:0x7A9C	2-322
LVTMA_STEREOSYNC_CTL_SEL	GpuF0MMReg\:0x7AA0	2-322
LVTMA_SYNC_CHAR_PATTERN_SEL	GpuF0MMReg\:0x7AA4	2-323
LVTMA_SYNC_CHAR_PATTERN_0_1	GpuF0MMReg\:0x7AA8	2-323
LVTMA_SYNC_CHAR_PATTERN_2_3	GpuF0MMReg\:0x7AAC	2-323

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
LVTMA_CRC_CNTL	GpuF0MMReg\:0x7AB0	2-323
LVTMA_CRC_SIG_MASK	GpuF0MMReg\:0x7AB4	2-324
LVTMA_CRC_SIG_RGB	GpuF0MMReg\:0x7AB8	2-324
LVTMA_2ND_CRC_RESULT	GpuF0MMReg\:0x7ABC	2-324
LVTMA_TEST_PATTERN	GpuF0MMReg\:0x7AC0	2-324
LVTMA_RANDOM_PATTERN_SEED	GpuF0MMReg\:0x7AC4	2-325
LVTMA_DEBUG	GpuF0MMReg\:0x7AC8	2-325
LVTMA_CTL_BITS	GpuF0MMReg\:0x7ACC	2-326
LVTMA_DCBALANCER_CONTROL	GpuF0MMReg\:0x7AD0	2-326
LVTMA_RED_BLUE_SWITCH	GpuF0MMReg\:0x7AD4	2-326
LVTMA_DATA_SYNCHRONIZATION	GpuF0MMReg\:0x7ADC	2-326
LVTMA_CTL0_1_GEN_CNTL	GpuF0MMReg\:0x7AE0	2-327
LVTMA_CTL2_3_GEN_CNTL	GpuF0MMReg\:0x7AE4	2-328
LVTMA_PWRSEQ_REF_DIV	GpuF0MMReg\:0x7AE8	2-329
LVTMA_PWRSEQ_DELAY1	GpuF0MMReg\:0x7AEC	2-329
LVTMA_PWRSEQ_DELAY2	GpuF0MMReg\:0x7AF0	2-330
LVTMA_PWRSEQ_CNTL	GpuF0MMReg\:0x7AF4	2-330
LVTMA_PWRSEQ_STATE	GpuF0MMReg\:0x7AF8	2-331
LVTMA_BL_MOD_CNTL	GpuF0MMReg\:0x7AFC	2-332
LVTMA_LVTM_DATA_CNTL	GpuF0MMReg\:0x7B00	2-332
LVTMA_MODE	GpuF0MMReg\:0x7B04	2-332
LVTMA_TRANSMITTER_ENABLE	GpuF0MMReg\:0x7B08	2-333
LVTMA_LOAD_DETECT	GpuF0MMReg\:0x7B0C	2-333
LVTMA_MACRO_CONTROL	GpuF0MMReg\:0x7B10	2-334
LVTMA_TRANSMITTER_CONTROL	GpuF0MMReg\:0x7B14	2-334
LVTMA_REG_TEST_OUTPUT	GpuF0MMReg\:0x7B18	2-335
LVTMA_TRANSMITTER_DEBUG	GpuF0MMReg\:0x7B1C	2-335
LVTMA_DITHER_RAND_SEED	GpuF0MMReg\:0x7B20	2-336
LVTMA_TRANSMITTER_ADJUST	GpuF0MMReg\:0x7B24	2-336
LVTMA_PREEMPHASIS_CONTROL	GpuF0MMReg\:0x7B28	2-336
DC_HOT_PLUG_DETECT1_CONTROL	GpuF0MMReg\:0x7D00	2-339
DC_HOT_PLUG_DETECT1_INT_STATUS	GpuF0MMReg\:0x7D04	2-339
DC_HOT_PLUG_DETECT1_INT_CONTROL	GpuF0MMReg\:0x7D08	2-339
DC_HOT_PLUG_DETECT2_CONTROL	GpuF0MMReg\:0x7D10	2-339
DC_HOT_PLUG_DETECT2_INT_STATUS	GpuF0MMReg\:0x7D14	2-340
DC_HOT_PLUG_DETECT2_INT_CONTROL	GpuF0MMReg\:0x7D18	2-340
DC_HOT_PLUG_DETECT_CLOCK_CONTROL	GpuF0MMReg\:0x7D20	2-340
DC_HOT_PLUG_DETECT3_CONTROL	GpuF0MMReg\:0x7D24	2-340
DC_HOT_PLUG_DETECT3_INT_STATUS	GpuF0MMReg\:0x7D28	2-341
DC_HOT_PLUG_DETECT3_INT_CONTROL	GpuF0MMReg\:0x7D2C	2-341
DC_GENERICA	GpuF0MMReg\:0x7DC0	2-342
DC_GENERICB	GpuF0MMReg\:0x7DC4	2-342
DC_PAD_EXTERN_SIG	GpuF0MMReg\:0x7DCC	2-342

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
DC_REF_CLK_CNTL	GpuF0MMReg\:0x7DD4	2-343
DC_GPIO_GENERIC_MASK	GpuF0MMReg\:0x7DE0	2-349
DC_GPIO_GENERIC_A	GpuF0MMReg\:0x7DE4	2-349
DC_GPIO_GENERIC_EN	GpuF0MMReg\:0x7DE8	2-349
DC_GPIO_GENERIC_Y	GpuF0MMReg\:0x7DEC	2-349
DC_GPIO_DDC4_MASK	GpuF0MMReg\:0x7E00	2-350
DC_GPIO_DDC4_A	GpuF0MMReg\:0x7E04	2-350
DC_GPIO_DDC4_EN	GpuF0MMReg\:0x7E08	2-350
DC_GPIO_DDC4_Y	GpuF0MMReg\:0x7E0C	2-350
DC_GPIO_DEBUG	GpuF0MMReg\:0x7E2C	2-350
DC_GPIO_DVODATA_MASK	GpuF0MMReg\:0x7E30	2-351
DC_GPIO_DVODATA_A	GpuF0MMReg\:0x7E34	2-351
DC_GPIO_DVODATA_EN	GpuF0MMReg\:0x7E38	2-351
DC_GPIO_DVODATA_Y	GpuF0MMReg\:0x7E3C	2-352
DC_GPIO_DDC1_MASK	GpuF0MMReg\:0x7E40	2-352
DC_GPIO_DDC1_A	GpuF0MMReg\:0x7E44	2-352
DC_GPIO_DDC1_EN	GpuF0MMReg\:0x7E48	2-353
DC_GPIO_DDC1_Y	GpuF0MMReg\:0x7E4C	2-353
DC_GPIO_DDC2_MASK	GpuF0MMReg\:0x7E50	2-353
DC_GPIO_DDC2_A	GpuF0MMReg\:0x7E54	2-353
DC_GPIO_DDC2_EN	GpuF0MMReg\:0x7E58	2-354
DC_GPIO_DDC2_Y	GpuF0MMReg\:0x7E5C	2-354
DC_GPIO_DDC3_MASK	GpuF0MMReg\:0x7E60	2-354
DC_GPIO_DDC3_A	GpuF0MMReg\:0x7E64	2-354
DC_GPIO_DDC3_EN	GpuF0MMReg\:0x7E68	2-355
DC_GPIO_DDC3_Y	GpuF0MMReg\:0x7E6C	2-355
DC_GPIO_SYNCA_MASK	GpuF0MMReg\:0x7E70	2-355
DC_GPIO_SYNCA_A	GpuF0MMReg\:0x7E74	2-355
DC_GPIO_SYNCA_EN	GpuF0MMReg\:0x7E78	2-356
DC_GPIO_SYNCA_Y	GpuF0MMReg\:0x7E7C	2-356
DC_GPIO_SYNCB_MASK	GpuF0MMReg\:0x7E80	2-356
DC_GPIO_SYNCB_A	GpuF0MMReg\:0x7E84	2-356
DC_GPIO_SYNCB_EN	GpuF0MMReg\:0x7E88	2-357
DC_GPIO_SYNCB_Y	GpuF0MMReg\:0x7E8C	2-357
DC_GPIO_HPD_MASK	GpuF0MMReg\:0x7E90	2-357
DC_GPIO_HPD_A	GpuF0MMReg\:0x7E94	2-357
DC_GPIO_HPD_EN	GpuF0MMReg\:0x7E98	2-358
DC_GPIO_HPD_Y	GpuF0MMReg\:0x7E9C	2-358
DC_GPIO_PWRSEQ_MASK	GpuF0MMReg\:0x7EA0	2-358
DC_GPIO_PWRSEQ_A	GpuF0MMReg\:0x7EA4	2-359
DC_GPIO_PWRSEQ_EN	GpuF0MMReg\:0x7EA8	2-359
DC_GPIO_PWRSEQ_Y	GpuF0MMReg\:0x7EAC	2-359
CAPTURE_START_STATUS	GpuF0MMReg\:0x7ED0	2-347

Table A-6 Display Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
DC_GPIO_PAD_STRENGTH_1	*GpuF0MMReg\:0x7ED4*	*2-359*
DC_GPIO_PAD_STRENGTH_2	*GpuF0MMReg\:0x7ED8*	*2-360*
DISP_INTERRUPT_STATUS	*GpuF0MMReg\:0x7EDC*	*2-343*
DOUT_POWER_MANAGEMENT_CNTL	*GpuF0MMReg\:0x7EE0*	*2-346*
DISP_INTERRUPT_STATUS_CONTINUE	*GpuF0MMReg\:0x7EE8*	*2-345*
DISP_TIMER_CONTROL	*GpuF0MMReg\:0x7EF0*	*2-347*

A.8 Host Interface Decode Space Registers Sorted by Name

Table A-7 Host Interface Decode Space Registers Sorted by Name

Register Name	Address	Page
BUS_CNTL	*GpuF0MMReg\:0x5420*	*2-51*
CONFIG_APER_SIZE	*GpuF0MMReg\:0x5430*	*2-53*
CONFIG_CNTL	*GpuF0MMReg\:0x5424*	*2-52*
CONFIG_MEMSIZE	*GpuF0MMReg\:0x5428*	*2-52*
CONFIG_REG_APER_SIZE	*GpuF0MMReg\:0x5434*	*2-53*
GENENB	*GpuF0MMReg\:0x3C* *VGA_IO\:0x3C33*	*2-139*
GENMO_RD	*GpuF0MMReg\:0x3CC* *VGA_IO\:0x3CC*	*2-139*
GENMO_WT	*GpuF0MMReg\:0x3C2* *VGA_IO\:0x3C2*	*2-139*
MM_DATA	*GpuF0MMReg\:0x4* *GpuIOReg\:0x4*	*2-51*
MM_INDEX	*GpuF0MMReg\:0x0* *GpuIOReg\:0x0*	*2-51*

A.9 Memory Controller Registers Sorted By Name

Table A-8 Memory Controller Registers Sorted by Name

Register Name	Address	Page
MC_CONFIG	GpuF0MMReg\:0x2000	2-2
MC_IMP_CNTL	GpuF0MMReg\:0x26D4	2-10
MC_IMP_DEBUG	GpuF0MMReg\:0x2878	2-10
MC_IMP_STATUS	GpuF0MMReg\:0x2874	2-10
MC_IO_A_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2750	2-33
MC_IO_A_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2754	2-34
MC_IO_A_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27D8	2-45
MC_IO_A_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27DC	2-45
MC_IO_CK_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2730	2-31
MC_IO_CK_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2734	2-31
MC_IO_CK_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27B8	2-43
MC_IO_CK_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27BC	2-43
MC_IO_CMD_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2738	2-31
MC_IO_CMD_PAD_CNTL_D0_I1	GpuF0MMReg\:0x273C	2-32
MC_IO_CMD_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27C0	2-43
MC_IO_CMD_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27C4	2-43
MC_IO_DQ_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2740	2-32
MC_IO_DQ_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2744	2-32
MC_IO_DQ_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27C8	2-44
MC_IO_DQ_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27CC	2-44
MC_IO_PAD_CNTL	GpuF0MMReg\:0x2700	2-11
MC_IO_PAD_CNTL_D0	GpuF0MMReg\:0x27F0	2-25
MC_IO_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2704	2-28
MC_IO_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2708	2-28
MC_IO_PAD_CNTL_D1	GpuF0MMReg\:0x27F4	2-25
MC_IO_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2790	2-39
MC_IO_PAD_CNTL_D1_I1	GpuF0MMReg\:0x2794	2-40
MC_IO_QS_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2748	2-33
MC_IO_QS_PAD_CNTL_D0_I1	GpuF0MMReg\:0x274C	2-33
MC_IO_QS_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27D0	2-44
MC_IO_QS_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27D4	2-45
MC_IO_RD_DQ_CNTL_D0_I0	GpuF0MMReg\:0x2710	2-29
MC_IO_RD_DQ_CNTL_D0_I1	GpuF0MMReg\:0x2714	2-29
MC_IO_RD_DQ_CNTL_D1_I0	GpuF0MMReg\:0x2798	2-41
MC_IO_RD_DQ_CNTL_D1_I1	GpuF0MMReg\:0x279C	2-41
MC_IO_RD_QS_CNTL_D0_I0	GpuF0MMReg\:0x2718	2-30
MC_IO_RD_QS_CNTL_D0_I1	GpuF0MMReg\:0x271C	2-30
MC_IO_RD_QS_CNTL_D1_I0	GpuF0MMReg\:0x27A0	2-41
MC_IO_RD_QS_CNTL_D1_I1	GpuF0MMReg\:0x27A4	2-42
MC_IO_RD_QS2_CNTL_D0_I0	GpuF0MMReg\:0x2720	2-30
MC_IO_RD_QS2_CNTL_D0_I1	GpuF0MMReg\:0x2724	2-30

Table A-8 Memory Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
MC_IO_RD_QS2_CNTL_D1_I0	GpuF0MMReg\:0x27A8	2-42
MC_IO_RD_QS2_CNTL_D1_I1	GpuF0MMReg\:0x27AC	2-42
MC_IO_RD_STR_NCNTL_B0_D0	GpuF0MMReg\:0x26E8	2-35
MC_IO_RD_STR_NCNTL_B0_D1	GpuF0MMReg\:0x2820	2-46
MC_IO_RD_STR_NCNTL_B1_D0	GpuF0MMReg\:0x280C	2-35
MC_IO_RD_STR_NCNTL_B1_D1	GpuF0MMReg\:0x2828	2-46
MC_IO_RD_STR_NCNTL_B2_D0	GpuF0MMReg\:0x26F8	2-35
MC_IO_RD_STR_NCNTL_B2_D1	GpuF0MMReg\:0x2830	2-47
MC_IO_RD_STR_NCNTL_B3_D0	GpuF0MMReg\:0x27F8	2-35
MC_IO_RD_STR_NCNTL_B3_D1	GpuF0MMReg\:0x2838	2-47
MC_IO_RD_STR_NCNTL_B4_D0	GpuF0MMReg\:0x2800	2-36
MC_IO_RD_STR_NCNTL_B4_D1	GpuF0MMReg\:0x2840	2-47
MC_IO_RD_STR_NCNTL_B5_D0	GpuF0MMReg\:0x2808	2-36
MC_IO_RD_STR_NCNTL_B5_D1	GpuF0MMReg\:0x2848	2-47
MC_IO_RD_STR_NCNTL_B6_D0	GpuF0MMReg\:0x2810	2-36
MC_IO_RD_STR_NCNTL_B6_D1	GpuF0MMReg\:0x2850	2-48
MC_IO_RD_STR_NCNTL_B7_D0	GpuF0MMReg\:0x2818	2-37
MC_IO_RD_STR_NCNTL_B7_D1	GpuF0MMReg\:0x2858	2-48
MC_IO_WR_CNTL_D0_I0	GpuF0MMReg\:0x2728	2-30
MC_IO_WR_CNTL_D0_I1	GpuF0MMReg\:0x272C	2-31
MC_IO_WR_CNTL_D1_I0	GpuF0MMReg\:0x27B0	2-42
MC_IO_WR_CNTL_D1_I1	GpuF0MMReg\:0x27B4	2-42
MC_IO_WR_DQ_CNTL_D0_I0	GpuF0MMReg\:0x2758	2-34
MC_IO_WR_DQ_CNTL_D0_I1	GpuF0MMReg\:0x275C	2-34
MC_IO_WR_DQ_CNTL_D1_I0	GpuF0MMReg\:0x27E0	2-46
MC_IO_WR_DQ_CNTL_D1_I1	GpuF0MMReg\:0x27E4	2-46
MC_IO_WR_QS_CNTL_D0_I0	GpuF0MMReg\:0x2760	2-34
MC_IO_WR_QS_CNTL_D0_I1	GpuF0MMReg\:0x2764	2-34
MC_IO_WR_QS_CNTL_D1_I0	GpuF0MMReg\:0x27E8	2-46
MC_IO_WR_QS_CNTL_D1_I1	GpuF0MMReg\:0x27EC	2-46
MC_NPL_STATUS	GpuF0MMReg\:0x2888	2-50
MC_PMG_CFG	GpuF0MMReg\:0x26D0	2-9
MC_PMG_CMD	GpuF0MMReg\:0x26CC	2-8
MC_SEQ_A_PAD_CNTL_D0_I0	GpuF0MMReg\:0x268C	2-27
MC_SEQ_A_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2690	2-27
MC_SEQ_A_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2788	2-39
MC_SEQ_A_PAD_CNTL_D1_I1	GpuF0MMReg\:0x278C	2-39
MC_SEQ_CAS_TIMING_B	GpuF0MMReg\:0x2620	2-5
MC_SEQ_CAS_TIMING_C	GpuF0MMReg\:0x2640	2-7
MC_SEQ_CAS_TIMING_P	GpuF0MMReg\:0x2610	2-4
MC_SEQ_CAS_TIMING_S	GpuF0MMReg\:0x2630	2-6
MC_SEQ_CK_PAD_CNTL_D0_I0	GpuF0MMReg\:0x266C	2-25
MC_SEQ_CK_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2670	2-25

Table A-8 Memory Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
MC_SEQ_CK_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2768	2-37
MC_SEQ_CK_PAD_CNTL_D1_I1	GpuF0MMReg\:0x276C	2-37
MC_SEQ_CMD	GpuF0MMReg\:0x26C4	2-8
MC_SEQ_CMD_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2674	2-26
MC_SEQ_CMD_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2678	2-26
MC_SEQ_CMD_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2770	2-37
MC_SEQ_CMD_PAD_CNTL_D1_I1	GpuF0MMReg\:0x2774	2-38
MC_SEQ_CNTL	GpuF0MMReg\:0x2600	2-2
MC_SEQ_DQ_PAD_CNTL_D0_I0	GpuF0MMReg\:0x267C	2-26
MC_SEQ_DQ_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2680	2-26
MC_SEQ_DQ_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2778	2-38
MC_SEQ_DQ_PAD_CNTL_D1_I1	GpuF0MMReg\:0x277C	2-38
MC_SEQ_DRAM	GpuF0MMReg\:0x2608	2-3
MC_SEQ_GENERAL_CONFIG	GpuF0MMReg\:0x26D8	2-48
MC_SEQ_IO_CTL_D0	GpuF0MMReg\:0x265C	2-23
MC_SEQ_IO_CTL_D1	GpuF0MMReg\:0x2660	2-23
MC_SEQ_IO_CTL_UNUSED	GpuF0MMReg\:0x2898	2-24
MC_SEQ_MISC_TIMING_B	GpuF0MMReg\:0x2624	2-5
MC_SEQ_MISC_TIMING_C	GpuF0MMReg\:0x2644	2-7
MC_SEQ_MISC_TIMING_P	GpuF0MMReg\:0x2614	2-4
MC_SEQ_MISC_TIMING_S	GpuF0MMReg\:0x2634	2-6
MC_SEQ_MISC_TIMING2_B	GpuF0MMReg\:0x2628	2-5
MC_SEQ_MISC_TIMING2_C	GpuF0MMReg\:0x2648	2-8
MC_SEQ_MISC_TIMING2_P	GpuF0MMReg\:0x2618	2-4
MC_SEQ_MISC_TIMING2_S	GpuF0MMReg\:0x2638	2-7
MC_SEQ_NPL_CTL_D0	GpuF0MMReg\:0x2664	2-24
MC_SEQ_NPL_CTL_D1	GpuF0MMReg\:0x2668	2-24
MC_SEQ_QS_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2684	2-27
MC_SEQ_QS_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2688	2-27
MC_SEQ_QS_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2780	2-38
MC_SEQ_QS_PAD_CNTL_D1_I1	GpuF0MMReg\:0x2784	2-39
MC_SEQ_RAS_TIMING_B	GpuF0MMReg\:0x261C	2-5
MC_SEQ_RAS_TIMING_C	GpuF0MMReg\:0x263C	2-7
MC_SEQ_RAS_TIMING_P	GpuF0MMReg\:0x260C	2-4
MC_SEQ_RAS_TIMING_S	GpuF0MMReg\:0x262C	2-6
MC_SEQ_RD_CTL_D0_B	GpuF0MMReg\:0x2694	2-14
MC_SEQ_RD_CTL_D0_C	GpuF0MMReg\:0x26B4	2-20
MC_SEQ_RD_CTL_D0_P	GpuF0MMReg\:0x264C	2-11
MC_SEQ_RD_CTL_D0_S	GpuF0MMReg\:0x26A4	2-17
MC_SEQ_RD_CTL_D1_B	GpuF0MMReg\:0x2698	2-15
MC_SEQ_RD_CTL_D1_C	GpuF0MMReg\:0x26B8	2-21
MC_SEQ_RD_CTL_D1_P	GpuF0MMReg\:0x2650	2-12
MC_SEQ_RD_CTL_D1_S	GpuF0MMReg\:0x26A8	2-18

Table A-8 Memory Controller Registers Sorted by Name (Continued)

Register Name	Address	Page
MC_SEQ_RS_CNTL	*GpuF0MMReg\:0x26DC*	*2-48*
MC_SEQ_STATUS_M	*GpuF0MMReg\:0x26C8*	*2-49*
MC_SEQ_STATUS_S	*GpuF0MMReg\:0x288C*	*2-49*
MC_SEQ_WR_CTL_D0_B	*GpuF0MMReg\:0x269C*	*2-17*
MC_SEQ_WR_CTL_D0_C	*GpuF0MMReg\:0x26BC*	*2-22*
MC_SEQ_WR_CTL_D0_P	*GpuF0MMReg\:0x2654*	*2-14*
MC_SEQ_WR_CTL_D0_S	*GpuF0MMReg\:0x26AC*	*2-19*
MC_SEQ_WR_CTL_D1_B	*GpuF0MMReg\:0x26A0*	*2-17*
MC_SEQ_WR_CTL_D1_C	*GpuF0MMReg\:0x26C0*	*2-23*
MC_SEQ_WR_CTL_D1_P	*GpuF0MMReg\:0x2658*	*2-14*
MC_SEQ_WR_CTL_D1_S	*GpuF0MMReg\:0x26B0*	*2-20*

A.10 Memory Controller Registers Sorted By Address

Table A-9 Memory Controller Registers Sorted by Address

Register Name	Address	Page
MC_CONFIG	GpuF0MMReg\:0x2000	2-2
MC_SEQ_CNTL	GpuF0MMReg\:0x2600	2-2
MC_SEQ_DRAM	GpuF0MMReg\:0x2608	2-3
MC_SEQ_RAS_TIMING_P	GpuF0MMReg\:0x260C	2-4
MC_SEQ_CAS_TIMING_P	GpuF0MMReg\:0x2610	2-4
MC_SEQ_MISC_TIMING_P	GpuF0MMReg\:0x2614	2-4
MC_SEQ_MISC_TIMING2_P	GpuF0MMReg\:0x2618	2-4
MC_SEQ_RAS_TIMING_B	GpuF0MMReg\:0x261C	2-5
MC_SEQ_CAS_TIMING_B	GpuF0MMReg\:0x2620	2-5
MC_SEQ_MISC_TIMING_B	GpuF0MMReg\:0x2624	2-5
MC_SEQ_MISC_TIMING2_B	GpuF0MMReg\:0x2628	2-5
MC_SEQ_RAS_TIMING_S	GpuF0MMReg\:0x262C	2-6
MC_SEQ_CAS_TIMING_S	GpuF0MMReg\:0x2630	2-6
MC_SEQ_MISC_TIMING_S	GpuF0MMReg\:0x2634	2-6
MC_SEQ_MISC_TIMING2_S	GpuF0MMReg\:0x2638	2-7
MC_SEQ_RAS_TIMING_C	GpuF0MMReg\:0x263C	2-7
MC_SEQ_CAS_TIMING_C	GpuF0MMReg\:0x2640	2-7
MC_SEQ_MISC_TIMING_C	GpuF0MMReg\:0x2644	2-7
MC_SEQ_MISC_TIMING2_C	GpuF0MMReg\:0x2648	2-8
MC_SEQ_RD_CTL_D0_P	GpuF0MMReg\:0x264C	2-11
MC_SEQ_RD_CTL_D1_P	GpuF0MMReg\:0x2650	2-12
MC_SEQ_WR_CTL_D0_P	GpuF0MMReg\:0x2654	2-14
MC_SEQ_WR_CTL_D1_P	GpuF0MMReg\:0x2658	2-14
MC_SEQ_IO_CTL_D0	GpuF0MMReg\:0x265C	2-23
MC_SEQ_IO_CTL_D1	GpuF0MMReg\:0x2660	2-23
MC_SEQ_NPL_CTL_D0	GpuF0MMReg\:0x2664	2-24
MC_SEQ_NPL_CTL_D1	GpuF0MMReg\:0x2668	2-24
MC_SEQ_CK_PAD_CNTL_D0_I0	GpuF0MMReg\:0x266C	2-25
MC_SEQ_CK_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2670	2-25
MC_SEQ_CMD_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2674	2-26
MC_SEQ_CMD_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2678	2-26
MC_SEQ_DQ_PAD_CNTL_D0_I0	GpuF0MMReg\:0x267C	2-26
MC_SEQ_DQ_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2680	2-26
MC_SEQ_QS_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2684	2-27
MC_SEQ_QS_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2688	2-27
MC_SEQ_A_PAD_CNTL_D0_I0	GpuF0MMReg\:0x268C	2-27
MC_SEQ_A_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2690	2-27
MC_SEQ_RD_CTL_D0_B	GpuF0MMReg\:0x2694	2-14
MC_SEQ_RD_CTL_D1_B	GpuF0MMReg\:0x2698	2-15
MC_SEQ_WR_CTL_D0_B	GpuF0MMReg\:0x269C	2-17

Table A-9 Memory Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
MC_SEQ_WR_CTL_D1_B	GpuF0MMReg\:0x26A0	2-17
MC_SEQ_RD_CTL_D0_S	GpuF0MMReg\:0x26A4	2-17
MC_SEQ_RD_CTL_D1_S	GpuF0MMReg\:0x26A8	2-18
MC_SEQ_WR_CTL_D0_S	GpuF0MMReg\:0x26AC	2-19
MC_SEQ_WR_CTL_D1_S	GpuF0MMReg\:0x26B0	2-20
MC_SEQ_RD_CTL_D0_C	GpuF0MMReg\:0x26B4	2-20
MC_SEQ_RD_CTL_D1_C	GpuF0MMReg\:0x26B8	2-21
MC_SEQ_WR_CTL_D0_C	GpuF0MMReg\:0x26BC	2-22
MC_SEQ_WR_CTL_D1_C	GpuF0MMReg\:0x26C0	2-23
MC_SEQ_CMD	GpuF0MMReg\:0x26C4	2-8
MC_SEQ_STATUS_M	GpuF0MMReg\:0x26C8	2-49
MC_PMG_CMD	GpuF0MMReg\:0x26CC	2-8
MC_PMG_CFG	GpuF0MMReg\:0x26D0	2-9
MC_IMP_CNTL	GpuF0MMReg\:0x26D4	2-10
MC_SEQ_GENERAL_CONFIG	GpuF0MMReg\:0x26D8	2-48
MC_SEQ_RS_CNTL	GpuF0MMReg\:0x26DC	2-48
MC_IO_RD_STR_NCNTL_B0_D0	GpuF0MMReg\:0x26E8	2-35
MC_IO_RD_STR_NCNTL_B2_D0	GpuF0MMReg\:0x26F8	2-35
MC_IO_PAD_CNTL	GpuF0MMReg\:0x2700	2-11
MC_IO_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2704	2-28
MC_IO_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2708	2-28
MC_IO_RD_DQ_CNTL_D0_I0	GpuF0MMReg\:0x2710	2-29
MC_IO_RD_DQ_CNTL_D0_I1	GpuF0MMReg\:0x2714	2-29
MC_IO_RD_QS_CNTL_D0_I0	GpuF0MMReg\:0x2718	2-30
MC_IO_RD_QS_CNTL_D0_I1	GpuF0MMReg\:0x271C	2-30
MC_IO_RD_QS2_CNTL_D0_I0	GpuF0MMReg\:0x2720	2-30
MC_IO_RD_QS2_CNTL_D0_I1	GpuF0MMReg\:0x2724	2-30
MC_IO_WR_CNTL_D0_I0	GpuF0MMReg\:0x2728	2-30
MC_IO_WR_CNTL_D0_I1	GpuF0MMReg\:0x272C	2-31
MC_IO_CK_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2730	2-31
MC_IO_CK_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2734	2-31
MC_IO_CMD_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2738	2-31
MC_IO_CMD_PAD_CNTL_D0_I1	GpuF0MMReg\:0x273C	2-32
MC_IO_DQ_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2740	2-32
MC_IO_DQ_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2744	2-32
MC_IO_QS_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2748	2-33
MC_IO_QS_PAD_CNTL_D0_I1	GpuF0MMReg\:0x274C	2-33
MC_IO_A_PAD_CNTL_D0_I0	GpuF0MMReg\:0x2750	2-33
MC_IO_A_PAD_CNTL_D0_I1	GpuF0MMReg\:0x2754	2-34
MC_IO_WR_DQ_CNTL_D0_I0	GpuF0MMReg\:0x2758	2-34
MC_IO_WR_DQ_CNTL_D0_I1	GpuF0MMReg\:0x275C	2-34
MC_IO_WR_QS_CNTL_D0_I0	GpuF0MMReg\:0x2760	2-34
MC_IO_WR_QS_CNTL_D0_I1	GpuF0MMReg\:0x2764	2-34

Table A-9 Memory Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
MC_SEQ_CK_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2768	2-37
MC_SEQ_CK_PAD_CNTL_D1_I1	GpuF0MMReg\:0x276C	2-37
MC_SEQ_CMD_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2770	2-37
MC_SEQ_CMD_PAD_CNTL_D1_I1	GpuF0MMReg\:0x2774	2-38
MC_SEQ_DQ_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2778	2-38
MC_SEQ_DQ_PAD_CNTL_D1_I1	GpuF0MMReg\:0x277C	2-38
MC_SEQ_QS_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2780	2-38
MC_SEQ_QS_PAD_CNTL_D1_I1	GpuF0MMReg\:0x2784	2-39
MC_SEQ_A_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2788	2-39
MC_SEQ_A_PAD_CNTL_D1_I1	GpuF0MMReg\:0x278C	2-39
MC_IO_PAD_CNTL_D1_I0	GpuF0MMReg\:0x2790	2-39
MC_IO_PAD_CNTL_D1_I1	GpuF0MMReg\:0x2794	2-40
MC_IO_RD_DQ_CNTL_D1_I0	GpuF0MMReg\:0x2798	2-41
MC_IO_RD_DQ_CNTL_D1_I1	GpuF0MMReg\:0x279C	2-41
MC_IO_RD_QS_CNTL_D1_I0	GpuF0MMReg\:0x27A0	2-41
MC_IO_RD_QS_CNTL_D1_I1	GpuF0MMReg\:0x27A4	2-42
MC_IO_RD_QS2_CNTL_D1_I0	GpuF0MMReg\:0x27A8	2-42
MC_IO_RD_QS2_CNTL_D1_I1	GpuF0MMReg\:0x27AC	2-42
MC_IO_WR_CNTL_D1_I0	GpuF0MMReg\:0x27B0	2-42
MC_IO_WR_CNTL_D1_I1	GpuF0MMReg\:0x27B4	2-42
MC_IO_CK_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27B8	2-43
MC_IO_CK_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27BC	2-43
MC_IO_CMD_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27C0	2-43
MC_IO_CMD_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27C4	2-43
MC_IO_DQ_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27C8	2-44
MC_IO_DQ_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27CC	2-44
MC_IO_QS_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27D0	2-44
MC_IO_QS_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27D4	2-45
MC_IO_A_PAD_CNTL_D1_I0	GpuF0MMReg\:0x27D8	2-45
MC_IO_A_PAD_CNTL_D1_I1	GpuF0MMReg\:0x27DC	2-45
MC_IO_WR_DQ_CNTL_D1_I0	GpuF0MMReg\:0x27E0	2-46
MC_IO_WR_DQ_CNTL_D1_I1	GpuF0MMReg\:0x27E4	2-46
MC_IO_WR_QS_CNTL_D1_I0	GpuF0MMReg\:0x27E8	2-46
MC_IO_WR_QS_CNTL_D1_I1	GpuF0MMReg\:0x27EC	2-46
MC_IO_PAD_CNTL_D0	GpuF0MMReg\:0x27F0	2-25
MC_IO_PAD_CNTL_D1	GpuF0MMReg\:0x27F4	2-25
MC_IO_RD_STR_NCNTL_B3_D0	GpuF0MMReg\:0x27F8	2-35
MC_IO_RD_STR_NCNTL_B4_D0	GpuF0MMReg\:0x2800	2-36
MC_IO_RD_STR_NCNTL_B5_D0	GpuF0MMReg\:0x2808	2-36
MC_IO_RD_STR_NCNTL_B1_D0	GpuF0MMReg\:0x280C	2-35
MC_IO_RD_STR_NCNTL_B6_D0	GpuF0MMReg\:0x2810	2-36
MC_IO_RD_STR_NCNTL_B7_D0	GpuF0MMReg\:0x2818	2-37
MC_IO_RD_STR_NCNTL_B0_D1	GpuF0MMReg\:0x2820	2-46

Table A-9 Memory Controller Registers Sorted by Address (Continued)

Register Name	Address	Page
MC_IO_RD_STR_NCNTL_B1_D1	*GpuF0MMReg\:0x2828*	*2-46*
MC_IO_RD_STR_NCNTL_B2_D1	*GpuF0MMReg\:0x2830*	*2-47*
MC_IO_RD_STR_NCNTL_B3_D1	*GpuF0MMReg\:0x2838*	*2-47*
MC_IO_RD_STR_NCNTL_B4_D1	*GpuF0MMReg\:0x2840*	*2-47*
MC_IO_RD_STR_NCNTL_B5_D1	*GpuF0MMReg\:0x2848*	*2-47*
MC_IO_RD_STR_NCNTL_B6_D1	*GpuF0MMReg\:0x2850*	*2-48*
MC_IO_RD_STR_NCNTL_B7_D1	*GpuF0MMReg\:0x2858*	*2-48*
MC_IMP_STATUS	*GpuF0MMReg\:0x2874*	*2-10*
MC_IMP_DEBUG	*GpuF0MMReg\:0x2878*	*2-10*
MC_NPL_STATUS	*GpuF0MMReg\:0x2888*	*2-50*
MC_SEQ_STATUS_S	*GpuF0MMReg\:0x288C*	*2-49*
MC_SEQ_IO_CTL_UNUSED	*GpuF0MMReg\:0x2898*	*2-24*

A.11 PCIE Registers Sorted By Name

Table A-10 PCIE Registers Sorted by Name

Register Name	Address	Page
PCIE_CI_CNTL	PCIEIND\:0x20	2-54
PCIE_DATA	GpuF0MMReg\:0x34 GpuIOReg\:0x34	2-54
PCIE_ERR_CNTL	PCIEIND_P\:0x6A	2-66
PCIE_INDEX	GpuF0MMReg\:0x30 GpuIOReg\:0x30	2-54
PCIE_LC_CNTL	PCIEIND_P\:0xA0	2-69
PCIE_LC_CNTL2	PCIEIND_P\:0xB1	2-70
PCIE_LC_LINK_WIDTH_CNTL	PCIEIND_P\:0xA2	2-71
PCIE_LC_N_FTS_CNTL	PCIEIND_P\:0xA3	2-71
PCIE_LC_STATE0	PCIEIND_P\:0xA5	2-72
PCIE_LC_STATE1	PCIEIND_P\:0xA6	2-72
PCIE_LC_STATE10	PCIEIND\:0x26	2-56
PCIE_LC_STATE11	PCIEIND\:0x27	2-56
PCIE_LC_STATE2	PCIEIND_P\:0xA7	2-72
PCIE_LC_STATE3	PCIEIND_P\:0xA8	2-72
PCIE_LC_STATE4	PCIEIND_P\:0xA9	2-72
PCIE_LC_STATE5	PCIEIND_P\:0xAA	2-73
PCIE_LC_STATE6	PCIEIND\:0x22	2-55
PCIE_LC_STATE7	PCIEIND\:0x23	2-55
PCIE_LC_STATE8	PCIEIND\:0x24	2-55
PCIE_LC_STATE9	PCIEIND\:0x25	2-55
PCIE_P_BUF_STATUS	PCIEIND\:0x41	2-57
PCIE_P_CNTL	PCIEIND\:0x40	2-56
PCIE_P_DECODE_ERR_CNT_0	PCIEIND\:0xF0	2-63
PCIE_P_DECODE_ERR_CNT_1	PCIEIND\:0xF1	2-63
PCIE_P_DECODE_ERR_CNT_10	PCIEIND\:0xFA	2-64
PCIE_P_DECODE_ERR_CNT_11	PCIEIND\:0xFB	2-64
PCIE_P_DECODE_ERR_CNT_12	PCIEIND\:0xFC	2-65
PCIE_P_DECODE_ERR_CNT_13	PCIEIND\:0xFD	2-65
PCIE_P_DECODE_ERR_CNT_14	PCIEIND\:0xFE	2-65
PCIE_P_DECODE_ERR_CNT_15	PCIEIND\:0xFF	2-65
PCIE_P_DECODE_ERR_CNT_2	PCIEIND\:0xF2	2-63
PCIE_P_DECODE_ERR_CNT_3	PCIEIND\:0xF3	2-63
PCIE_P_DECODE_ERR_CNT_4	PCIEIND\:0xF4	2-63
PCIE_P_DECODE_ERR_CNT_5	PCIEIND\:0xF5	2-63
PCIE_P_DECODE_ERR_CNT_6	PCIEIND\:0xF6	2-64
PCIE_P_DECODE_ERR_CNT_7	PCIEIND\:0xF7	2-64
PCIE_P_DECODE_ERR_CNT_8	PCIEIND\:0xF8	2-64
PCIE_P_DECODE_ERR_CNT_9	PCIEIND\:0xF9	2-64
PCIE_P_DECODE_ERR_CNTL	PCIEIND\:0xEF	2-62

Table A-10 PCIE Registers Sorted by Name (Continued)

Register Name	Address	Page
PCIE_P_DECODER_STATUS	*PCIEIND\:0x42*	*2-58*
PCIE_P_IMP_CNTL_STRENGTH	*PCIEIND\:0x60*	*2-61*
PCIE_P_IMP_CNTL_UPDATE	*PCIEIND\:0x61*	*2-62*
PCIE_P_MISC_DEBUG_STATUS	*PCIEIND\:0x43*	*2-60*
PCIE_P_PAD_MISC_CNTL	*PCIEIND\:0x63*	*2-62*
PCIE_P_STR_CNTL_UPDATE	*PCIEIND\:0x62*	*2-62*
PCIE_P_SYMSYNC_CTL	*PCIEIND\:0x46*	*2-61*
PCIE_RX_CNTL	*PCIEIND_P\:0x70*	*2-67*
PCIE_RX_CREDITS_ALLOCATED_CPL	*PCIEIND_P\:0x82*	*2-68*
PCIE_RX_CREDITS_ALLOCATED_NP	*PCIEIND_P\:0x81*	*2-67*
PCIE_RX_CREDITS_ALLOCATED_P	*PCIEIND_P\:0x80*	*2-67*
PCIE_RX_CREDITS_RECEIVED_CPL	*PCIEIND_P\:0x85*	*2-68*
PCIE_RX_CREDITS_RECEIVED_NP	*PCIEIND_P\:0x84*	*2-68*
PCIE_RX_CREDITS_RECEIVED_P	*PCIEIND_P\:0x83*	*2-68*
PCIE_RX_NUM_NACK	*PCIEIND\:0xE*	*2-54*
PCIE_RX_NUM_NACK_GENERATED	*PCIEIND\:0xF*	*2-54*
PCIE_TX_CNTL	*PCIEIND_P\:0x20*	*2-65*
PCIE_TX_REPLAY	*PCIEIND_P\:0x25*	*2-66*
PCIE_TX_SEQ	*PCIEIND_P\:0x24*	*2-66*

A.12 PCIE Registers Sorted By Address

Table A-11 PCIE Registers Sorted by Address

Register Name	Address	Page
PCIE_INDEX	*GpuF0MMReg\:0x30* *GpuIOReg\:0x30*	*2-54*
PCIE_DATA	*GpuF0MMReg\:0x34* *GpuIOReg\:0x34*	*2-54*
PCIE_CI_CNTL	*PCIEIND\:0x20*	*2-54*
PCIE_LC_STATE6	*PCIEIND\:0x22*	*2-55*
PCIE_LC_STATE7	*PCIEIND\:0x23*	*2-55*
PCIE_LC_STATE8	*PCIEIND\:0x24*	*2-55*
PCIE_LC_STATE9	*PCIEIND\:0x25*	*2-55*
PCIE_LC_STATE10	*PCIEIND\:0x26*	*2-56*
PCIE_LC_STATE11	*PCIEIND\:0x27*	*2-56*
PCIE_P_CNTL	*PCIEIND\:0x40*	*2-56*
PCIE_P_BUF_STATUS	*PCIEIND\:0x41*	*2-57*
PCIE_P_DECODER_STATUS	*PCIEIND\:0x42*	*2-58*
PCIE_P_MISC_DEBUG_STATUS	*PCIEIND\:0x43*	*2-60*
PCIE_P_SYMSYNC_CTL	*PCIEIND\:0x46*	*2-61*
PCIE_P_IMP_CNTL_STRENGTH	*PCIEIND\:0x60*	*2-61*
PCIE_P_IMP_CNTL_UPDATE	*PCIEIND\:0x61*	*2-62*
PCIE_P_STR_CNTL_UPDATE	*PCIEIND\:0x62*	*2-62*
PCIE_P_PAD_MISC_CNTL	*PCIEIND\:0x63*	*2-62*
PCIE_RX_NUM_NACK	*PCIEIND\:0xE*	*2-54*
PCIE_P_DECODE_ERR_CNTL	*PCIEIND\:0xEF*	*2-62*
PCIE_RX_NUM_NACK_GENERATED	*PCIEIND\:0xF*	*2-54*
PCIE_P_DECODE_ERR_CNT_0	*PCIEIND\:0xF0*	*2-63*
PCIE_P_DECODE_ERR_CNT_1	*PCIEIND\:0xF1*	*2-63*
PCIE_P_DECODE_ERR_CNT_2	*PCIEIND\:0xF2*	*2-63*
PCIE_P_DECODE_ERR_CNT_3	*PCIEIND\:0xF3*	*2-63*
PCIE_P_DECODE_ERR_CNT_4	*PCIEIND\:0xF4*	*2-63*
PCIE_P_DECODE_ERR_CNT_5	*PCIEIND\:0xF5*	*2-63*
PCIE_P_DECODE_ERR_CNT_6	*PCIEIND\:0xF6*	*2-64*
PCIE_P_DECODE_ERR_CNT_7	*PCIEIND\:0xF7*	*2-64*
PCIE_P_DECODE_ERR_CNT_8	*PCIEIND\:0xF8*	*2-64*
PCIE_P_DECODE_ERR_CNT_9	*PCIEIND\:0xF9*	*2-64*
PCIE_P_DECODE_ERR_CNT_10	*PCIEIND\:0xFA*	*2-64*
PCIE_P_DECODE_ERR_CNT_11	*PCIEIND\:0xFB*	*2-64*
PCIE_P_DECODE_ERR_CNT_12	*PCIEIND\:0xFC*	*2-65*
PCIE_P_DECODE_ERR_CNT_13	*PCIEIND\:0xFD*	*2-65*
PCIE_P_DECODE_ERR_CNT_14	*PCIEIND\:0xFE*	*2-65*
PCIE_P_DECODE_ERR_CNT_15	*PCIEIND\:0xFF*	*2-65*
PCIE_TX_CNTL	*PCIEIND_P\:0x20*	*2-65*
PCIE_TX_SEQ	*PCIEIND_P\:0x24*	*2-66*

Table A-11 PCIE Registers Sorted by Address (Continued)

Register Name	Address	Page
PCIE_TX_REPLAY	*PCIEIND_P\:0x25*	*2-66*
PCIE_ERR_CNTL	*PCIEIND_P\:0x6A*	*2-66*
PCIE_RX_CNTL	*PCIEIND_P\:0x70*	*2-67*
PCIE_RX_CREDITS_ALLOCATED_P	*PCIEIND_P\:0x80*	*2-67*
PCIE_RX_CREDITS_ALLOCATED_NP	*PCIEIND_P\:0x81*	*2-67*
PCIE_RX_CREDITS_ALLOCATED_CPL	*PCIEIND_P\:0x82*	*2-68*
PCIE_RX_CREDITS_RECEIVED_P	*PCIEIND_P\:0x83*	*2-68*
PCIE_RX_CREDITS_RECEIVED_NP	*PCIEIND_P\:0x84*	*2-68*
PCIE_RX_CREDITS_RECEIVED_CPL	*PCIEIND_P\:0x85*	*2-68*
PCIE_LC_CNTL	*PCIEIND_P\:0xA0*	*2-69*
PCIE_LC_LINK_WIDTH_CNTL	*PCIEIND_P\:0xA2*	*2-71*
PCIE_LC_N_FTS_CNTL	*PCIEIND_P\:0xA3*	*2-71*
PCIE_LC_STATE0	*PCIEIND_P\:0xA5*	*2-72*
PCIE_LC_STATE1	*PCIEIND_P\:0xA6*	*2-72*
PCIE_LC_STATE2	*PCIEIND_P\:0xA7*	*2-72*
PCIE_LC_STATE3	*PCIEIND_P\:0xA8*	*2-72*
PCIE_LC_STATE4	*PCIEIND_P\:0xA9*	*2-72*
PCIE_LC_STATE5	*PCIEIND_P\:0xAA*	*2-73*
PCIE_LC_CNTL2	*PCIEIND_P\:0xB1*	*2-70*

A.13 VIP Registers Sorted By Name

Table A-12 VIP Registers Sorted by Name

Register Name	Address	Page
CAP_INT_CNTL	GpuF0MMReg\:0xB08	2-116
CAP_INT_STATUS	GpuF0MMReg\:0xB0C	2-117
CAP0_ANC_BUF01_BLOCK_CNT	GpuF0MMReg\:0xB74	2-125
CAP0_ANC_BUF23_BLOCK_CNT	GpuF0MMReg\:0xB7C	2-125
CAP0_ANC_H_WINDOW	GpuF0MMReg\:0xB64	2-123
CAP0_ANC0_OFFSET	GpuF0MMReg\:0xB5C	2-123
CAP0_ANC1_OFFSET	GpuF0MMReg\:0xB60	2-123
CAP0_ANC2_OFFSET	GpuF0MMReg\:0xB88	2-125
CAP0_ANC3_OFFSET	GpuF0MMReg\:0xB8C	2-125
CAP0_BUF_PITCH	GpuF0MMReg\:0xB30	2-119
CAP0_BUF_STATUS	GpuF0MMReg\:0xB70	2-124
CAP0_BUF0_EVEN_OFFSET	GpuF0MMReg\:0xB28	2-119
CAP0_BUF0_OFFSET	GpuF0MMReg\:0xB20	2-119
CAP0_BUF1_EVEN_OFFSET	GpuF0MMReg\:0xB2C	2-119
CAP0_BUF1_OFFSET	GpuF0MMReg\:0xB24	2-119
CAP0_CONFIG	GpuF0MMReg\:0xB58	2-122
CAP0_DEBUG	GpuF0MMReg\:0xB54	2-121
CAP0_H_WINDOW	GpuF0MMReg\:0xB38	2-120
CAP0_ONESHOT_BUF_OFFSET	GpuF0MMReg\:0xB6C	2-124
CAP0_PORT_MODE_CNTL	GpuF0MMReg\:0xB4C	2-121
CAP0_TRIG_CNTL	GpuF0MMReg\:0xB50	2-121
CAP0_V_WINDOW	GpuF0MMReg\:0xB34	2-120
CAP0_VBI_H_WINDOW	GpuF0MMReg\:0xB48	2-120
CAP0_VBI_V_WINDOW	GpuF0MMReg\:0xB44	2-120
CAP0_VBI0_OFFSET	GpuF0MMReg\:0xB3C	2-120
CAP0_VBI1_OFFSET	GpuF0MMReg\:0xB40	2-120
CAP0_VBI2_OFFSET	GpuF0MMReg\:0xB80	2-125
CAP0_VBI3_OFFSET	GpuF0MMReg\:0xB84	2-125
CAP0_VIDEO_SYNC_TEST	GpuF0MMReg\:0xB68	2-124
DC_I2C_ARBITRATION	GpuF0MMReg\:0x7D34	2-99
DC_I2C_CONTROL	GpuF0MMReg\:0x7D30	2-98
DC_I2C_DATA	GpuF0MMReg\:0x7D74	2-106
DC_I2C_DDC1_HW_STATUS	GpuF0MMReg\:0x7D40	2-100
DC_I2C_DDC1_SETUP	GpuF0MMReg\:0x7D50	2-102
DC_I2C_DDC1_SPEED	GpuF0MMReg\:0x7D4C	2-101
DC_I2C_DDC2_HW_STATUS	GpuF0MMReg\:0x7D44	2-101
DC_I2C_DDC2_SETUP	GpuF0MMReg\:0x7D58	2-102
DC_I2C_DDC2_SPEED	GpuF0MMReg\:0x7D54	2-102
DC_I2C_DDC3_HW_STATUS	GpuF0MMReg\:0x7D48	2-101
DC_I2C_DDC3_SETUP	GpuF0MMReg\:0x7D60	2-103

Table A-12 VIP Registers Sorted by Name (Continued)

Register Name	Address	Page
DC_I2C_DDC3_SPEED	GpuF0MMReg\:0x7D5C	2-103
DC_I2C_DDC4_HW_STATUS	GpuF0MMReg\:0x7DB0	2-109
DC_I2C_DDC4_SETUP	GpuF0MMReg\:0x7DBC	2-110
DC_I2C_DDC4_SPEED	GpuF0MMReg\:0x7DB4	2-110
DC_I2C_INTERRUPT_CONTROL	GpuF0MMReg\:0x7D38	2-99
DC_I2C_SW_STATUS	GpuF0MMReg\:0x7D3C	2-100
DC_I2C_TRANSACTION0	GpuF0MMReg\:0x7D64	2-104
DC_I2C_TRANSACTION1	GpuF0MMReg\:0x7D68	2-104
DC_I2C_TRANSACTION2	GpuF0MMReg\:0x7D6C	2-105
DC_I2C_TRANSACTION3	GpuF0MMReg\:0x7D70	2-105
DMA_VIP0_TABLE_ADDR	GpuF0MMReg\:0xA20	2-130
DMA_VIP1_TABLE_ADDR	GpuF0MMReg\:0xA30	2-130
DMA_VIP2_TABLE_ADDR	GpuF0MMReg\:0xA40	2-130
DMA_VIP3_TABLE_ADDR	GpuF0MMReg\:0xA50	2-130
DMA_VIPH_ABORT	GpuF0MMReg\:0xA88	2-131
DMA_VIPH_CHUNK_0	GpuF0MMReg\:0xA18	2-129
DMA_VIPH_CHUNK_1_VAL	GpuF0MMReg\:0xA1C	2-129
DMA_VIPH_MISC_CNTL	GpuF0MMReg\:0xA14	2-128
DMA_VIPH_STATUS	GpuF0MMReg\:0xA10	2-128
DMA_VIPH0_ACTIVE	GpuF0MMReg\:0xA24	2-130
DMA_VIPH0_COMMAND	GpuF0MMReg\:0xA00	2-126
DMA_VIPH1_ACTIVE	GpuF0MMReg\:0xA34	2-130
DMA_VIPH1_COMMAND	GpuF0MMReg\:0xA04	2-126
DMA_VIPH2_ACTIVE	GpuF0MMReg\:0xA44	2-131
DMA_VIPH2_COMMAND	GpuF0MMReg\:0xA08	2-127
DMA_VIPH3_ACTIVE	GpuF0MMReg\:0xA54	2-131
DMA_VIPH3_COMMAND	GpuF0MMReg\:0xA0C	2-127
EXTERN_TRIG_CNTL	GpuF0MMReg\:0xE54	2-137
GENERIC_I2C_CONTROL	GpuF0MMReg\:0x7D80	2-106
GENERIC_I2C_DATA	GpuF0MMReg\:0x7D98	2-108
GENERIC_I2C_INTERRUPT_CONTROL	GpuF0MMReg\:0x7D84	2-106
GENERIC_I2C_PIN_DEBUG	GpuF0MMReg\:0x7DA0	2-109
GENERIC_I2C_PIN_SELECTION	GpuF0MMReg\:0x7D9C	2-109
GENERIC_I2C_SETUP	GpuF0MMReg\:0x7D90	2-107
GENERIC_I2C_SPEED	GpuF0MMReg\:0x7D8C	2-107
GENERIC_I2C_STATUS	GpuF0MMReg\:0x7D88	2-107
GENERIC_I2C_TRANSACTION	GpuF0MMReg\:0x7D94	2-108
GPIOPAD_A	GpuF0MMReg\:0x179C	2-132
GPIOPAD_EN	GpuF0MMReg\:0x17A0	2-132
GPIOPAD_EXTERN_TRIG_CNTL	GpuF0MMReg\:0x17C4	2-133
GPIOPAD_MASK	GpuF0MMReg\:0x1798	2-132
GPIOPAD_STRENGTH	GpuF0MMReg\:0x1794	2-132
GPIOPAD_Y	GpuF0MMReg\:0x17A4	2-132

Table A-12 VIP Registers Sorted by Name (Continued)

Register Name	Address	Page
I2C_CNTL_0	*GpuF0MMReg\:0xBC0*	*2-97*
I2C_CNTL_1	*GpuF0MMReg\:0xBC4*	*2-97*
I2C_DATA	*GpuF0MMReg\:0xBC8*	*2-98*
ROM_CNTL	*GpuF0MMReg\:0x1600*	*2-137*
ROM_DATA	*GpuF0MMReg\:0xAC* *GpuIOReg\:0xAC*	*2-138*
ROM_INDEX	*GpuF0MMReg\:0xA8* *GpuIOReg\:0xA8*	*2-138*
ROM_START	*GpuF0MMReg\:0x1614*	*2-138*
ROM_STATUS	*GpuF0MMReg\:0x1608*	*2-137*
VID_BUFFER_CONTROL	*GpuF0MMReg\:0xB00*	*2-116*
VIPH_CH0_ABCNT	*GpuF0MMReg\:0xC30*	*2-113*
VIPH_CH0_ADDR	*GpuF0MMReg\:0xC10*	*2-111*
VIPH_CH0_DATA	*GpuF0MMReg\:0xC00*	*2-111*
VIPH_CH0_SBCNT	*GpuF0MMReg\:0xC20*	*2-112*
VIPH_CH1_ABCNT	*GpuF0MMReg\:0xC34*	*2-113*
VIPH_CH1_ADDR	*GpuF0MMReg\:0xC14*	*2-112*
VIPH_CH1_DATA	*GpuF0MMReg\:0xC04*	*2-111*
VIPH_CH1_SBCNT	*GpuF0MMReg\:0xC24*	*2-112*
VIPH_CH2_ABCNT	*GpuF0MMReg\:0xC38*	*2-113*
VIPH_CH2_ADDR	*GpuF0MMReg\:0xC18*	*2-112*
VIPH_CH2_DATA	*GpuF0MMReg\:0xC08*	*2-111*
VIPH_CH2_SBCNT	*GpuF0MMReg\:0xC28*	*2-113*
VIPH_CH3_ABCNT	*GpuF0MMReg\:0xC3C*	*2-113*
VIPH_CH3_ADDR	*GpuF0MMReg\:0xC1C*	*2-112*
VIPH_CH3_DATA	*GpuF0MMReg\:0xC0C*	*2-111*
VIPH_CH3_SBCNT	*GpuF0MMReg\:0xC2C*	*2-113*
VIPH_CONTROL	*GpuF0MMReg\:0xC40*	*2-114*
VIPH_DMA_CHUNK	*GpuF0MMReg\:0xC48*	*2-115*
VIPH_DV_INT	*GpuF0MMReg\:0xC4C*	*2-115*
VIPH_DV_LAT	*GpuF0MMReg\:0xC44*	*2-114*
VIPH_REG_ADDR	*GpuF0MMReg\:0xC80*	*2-110*
VIPH_REG_DATA	*GpuF0MMReg\:0xC84*	*2-111*
VIPH_TIMEOUT_STAT	*GpuF0MMReg\:0xC50*	*2-115*
VIPPAD_A	*GpuF0MMReg\:0xAC4*	*2-134*
VIPPAD_EN	*GpuF0MMReg\:0xAC8*	*2-135*
VIPPAD_MASK	*GpuF0MMReg\:0xAC0*	*2-133*
VIPPAD_STRENGTH	*GpuF0MMReg\:0xAD0*	*2-136*
VIPPAD_Y	*GpuF0MMReg\:0xACC*	*2-136*

A.14 VIP Registers Sorted By Address

Table A-13 VIP Registers Sorted by Address

Register Name	Address	Page
ROM_CNTL	GpuF0MMReg\:0x1600	2-137
ROM_STATUS	GpuF0MMReg\:0x1608	2-137
ROM_START	GpuF0MMReg\:0x1614	2-138
GPIOPAD_STRENGTH	GpuF0MMReg\:0x1794	2-132
GPIOPAD_MASK	GpuF0MMReg\:0x1798	2-132
GPIOPAD_A	GpuF0MMReg\:0x179C	2-132
GPIOPAD_EN	GpuF0MMReg\:0x17A0	2-132
GPIOPAD_Y	GpuF0MMReg\:0x17A4	2-132
GPIOPAD_EXTERN_TRIG_CNTL	GpuF0MMReg\:0x17C4	2-133
DC_I2C_CONTROL	GpuF0MMReg\:0x7D30	2-98
DC_I2C_ARBITRATION	GpuF0MMReg\:0x7D34	2-99
DC_I2C_INTERRUPT_CONTROL	GpuF0MMReg\:0x7D38	2-99
DC_I2C_SW_STATUS	GpuF0MMReg\:0x7D3C	2-100
DC_I2C_DDC1_HW_STATUS	GpuF0MMReg\:0x7D40	2-100
DC_I2C_DDC2_HW_STATUS	GpuF0MMReg\:0x7D44	2-101
DC_I2C_DDC3_HW_STATUS	GpuF0MMReg\:0x7D48	2-101
DC_I2C_DDC1_SPEED	GpuF0MMReg\:0x7D4C	2-101
DC_I2C_DDC1_SETUP	GpuF0MMReg\:0x7D50	2-102
DC_I2C_DDC2_SPEED	GpuF0MMReg\:0x7D54	2-102
DC_I2C_DDC2_SETUP	GpuF0MMReg\:0x7D58	2-102
DC_I2C_DDC3_SPEED	GpuF0MMReg\:0x7D5C	2-103
DC_I2C_DDC3_SETUP	GpuF0MMReg\:0x7D60	2-103
DC_I2C_TRANSACTION0	GpuF0MMReg\:0x7D64	2-104
DC_I2C_TRANSACTION1	GpuF0MMReg\:0x7D68	2-104
DC_I2C_TRANSACTION2	GpuF0MMReg\:0x7D6C	2-105
DC_I2C_TRANSACTION3	GpuF0MMReg\:0x7D70	2-105
DC_I2C_DATA	GpuF0MMReg\:0x7D74	2-106
GENERIC_I2C_CONTROL	GpuF0MMReg\:0x7D80	2-106
GENERIC_I2C_INTERRUPT_CONTROL	GpuF0MMReg\:0x7D84	2-106
GENERIC_I2C_STATUS	GpuF0MMReg\:0x7D88	2-107
GENERIC_I2C_SPEED	GpuF0MMReg\:0x7D8C	2-107
GENERIC_I2C_SETUP	GpuF0MMReg\:0x7D90	2-107
GENERIC_I2C_TRANSACTION	GpuF0MMReg\:0x7D94	2-108
GENERIC_I2C_DATA	GpuF0MMReg\:0x7D98	2-108
GENERIC_I2C_PIN_SELECTION	GpuF0MMReg\:0x7D9C	2-109
GENERIC_I2C_PIN_DEBUG	GpuF0MMReg\:0x7DA0	2-109
DC_I2C_DDC4_HW_STATUS	GpuF0MMReg\:0x7DB0	2-109
DC_I2C_DDC4_SPEED	GpuF0MMReg\:0x7DB4	2-110
DC_I2C_DDC4_SETUP	GpuF0MMReg\:0x7DBC	2-110
DMA_VIPH0_COMMAND	GpuF0MMReg\:0xA00	2-126
DMA_VIPH1_COMMAND	GpuF0MMReg\:0xA04	2-126

Table A-13 VIP Registers Sorted by Address (Continued)

Register Name	Address	Page
DMA_VIPH2_COMMAND	GpuF0MMReg\:0xA08	2-127
DMA_VIPH3_COMMAND	GpuF0MMReg\:0xA0C	2-127
DMA_VIPH_STATUS	GpuF0MMReg\:0xA10	2-128
DMA_VIPH_MISC_CNTL	GpuF0MMReg\:0xA14	2-128
DMA_VIPH_CHUNK_0	GpuF0MMReg\:0xA18	2-129
DMA_VIPH_CHUNK_1_VAL	GpuF0MMReg\:0xA1C	2-129
DMA_VIP0_TABLE_ADDR	GpuF0MMReg\:0xA20	2-130
DMA_VIPH0_ACTIVE	GpuF0MMReg\:0xA24	2-130
DMA_VIP1_TABLE_ADDR	GpuF0MMReg\:0xA30	2-130
DMA_VIPH1_ACTIVE	GpuF0MMReg\:0xA34	2-130
DMA_VIP2_TABLE_ADDR	GpuF0MMReg\:0xA40	2-130
DMA_VIPH2_ACTIVE	GpuF0MMReg\:0xA44	2-131
DMA_VIP3_TABLE_ADDR	GpuF0MMReg\:0xA50	2-130
DMA_VIPH3_ACTIVE	GpuF0MMReg\:0xA54	2-131
DMA_VIPH_ABORT	GpuF0MMReg\:0xA88	2-131
ROM_INDEX	GpuF0MMReg\:0xA8 GpuIOReg\:0xA8	2-138
VIPPAD_MASK	GpuF0MMReg\:0xAC0	2-133
VIPPAD_A	GpuF0MMReg\:0xAC4	2-134
VIPPAD_EN	GpuF0MMReg\:0xAC8	2-135
VIPPAD_Y	GpuF0MMReg\:0xACC	2-136
ROM_DATA	GpuF0MMReg\:0xAC GpuIOReg\:0xAC	2-138
VIPPAD_STRENGTH	GpuF0MMReg\:0xAD0	2-136
VID_BUFFER_CONTROL	GpuF0MMReg\:0xB00	2-116
CAP_INT_CNTL	GpuF0MMReg\:0xB08	2-116
CAP_INT_STATUS	GpuF0MMReg\:0xB0C	2-117
CAP0_BUF0_OFFSET	GpuF0MMReg\:0xB20	2-119
CAP0_BUF1_OFFSET	GpuF0MMReg\:0xB24	2-119
CAP0_BUF0_EVEN_OFFSET	GpuF0MMReg\:0xB28	2-119
CAP0_BUF1_EVEN_OFFSET	GpuF0MMReg\:0xB2C	2-119
CAP0_BUF_PITCH	GpuF0MMReg\:0xB30	2-119
CAP0_V_WINDOW	GpuF0MMReg\:0xB34	2-120
CAP0_H_WINDOW	GpuF0MMReg\:0xB38	2-120
CAP0_VBI0_OFFSET	GpuF0MMReg\:0xB3C	2-120
CAP0_VBI1_OFFSET	GpuF0MMReg\:0xB40	2-120
CAP0_VBI_V_WINDOW	GpuF0MMReg\:0xB44	2-120
CAP0_VBI_H_WINDOW	GpuF0MMReg\:0xB48	2-120
CAP0_PORT_MODE_CNTL	GpuF0MMReg\:0xB4C	2-121
CAP0_TRIG_CNTL	GpuF0MMReg\:0xB50	2-121
CAP0_DEBUG	GpuF0MMReg\:0xB54	2-121
CAP0_CONFIG	GpuF0MMReg\:0xB58	2-122
CAP0_ANC0_OFFSET	GpuF0MMReg\:0xB5C	2-123

Table A-13 VIP Registers Sorted by Address (Continued)

Register Name	Address	Page
CAP0_ANC1_OFFSET	GpuF0MMReg\:0xB60	2-123
CAP0_ANC_H_WINDOW	GpuF0MMReg\:0xB64	2-123
CAP0_VIDEO_SYNC_TEST	GpuF0MMReg\:0xB68	2-124
CAP0_ONESHOT_BUF_OFFSET	GpuF0MMReg\:0xB6C	2-124
CAP0_BUF_STATUS	GpuF0MMReg\:0xB70	2-124
CAP0_ANC_BUF01_BLOCK_CNT	GpuF0MMReg\:0xB74	2-125
CAP0_ANC_BUF23_BLOCK_CNT	GpuF0MMReg\:0xB7C	2-125
CAP0_VBI2_OFFSET	GpuF0MMReg\:0xB80	2-125
CAP0_VBI3_OFFSET	GpuF0MMReg\:0xB84	2-125
CAP0_ANC2_OFFSET	GpuF0MMReg\:0xB88	2-125
CAP0_ANC3_OFFSET	GpuF0MMReg\:0xB8C	2-125
I2C_CNTL_0	GpuF0MMReg\:0xBC0	2-97
I2C_CNTL_1	GpuF0MMReg\:0xBC4	2-97
I2C_DATA	GpuF0MMReg\:0xBC8	2-98
VIPH_CH0_DATA	GpuF0MMReg\:0xC00	2-111
VIPH_CH1_DATA	GpuF0MMReg\:0xC04	2-111
VIPH_CH2_DATA	GpuF0MMReg\:0xC08	2-111
VIPH_CH3_DATA	GpuF0MMReg\:0xC0C	2-111
VIPH_CH0_ADDR	GpuF0MMReg\:0xC10	2-111
VIPH_CH1_ADDR	GpuF0MMReg\:0xC14	2-112
VIPH_CH2_ADDR	GpuF0MMReg\:0xC18	2-112
VIPH_CH3_ADDR	GpuF0MMReg\:0xC1C	2-112
VIPH_CH0_SBCNT	GpuF0MMReg\:0xC20	2-112
VIPH_CH1_SBCNT	GpuF0MMReg\:0xC24	2-112
VIPH_CH2_SBCNT	GpuF0MMReg\:0xC28	2-113
VIPH_CH3_SBCNT	GpuF0MMReg\:0xC2C	2-113
VIPH_CH0_ABCNT	GpuF0MMReg\:0xC30	2-113
VIPH_CH1_ABCNT	GpuF0MMReg\:0xC34	2-113
VIPH_CH2_ABCNT	GpuF0MMReg\:0xC38	2-113
VIPH_CH3_ABCNT	GpuF0MMReg\:0xC3C	2-113
VIPH_CONTROL	GpuF0MMReg\:0xC40	2-114
VIPH_DV_LAT	GpuF0MMReg\:0xC44	2-114
VIPH_DMA_CHUNK	GpuF0MMReg\:0xC48	2-115
VIPH_DV_INT	GpuF0MMReg\:0xC4C	2-115
VIPH_TIMEOUT_STAT	GpuF0MMReg\:0xC50	2-115
VIPH_REG_ADDR	GpuF0MMReg\:0xC80	2-110
VIPH_REG_DATA	GpuF0MMReg\:0xC84	2-111
EXTERN_TRIG_CNTL	GpuF0MMReg\:0xE54	2-137

A.15 VGA ATTR Registers Sorted By Name

Table A-14 VGA ATTR Registers Sorted by Name

Register Name	Address	Page
ATTR00	*VGAATTRIND\:0x0*	*2-154*
ATTR01	*VGAATTRIND\:0x1*	*2-154*
ATTR02	*VGAATTRIND\:0x2*	*2-155*
ATTR03	*VGAATTRIND\:0x3*	*2-155*
ATTR04	*VGAATTRIND\:0x4*	*2-155*
ATTR05	*VGAATTRIND\:0x5*	*2-155*
ATTR06	*VGAATTRIND\:0x6*	*2-155*
ATTR07	*VGAATTRIND\:0x7*	*2-156*
ATTR08	*VGAATTRIND\:0x8*	*2-156*
ATTR09	*VGAATTRIND\:0x9*	*2-156*
ATTR0A	*VGAATTRIND\:0xA*	*2-156*
ATTR0B	*VGAATTRIND\:0xB*	*2-156*
ATTR0C	*VGAATTRIND\:0xC*	*2-157*
ATTR0D	*VGAATTRIND\:0xD*	*2-157*
ATTR0E	*VGAATTRIND\:0xE*	*2-157*
ATTR0F	*VGAATTRIND\:0xF*	*2-157*
ATTR10	*VGAATTRIND\:0x10*	*2-157*
ATTR11	*VGAATTRIND\:0x11*	*2-158*
ATTR12	*VGAATTRIND\:0x12*	*2-158*
ATTR13	*VGAATTRIND\:0x13*	*2-159*
ATTR14	*VGAATTRIND\:0x14*	*2-159*

A.16 VGA CRT Registers Sorted By Name

Table A-15 VGA CRT Registers Sorted by Name

Register Name	Address	Page
CRT00	*VGACRTIND\:0x0*	*2-144*
CRT01	*VGACRTIND\:0x1*	*2-144*
CRT02	*VGACRTIND\:0x2*	*2-144*
CRT03	*VGACRTIND\:0x3*	*2-144*
CRT04	*VGACRTIND\:0x4*	*2-145*
CRT05	*VGACRTIND\:0x5*	*2-145*
CRT06	*VGACRTIND\:0x6*	*2-145*
CRT07	*VGACRTIND\:0x7*	*2-145*
CRT08	*VGACRTIND\:0x8*	*2-146*
CRT09	*VGACRTIND\:0x9*	*2-146*
CRT0A	*VGACRTIND\:0xA*	*2-147*
CRT0B	*VGACRTIND\:0xB*	*2-147*
CRT0C	*VGACRTIND\:0xC*	*2-147*
CRT0D	*VGACRTIND\:0xD*	*2-147*
CRT0E	*VGACRTIND\:0xE*	*2-148*
CRT0F	*VGACRTIND\:0xF*	*2-148*
CRT10	*VGACRTIND\:0x10*	*2-148*
CRT11	*VGACRTIND\:0x11*	*2-148*
CRT12	*VGACRTIND\:0x12*	*2-149*
CRT13	*VGACRTIND\:0x13*	*2-149*
CRT14	*VGACRTIND\:0x14*	*2-149*
CRT15	*VGACRTIND\:0x15*	*2-150*
CRT16	*VGACRTIND\:0x16*	*2-150*
CRT17	*VGACRTIND\:0x17*	*2-150*
CRT18	*VGACRTIND\:0x18*	*2-150*
CRT1E	*VGACRTIND\:0x1E*	*2-151*
CRT1F	*VGACRTIND\:0x1F*	*2-151*
CRT22	*VGACRTIND\:0x22*	*2-151*

A.17 VGA GRPH Registers Sorted By Name

Table A-16 VGA GRPH Registers Sorted by Name

Register Name	Address	Page
GRA00	*VGAGRPHIND\:0x0*	*2-151*
GRA01	*VGAGRPHIND\:0x1*	*2-152*
GRA02	*VGAGRPHIND\:0x2*	*2-152*
GRA03	*VGAGRPHIND\:0x3*	*2-152*
GRA04	*VGAGRPHIND\:0x4*	*2-152*
GRA05	*VGAGRPHIND\:0x5*	*2-152*
GRA06	*VGAGRPHIND\:0x6*	*2-153*
GRA07	*VGAGRPHIND\:0x7*	*2-153*
GRA08	*VGAGRPHIND\:0x8*	*2-153*

A.18 VGA SEQ Registers Sorted By Name

Table A-17 VGA SEQ Registers Sorted by Name

Register Name	Address	Page
SEQ00	*VGASEQIND\:0x0*	*2-142*
SEQ01	*VGASEQIND\:0x1*	*2-142*
SEQ02	*VGASEQIND\:0x2*	*2-143*
SEQ03	*VGASEQIND\:0x3*	*2-143*
SEQ04	*VGASEQIND\:0x4*	*2-143*

A.19 All Registers Sorted by Name

Table A-18 All Registers Sorted by Name

Register Name	Page
ADAPTER_ID	*2-77*
ADAPTER_ID_W	*2-77*
ATTR00	*2-154*
ATTR01	*2-154*
ATTR02	*2-155*
ATTR03	*2-155*
ATTR04	*2-155*
ATTR05	*2-155*
ATTR06	*2-155*
ATTR07	*2-156*
ATTR08	*2-156*
ATTR09	*2-156*
ATTR0A	*2-156*
ATTR0B	*2-156*
ATTR0C	*2-157*
ATTR0D	*2-157*
ATTR0E	*2-157*
ATTR0F	*2-157*
ATTR10	*2-157*
ATTR11	*2-158*
ATTR12	*2-158*
ATTR13	*2-159*
ATTR14	*2-159*
ATTRDR	*2-154*
ATTRDW	*2-154*
ATTRX	*2-154*
BASE_CLASS	*2-75*
BIST	*2-76*
BUS_CNTL	*2-51*
CACHE_LINE	*2-76*
CAP_INT_CNTL	*2-116*
CAP_INT_STATUS	*2-117*
CAP_PTR	*2-76*
CAP0_ANC_BUF01_BLOCK_CNT	*2-125*
CAP0_ANC_BUF23_BLOCK_CNT	*2-125*
CAP0_ANC_H_WINDOW	*2-123*
CAP0_ANC0_OFFSET	*2-123*
CAP0_ANC1_OFFSET	*2-123*
CAP0_ANC2_OFFSET	*2-125*
CAP0_ANC3_OFFSET	*2-125*

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
CAP0_BUF_PITCH	2-119
CAP0_BUF_STATUS	2-124
CAP0_BUF0_EVEN_OFFSET	2-119
CAP0_BUF0_OFFSET	2-119
CAP0_BUF1_EVEN_OFFSET	2-119
CAP0_BUF1_OFFSET	2-119
CAP0_CONFIG	2-122
CAP0_DEBUG	2-121
CAP0_H_WINDOW	2-120
CAP0_ONESHOT_BUF_OFFSET	2-124
CAP0_PORT_MODE_CNTL	2-121
CAP0_TRIG_CNTL	2-121
CAP0_V_WINDOW	2-120
CAP0_VBI_H_WINDOW	2-120
CAP0_VBI_V_WINDOW	2-120
CAP0_VBI0_OFFSET	2-120
CAP0_VBI1_OFFSET	2-120
CAP0_VBI2_OFFSET	2-125
CAP0_VBI3_OFFSET	2-125
CAP0_VIDEO_SYNC_TEST	2-124
CAPTURE_START_STATUS	2-347
CG_CLKPIN_CNTL	2-94
CG_MISC_REG	2-95
CG_MPLL_SPREAD_SPECTRUM	2-96
CG_SPLL_SPREAD_SPECTRUM_CTXSW	2-96
CG_SPLL_SPREAD_SPECTRUM_HIGH	2-95
CG_SPLL_SPREAD_SPECTRUM_LOW	2-95
CG_SPLL_SPREAD_SPECTRUM_MED	2-95
CG_TC_JTAG_0	2-94
CG_TC_JTAG_1	2-95
CG_UPLL_SPREAD_SPECTRUM	2-96
COMMAND	2-73
CONFIG_APER_SIZE	2-53
CONFIG_CNTL	2-52
CONFIG_F0_BASE	2-53
CONFIG_MEMSIZE	2-52
CONFIG_REG_APER_SIZE	2-53
CRT00	2-144
CRT01	2-144
CRT02	2-144
CRT03	2-144
CRT04	2-145
CRT05	2-145

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
CRT06	2-145
CRT07	2-145
CRT08	2-146
CRT09	2-146
CRT0A	2-147
CRT0B	2-147
CRT0C	2-147
CRT0D	2-147
CRT0E	2-148
CRT0F	2-148
CRT10	2-148
CRT11	2-148
CRT12	2-149
CRT13	2-149
CRT14	2-149
CRT15	2-150
CRT16	2-150
CRT17	2-150
CRT18	2-150
CRT1E	2-151
CRT1F	2-151
CRT22	2-151
CRTC8_DATA	2-144
CRTC8_IDX	2-143
D1_MVP_AFR_FLIP_FIFO_CNTL	2-207
D1_MVP_AFR_FLIP_MODE	2-207
D1_MVP_FLIP_LINE_NUM_INSERT	2-208
D1COLOR_MATRIX_COEF_1_1	2-193
D1COLOR_MATRIX_COEF_1_2	2-193
D1COLOR_MATRIX_COEF_1_3	2-193
D1COLOR_MATRIX_COEF_1_4	2-193
D1COLOR_MATRIX_COEF_2_1	2-194
D1COLOR_MATRIX_COEF_2_2	2-194
D1COLOR_MATRIX_COEF_2_3	2-194
D1COLOR_MATRIX_COEF_2_4	2-194
D1COLOR_MATRIX_COEF_3_1	2-195
D1COLOR_MATRIX_COEF_3_2	2-195
D1COLOR_MATRIX_COEF_3_3	2-195
D1COLOR_MATRIX_COEF_3_4	2-195
D1COLOR_SPACE_CONVERT	2-196
D1CRTC_BLACK_COLOR	2-265
D1CRTC_BLANK_CONTROL	2-263

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D1CRTC_BLANK_DATA_COLOR	2-264
D1CRTC_CONTROL	2-263
D1CRTC_COUNT_CONTROL	2-266
D1CRTC_COUNT_RESET	2-266
D1CRTC_DOUBLE_BUFFER_CONTROL	2-270
D1CRTC_FLOW_CONTROL	2-261
D1CRTC_FORCE_COUNT_NOW_CNTL	2-261
D1CRTC_H_BLANK_START_END	2-255
D1CRTC_H_SYNC_A	2-256
D1CRTC_H_SYNC_A_CNTL	2-256
D1CRTC_H_SYNC_B	2-256
D1CRTC_H_SYNC_B_CNTL	2-257
D1CRTC_H_TOTAL	2-255
D1CRTC_INTERLACE_CONTROL	2-264
D1CRTC_INTERLACE_STATUS	2-264
D1CRTC_INTERRUPT_CONTROL	2-269
D1CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE	2-266
D1CRTC_MVP_BLACK_KEYER	2-205
D1CRTC_MVP_CONTROL1	2-202
D1CRTC_MVP_CONTROL2	2-203
D1CRTC_MVP_CONTROL3	2-206
D1CRTC_MVP_CRC_CNTL	2-205
D1CRTC_MVP_CRC_RESULT	2-206
D1CRTC_MVP_CRC2_CNTL	2-206
D1CRTC_MVP_CRC2_RESULT	2-206
D1CRTC_MVP_FIFO_CONTROL	2-203
D1CRTC_MVP_FIFO_STATUS	2-203
D1CRTC_MVP_INBAND_CNTL_CAP	2-204
D1CRTC_MVP_INBAND_CNTL_INSERT	2-204
D1CRTC_MVP_INBAND_CNTL_INSERT_TIMER	2-204
D1CRTC_MVP_RECEIVE_CNT_CNTL1	2-207
D1CRTC_MVP_RECEIVE_CNT_CNTL2	2-207
D1CRTC_MVP_SLAVE_STATUS	2-204
D1CRTC_MVP_STATUS	2-205
D1CRTC_OVERSCAN_COLOR	2-264
D1CRTC_PIXEL_DATA_READBACK	2-262
D1CRTC_SNAPSHOT_CONTROL	2-268
D1CRTC_SNAPSHOT_FRAME	2-268
D1CRTC_SNAPSHOT_POSITION	2-268
D1CRTC_SNAPSHOT_STATUS	2-268
D1CRTC_START_LINE_CONTROL	2-269
D1CRTC_STATUS	2-265
D1CRTC_STATUS_FRAME_COUNT	2-266

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D1CRTC_STATUS_HV_COUNT	2-266
D1CRTC_STATUS_POSITION	2-265
D1CRTC_STATUS_VF_COUNT	2-266
D1CRTC_STEREO_CONTROL	2-267
D1CRTC_STEREO_FORCE_NEXT_EYE	2-262
D1CRTC_STEREO_STATUS	2-267
D1CRTC_TRIGA_CNTL	2-258
D1CRTC_TRIGA_MANUAL_TRIG	2-260
D1CRTC_TRIGB_CNTL	2-260
D1CRTC_TRIGB_MANUAL_TRIG	2-261
D1CRTC_UPDATE_LOCK	2-270
D1CRTC_V_BLANK_START_END	2-257
D1CRTC_V_SYNC_A	2-257
D1CRTC_V_SYNC_A_CNTL	2-258
D1CRTC_V_SYNC_B	2-258
D1CRTC_V_SYNC_B_CNTL	2-258
D1CRTC_V_TOTAL	2-257
D1CRTC_VERT_SYNC_CONTROL	2-267
D1CRTC_VGA_PARAMETER_CAPTURE_MODE	2-271
D1CUR_COLOR1	2-199
D1CUR_COLOR2	2-199
D1CUR_CONTROL	2-198
D1CUR_HOT_SPOT	2-199
D1CUR_POSITION	2-199
D1CUR_SIZE	2-198
D1CUR_SURFACE_ADDRESS	2-198
D1CUR_UPDATE	2-199
D1GRPH_ALPHA	2-189
D1GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL	2-192
D1GRPH_CONTROL	2-170
D1GRPH_ENABLE	2-170
D1GRPH_FLIP_CONTROL	2-176
D1GRPH_KEY_RANGE_ALPHA	2-191
D1GRPH_KEY_RANGE_BLUE	2-191
D1GRPH_KEY_RANGE_GREEN	2-190
D1GRPH_KEY_RANGE_RED	2-190
D1GRPH_LUT_SEL	2-171
D1GRPH_PITCH	2-173
D1GRPH_PRIMARY_SURFACE_ADDRESS	2-172
D1GRPH_SECONDARY_SURFACE_ADDRESS	2-173
D1GRPH_SURFACE_ADDRESS_INUSE	2-176
D1GRPH_SURFACE_OFFSET_X	2-173

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D1GRPH_SURFACE_OFFSET_Y	2-173
D1GRPH_SWAP_CNTL	2-172
D1GRPH_UPDATE	2-174
D1GRPH_X_END	2-174
D1GRPH_X_START	2-174
D1GRPH_Y_END	2-174
D1GRPH_Y_START	2-174
D1ICON_COLOR1	2-201
D1ICON_COLOR2	2-201
D1ICON_CONTROL	2-200
D1ICON_SIZE	2-201
D1ICON_START_POSITION	2-201
D1ICON_SURFACE_ADDRESS	2-200
D1ICON_UPDATE	2-202
D1MODE_MASTER_UPDATE_LOCK	2-269
D1MODE_MASTER_UPDATE_MODE	2-270
D1OVL_ALPHA	2-189
D1OVL_ALPHA_CONTROL	2-189
D1OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	2-181
D1OVL_CONTROL1	2-176
D1OVL_CONTROL2	2-177
D1OVL_DFQ_CONTROL	2-180
D1OVL_DFQ_STATUS	2-181
D1OVL_ENABLE	2-176
D1OVL_END	2-179
D1OVL_KEY_ALPHA	2-192
D1OVL_KEY_CONTROL	2-188
D1OVL_KEY_RANGE_BLUE_CB	2-192
D1OVL_KEY_RANGE_GREEN_Y	2-191
D1OVL_KEY_RANGE_RED_CR	2-191
D1OVL_MATRIX_COEF_1_1	2-181
D1OVL_MATRIX_COEF_1_2	2-182
D1OVL_MATRIX_COEF_1_3	2-182
D1OVL_MATRIX_COEF_1_4	2-182
D1OVL_MATRIX_COEF_2_1	2-182
D1OVL_MATRIX_COEF_2_2	2-183
D1OVL_MATRIX_COEF_2_3	2-183
D1OVL_MATRIX_COEF_2_4	2-183
D1OVL_MATRIX_COEF_3_1	2-183
D1OVL_MATRIX_COEF_3_2	2-184
D1OVL_MATRIX_COEF_3_3	2-184
D1OVL_MATRIX_COEF_3_4	2-184
D1OVL_MATRIX_TRANSFORM_EN	2-181

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D1OVL_PITCH	2-178
D1OVL_PWL_0TOF	2-185
D1OVL_PWL_100TO13F	2-186
D1OVL_PWL_10TO1F	2-185
D1OVL_PWL_140TO17F	2-186
D1OVL_PWL_180TO1BF	2-186
D1OVL_PWL_1C0TO1FF	2-187
D1OVL_PWL_200TO23F	2-187
D1OVL_PWL_20TO3F	2-185
D1OVL_PWL_240TO27F	2-187
D1OVL_PWL_280TO2BF	2-187
D1OVL_PWL_2C0TO2FF	2-187
D1OVL_PWL_300TO33F	2-188
D1OVL_PWL_340TO37F	2-188
D1OVL_PWL_380TO3BF	2-188
D1OVL_PWL_3C0TO3FF	2-188
D1OVL_PWL_40TO7F	2-185
D1OVL_PWL_80TOBF	2-185
D1OVL_PWL_C0TOFF	2-186
D1OVL_PWL_TRANSFORM_EN	2-184
D1OVL_RT_BAND_POSITION	2-196
D1OVL_RT_PROCEED_COND	2-197
D1OVL_RT_SKEWCOMMAND	2-196
D1OVL_RT_SKEWCONTROL	2-196
D1OVL_RT_STAT	2-197
D1OVL_START	2-179
D1OVL_SURFACE_ADDRESS	2-178
D1OVL_SURFACE_ADDRESS_INUSE	2-180
D1OVL_SURFACE_OFFSET_X	2-179
D1OVL_SURFACE_OFFSET_Y	2-179
D1OVL_SWAP_CNTL	2-178
D1OVL_UPDATE	2-179
D1VGA_CONTROL	2-163
D2_MVP_AFR_FLIP_FIFO_CNTL	2-241
D2_MVP_AFR_FLIP_MODE	2-241
D2_MVP_FLIP_LINE_NUM_INSERT	2-241
D2COLOR_MATRIX_COEF_1_1	2-231
D2COLOR_MATRIX_COEF_1_2	2-231
D2COLOR_MATRIX_COEF_1_3	2-231
D2COLOR_MATRIX_COEF_1_4	2-231
D2COLOR_MATRIX_COEF_2_1	2-232
D2COLOR_MATRIX_COEF_2_2	2-232

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D2COLOR_MATRIX_COEF_2_3	2-232
D2COLOR_MATRIX_COEF_2_4	2-232
D2COLOR_MATRIX_COEF_3_1	2-232
D2COLOR_MATRIX_COEF_3_2	2-233
D2COLOR_MATRIX_COEF_3_3	2-233
D2COLOR_MATRIX_COEF_3_4	2-233
D2COLOR_SPACE_CONVERT	2-233
D2CRTC_BLACK_COLOR	2-281
D2CRTC_BLANK_CONTROL	2-279
D2CRTC_BLANK_DATA_COLOR	2-280
D2CRTC_CONTROL	2-279
D2CRTC_COUNT_CONTROL	2-282
D2CRTC_COUNT_RESET	2-282
D2CRTC_DOUBLE_BUFFER_CONTROL	2-286
D2CRTC_FLOW_CONTROL	2-277
D2CRTC_FORCE_COUNT_NOW_CNTL	2-277
D2CRTC_H_BLANK_START_END	2-271
D2CRTC_H_SYNC_A	2-271
D2CRTC_H_SYNC_A_CNTL	2-272
D2CRTC_H_SYNC_B	2-272
D2CRTC_H_SYNC_B_CNTL	2-272
D2CRTC_H_TOTAL	2-271
D2CRTC_INTERLACE_CONTROL	2-280
D2CRTC_INTERLACE_STATUS	2-280
D2CRTC_INTERRUPT_CONTROL	2-285
D2CRTC_MANUAL_FORCE_VSYNC_NEXT_LINE	2-282
D2CRTC_MVP_INBAND_CNTL_INSERT	2-205
D2CRTC_MVP_INBAND_CNTL_INSERT_TIMER	2-205
D2CRTC_MVP_STATUS	2-240
D2CRTC_OVERSCAN_COLOR	2-280
D2CRTC_PIXEL_DATA_READBACK	2-278
D2CRTC_SNAPSHOT_CONTROL	2-284
D2CRTC_SNAPSHOT_FRAME	2-284
D2CRTC_SNAPSHOT_POSITION	2-284
D2CRTC_SNAPSHOT_STATUS	2-284
D2CRTC_START_LINE_CONTROL	2-285
D2CRTC_STATUS	2-281
D2CRTC_STATUS_FRAME_COUNT	2-282
D2CRTC_STATUS_HV_COUNT	2-282
D2CRTC_STATUS_POSITION	2-281
D2CRTC_STATUS_VF_COUNT	2-282
D2CRTC_STEREO_CONTROL	2-283
D2CRTC_STEREO_FORCE_NEXT_EYE	2-278

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D2CRTC_STEREO_STATUS	2-283
D2CRTC_TRIGA_CNTL	2-274
D2CRTC_TRIGA_MANUAL_TRIG	2-276
D2CRTC_TRIGB_CNTL	2-276
D2CRTC_TRIGB_MANUAL_TRIG	2-277
D2CRTC_UPDATE_LOCK	2-286
D2CRTC_V_BLANK_START_END	2-273
D2CRTC_V_SYNC_A	2-273
D2CRTC_V_SYNC_A_CNTL	2-274
D2CRTC_V_SYNC_B	2-274
D2CRTC_V_SYNC_B_CNTL	2-274
D2CRTC_V_TOTAL	2-273
D2CRTC_VBI_END	2-273
D2CRTC_VERT_SYNC_CONTROL	2-283
D2CRTC_VGA_PARAMETER_CAPTURE_MODE	2-287
D2CUR_COLOR1	2-237
D2CUR_COLOR2	2-237
D2CUR_CONTROL	2-235
D2CUR_HOT_SPOT	2-237
D2CUR_POSITION	2-236
D2CUR_SIZE	2-236
D2CUR_SURFACE_ADDRESS	2-236
D2CUR_UPDATE	2-237
D2GRPH_ALPHA	2-227
D2GRPH_COLOR_MATRIX_TRANSFORMATION_CNTL	2-230
D2GRPH_CONTROL	2-208
D2GRPH_ENABLE	2-208
D2GRPH_FLIP_CONTROL	2-214
D2GRPH_KEY_RANGE_ALPHA	2-229
D2GRPH_KEY_RANGE_BLUE	2-229
D2GRPH_KEY_RANGE_GREEN	2-228
D2GRPH_KEY_RANGE_RED	2-228
D2GRPH_LUT_SEL	2-210
D2GRPH_PITCH	2-211
D2GRPH_PRIMARY_SURFACE_ADDRESS	2-211
D2GRPH_SECONDARY_SURFACE_ADDRESS	2-211
D2GRPH_SURFACE_ADDRESS_INUSE	2-214
D2GRPH_SURFACE_OFFSET_X	2-212
D2GRPH_SURFACE_OFFSET_Y	2-212
D2GRPH_SWAP_CNTL	2-210
D2GRPH_UPDATE	2-213
D2GRPH_X_END	2-212

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D2GRPH_X_START	2-212
D2GRPH_Y_END	2-213
D2GRPH_Y_START	2-212
D2ICON_COLOR1	2-239
D2ICON_COLOR2	2-239
D2ICON_CONTROL	2-238
D2ICON_SIZE	2-239
D2ICON_START_POSITION	2-239
D2ICON_SURFACE_ADDRESS	2-238
D2ICON_UPDATE	2-240
D2MODE_MASTER_UPDATE_LOCK	2-285
D2MODE_MASTER_UPDATE_MODE	2-286
D2OVL_ALPHA	2-227
D2OVL_ALPHA_CONTROL	2-228
D2OVL_COLOR_MATRIX_TRANSFORMATION_CNTL	2-222
D2OVL_CONTROL1	2-215
D2OVL_CONTROL2	2-216
D2OVL_DFQ_CONTROL	2-219
D2OVL_DFQ_STATUS	2-219
D2OVL_ENABLE	2-215
D2OVL_END	2-218
D2OVL_KEY_ALPHA	2-230
D2OVL_KEY_CONTROL	2-227
D2OVL_KEY_RANGE_BLUE_CB	2-230
D2OVL_KEY_RANGE_GREEN_Y	2-229
D2OVL_KEY_RANGE_RED_CR	2-229
D2OVL_MATRIX_COEF_1_1	2-220
D2OVL_MATRIX_COEF_1_2	2-220
D2OVL_MATRIX_COEF_1_3	2-220
D2OVL_MATRIX_COEF_1_4	2-220
D2OVL_MATRIX_COEF_2_1	2-221
D2OVL_MATRIX_COEF_2_2	2-221
D2OVL_MATRIX_COEF_2_3	2-221
D2OVL_MATRIX_COEF_2_4	2-221
D2OVL_MATRIX_COEF_3_1	2-221
D2OVL_MATRIX_COEF_3_2	2-222
D2OVL_MATRIX_COEF_3_3	2-222
D2OVL_MATRIX_COEF_3_4	2-222
D2OVL_MATRIX_TRANSFORM_EN	2-219
D2OVL_PITCH	2-217
D2OVL_PWL_0TOF	2-223
D2OVL_PWL_100TO13F	2-224
D2OVL_PWL_10TO1F	2-223

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
D2OVL_PWL_140TO17F	2-224
D2OVL_PWL_180TO1BF	2-224
D2OVL_PWL_1C0TO1FF	2-225
D2OVL_PWL_200TO23F	2-225
D2OVL_PWL_20TO3F	2-223
D2OVL_PWL_240TO27F	2-225
D2OVL_PWL_280TO2BF	2-225
D2OVL_PWL_2C0TO2FF	2-226
D2OVL_PWL_300TO33F	2-226
D2OVL_PWL_340TO37F	2-226
D2OVL_PWL_380TO3BF	2-226
D2OVL_PWL_3C0TO3FF	2-226
D2OVL_PWL_40TO7F	2-223
D2OVL_PWL_80TOBF	2-224
D2OVL_PWL_C0TOFF	2-224
D2OVL_PWL_TRANSFORM_EN	2-223
D2OVL_RT_BAND_POSITION	2-234
D2OVL_RT_PROCEED_COND	2-234
D2OVL_RT_SKEWCOMMAND	2-234
D2OVL_RT_SKEWCONTROL	2-234
D2OVL_RT_STAT	2-235
D2OVL_START	2-218
D2OVL_SURFACE_ADDRESS	2-217
D2OVL_SURFACE_ADDRESS_INUSE	2-219
D2OVL_SURFACE_OFFSET_X	2-217
D2OVL_SURFACE_OFFSET_Y	2-217
D2OVL_SWAP_CNTL	2-216
D2OVL_UPDATE	2-218
D2VGA_CONTROL	2-164
DAC_DATA	2-141
DAC_MASK	2-141
DAC_R_INDEX	2-141
DAC_W_INDEX	2-141
DACA_AUTODETECT_CONTROL	2-290
DACA_AUTODETECT_CONTROL2	2-290
DACA_AUTODETECT_CONTROL3	2-290
DACA_AUTODETECT_INT_CONTROL	2-291
DACA_AUTODETECT_STATUS	2-291
DACA_COMPARATOR_ENABLE	2-293
DACA_COMPARATOR_OUTPUT	2-293
DACA_CONTROL1	2-292
DACA_CONTROL2	2-292

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
DACA_CRC_CONTROL	2-288
DACA_CRC_EN	2-288
DACA_CRC_SIG_CONTROL	2-289
DACA_CRC_SIG_CONTROL_MASK	2-289
DACA_CRC_SIG_RGB	2-289
DACA_CRC_SIG_RGB_MASK	2-289
DACA_DFT_CONFIG	2-294
DACA_ENABLE	2-288
DACA_FORCE_DATA	2-292
DACA_FORCE_OUTPUT_CNTL	2-291
DACA_POWERDOWN	2-292
DACA_PWR_CNTL	2-294
DACA_SOURCE_SELECT	2-288
DACA_SYNC_SELECT	2-289
DACA_SYNC_TRISTATE_CONTROL	2-289
DACA_TEST_ENABLE	2-294
DACB_AUTODETECT_CONTROL	2-296
DACB_AUTODETECT_CONTROL2	2-297
DACB_AUTODETECT_CONTROL3	2-297
DACB_AUTODETECT_INT_CONTROL	2-298
DACB_AUTODETECT_STATUS	2-297
DACB_COMPARATOR_ENABLE	2-299
DACB_COMPARATOR_OUTPUT	2-300
DACB_CONTROL1	2-299
DACB_CONTROL2	2-299
DACB_CRC_CONTROL	2-295
DACB_CRC_EN	2-295
DACB_CRC_SIG_CONTROL	2-296
DACB_CRC_SIG_CONTROL_MASK	2-295
DACB_CRC_SIG_RGB	2-296
DACB_CRC_SIG_RGB_MASK	2-295
DACB_ENABLE	2-294
DACB_FORCE_DATA	2-298
DACB_FORCE_OUTPUT_CNTL	2-298
DACB_POWERDOWN	2-298
DACB_PWR_CNTL	2-300
DACB_SOURCE_SELECT	2-294
DACB_SYNC_SELECT	2-296
DACB_SYNC_TRISTATE_CONTROL	2-296
DACB_TEST_ENABLE	2-300
DC_CRTC_MASTER_EN	2-255
DC_CRTC_TV_CONTROL	2-255
DC_GENERICA	2-342

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
DC_GENERICB	2-342
DC_GPIO_DDC1_A	2-352
DC_GPIO_DDC1_EN	2-353
DC_GPIO_DDC1_MASK	2-352
DC_GPIO_DDC1_Y	2-353
DC_GPIO_DDC2_A	2-353
DC_GPIO_DDC2_EN	2-354
DC_GPIO_DDC2_MASK	2-353
DC_GPIO_DDC2_Y	2-354
DC_GPIO_DDC3_A	2-354
DC_GPIO_DDC3_EN	2-355
DC_GPIO_DDC3_MASK	2-354
DC_GPIO_DDC3_Y	2-355
DC_GPIO_DDC4_A	2-350
DC_GPIO_DDC4_EN	2-350
DC_GPIO_DDC4_MASK	2-350
DC_GPIO_DDC4_Y	2-350
DC_GPIO_DEBUG	2-350
DC_GPIO_DVODATA_A	2-351
DC_GPIO_DVODATA_EN	2-351
DC_GPIO_DVODATA_MASK	2-351
DC_GPIO_DVODATA_Y	2-352
DC_GPIO_GENERIC_A	2-349
DC_GPIO_GENERIC_EN	2-349
DC_GPIO_GENERIC_MASK	2-349
DC_GPIO_GENERIC_Y	2-349
DC_GPIO_HPD_A	2-357
DC_GPIO_HPD_EN	2-358
DC_GPIO_HPD_MASK	2-357
DC_GPIO_HPD_Y	2-358
DC_GPIO_PAD_STRENGTH_1	2-359
DC_GPIO_PAD_STRENGTH_2	2-360
DC_GPIO_PWRSEQ_A	2-359
DC_GPIO_PWRSEQ_EN	2-359
DC_GPIO_PWRSEQ_MASK	2-358
DC_GPIO_PWRSEQ_Y	2-359
DC_GPIO_SYNCA_A	2-355
DC_GPIO_SYNCA_EN	2-356
DC_GPIO_SYNCA_MASK	2-355
DC_GPIO_SYNCA_Y	2-356
DC_GPIO_SYNCB_A	2-356
DC_GPIO_SYNCB_EN	2-357

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
DC_GPIO_SYNCB_MASK	2-356
DC_GPIO_SYNCB_Y	2-357
DC_HOT_PLUG_DETECT_CLOCK_CONTROL	2-340
DC_HOT_PLUG_DETECT1_CONTROL	2-339
DC_HOT_PLUG_DETECT1_INT_CONTROL	2-339
DC_HOT_PLUG_DETECT1_INT_STATUS	2-339
DC_HOT_PLUG_DETECT2_CONTROL	2-339
DC_HOT_PLUG_DETECT2_INT_CONTROL	2-340
DC_HOT_PLUG_DETECT2_INT_STATUS	2-339
DC_HOT_PLUG_DETECT3_CONTROL	2-340
DC_HOT_PLUG_DETECT3_INT_CONTROL	2-341
DC_HOT_PLUG_DETECT3_INT_STATUS	2-340
DC_I2C_ARBITRATION	2-99
DC_I2C_CONTROL	2-98
DC_I2C_DATA	2-106
DC_I2C_DDC1_HW_STATUS	2-100
DC_I2C_DDC1_SETUP	2-102
DC_I2C_DDC1_SPEED	2-101
DC_I2C_DDC2_HW_STATUS	2-101
DC_I2C_DDC2_SETUP	2-102
DC_I2C_DDC2_SPEED	2-102
DC_I2C_DDC3_HW_STATUS	2-101
DC_I2C_DDC3_SETUP	2-103
DC_I2C_DDC3_SPEED	2-103
DC_I2C_DDC4_HW_STATUS	2-109
DC_I2C_DDC4_SETUP	2-110
DC_I2C_DDC4_SPEED	2-110
DC_I2C_INTERRUPT_CONTROL	2-99
DC_I2C_SW_STATUS	2-100
DC_I2C_TRANSACTION0	2-104
DC_I2C_TRANSACTION1	2-104
DC_I2C_TRANSACTION2	2-105
DC_I2C_TRANSACTION3	2-105
DC_LUT_30_COLOR	2-243
DC_LUT_AUTOFILL	2-243
DC_LUT_PWL_DATA	2-242
DC_LUT_READ_PIPE_SELECT	2-243
DC_LUT_RW_INDEX	2-242
DC_LUT_RW_MODE	2-242
DC_LUT_RW_SELECT	2-241
DC_LUT_SEQ_COLOR	2-242
DC_LUT_WRITE_EN_MASK	2-243
DC_LUTA_BLACK_OFFSET_BLUE	2-246

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
DC_LUTA_BLACK_OFFSET_GREEN	2-246
DC_LUTA_BLACK_OFFSET_RED	2-246
DC_LUTA_CONTROL	2-244
DC_LUTA_WHITE_OFFSET_BLUE	2-246
DC_LUTA_WHITE_OFFSET_GREEN	2-246
DC_LUTA_WHITE_OFFSET_RED	2-246
DC_LUTB_BLACK_OFFSET_BLUE	2-249
DC_LUTB_BLACK_OFFSET_GREEN	2-249
DC_LUTB_BLACK_OFFSET_RED	2-249
DC_LUTB_CONTROL	2-247
DC_LUTB_WHITE_OFFSET_BLUE	2-249
DC_LUTB_WHITE_OFFSET_GREEN	2-249
DC_LUTB_WHITE_OFFSET_RED	2-249
DC_MVP_LB_CONTROL	2-254
DC_PAD_EXTERN_SIG	2-342
DC_REF_CLK_CNTL	2-343
DCP_CRC_CONTROL	2-250
DCP_CRC_MASK	2-250
DCP_CRC_P0_CURRENT	2-250
DCP_CRC_P0_LAST	2-250
DCP_CRC_P1_CURRENT	2-250
DCP_CRC_P1_LAST	2-251
DCP_LB_DATA_GAP_BETWEEN_CHUNK	2-254
DCP_MULTI_CHIP_CNTL	2-252
DCP_TILING_CONFIG	2-251
DEVICE_CAP	2-79
DEVICE_CAP2	2-83
DEVICE_CNTL	2-80
DEVICE_CNTL2	2-83
DEVICE_ID	2-73
DEVICE_STATUS	2-81
DEVICE_STATUS2	2-83
DISP_INTERRUPT_STATUS	2-343
DISP_INTERRUPT_STATUS_CONTINUE	2-345
DISP_TIMER_CONTROL	2-347
DLL_CNTL	2-92
DMA_VIP0_TABLE_ADDR	2-130
DMA_VIP1_TABLE_ADDR	2-130
DMA_VIP2_TABLE_ADDR	2-130
DMA_VIP3_TABLE_ADDR	2-130
DMA_VIPH_ABORT	2-131
DMA_VIPH_CHUNK_0	2-129

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
DMA_VIPH_CHUNK_1_VAL	2-129
DMA_VIPH_MISC_CNTL	2-128
DMA_VIPH_STATUS	2-128
DMA_VIPH0_ACTIVE	2-130
DMA_VIPH0_COMMAND	2-126
DMA_VIPH1_ACTIVE	2-130
DMA_VIPH1_COMMAND	2-126
DMA_VIPH2_ACTIVE	2-131
DMA_VIPH2_COMMAND	2-127
DMA_VIPH3_ACTIVE	2-131
DMA_VIPH3_COMMAND	2-127
DMIF_CONTROL	2-252
DMIF_STATUS	2-253
DOUT_POWER_MANAGEMENT_CNTL	2-346
DVOA_BIT_DEPTH_CONTROL	2-315
DVOA_CONTROL	2-316
DVOA_CRC_CONTROL	2-317
DVOA_CRC_EN	2-316
DVOA_CRC_SIG_MASK1	2-317
DVOA_CRC_SIG_MASK2	2-317
DVOA_CRC_SIG_RESULT1	2-317
DVOA_CRC_SIG_RESULT2	2-317
DVOA_CRC2_SIG_MASK	2-318
DVOA_CRC2_SIG_RESULT	2-318
DVOA_ENABLE	2-314
DVOA_FORCE_DATA	2-319
DVOA_FORCE_OUTPUT_CNTL	2-318
DVOA_OUTPUT	2-316
DVOA_SOURCE_SELECT	2-314
DVOA_STRENGTH_CONTROL	2-318
ERROR_STATUS	2-93
EXTERN_TRIG_CNTL	2-137
GENENB	2-139
GENERAL_PWRMGT	2-90
GENERIC_I2C_CONTROL	2-106
GENERIC_I2C_DATA	2-108
GENERIC_I2C_INTERRUPT_CONTROL	2-106
GENERIC_I2C_PIN_DEBUG	2-109
GENERIC_I2C_PIN_SELECTION	2-109
GENERIC_I2C_SETUP	2-107
GENERIC_I2C_SPEED	2-107
GENERIC_I2C_STATUS	2-107
GENERIC_I2C_TRANSACTION	2-108

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
GENFC_RD	2-140
GENFC_WT	2-140
GENMO_RD	2-139
GENMO_WT	2-139
GENS0	2-140
GENS1	2-140
GPIOPAD_A	2-132
GPIOPAD_EN	2-132
GPIOPAD_EXTERN_TRIG_CNTL	2-133
GPIOPAD_MASK	2-132
GPIOPAD_STRENGTH	2-132
GPIOPAD_Y	2-132
GRA00	2-151
GRA01	2-152
GRA02	2-152
GRA03	2-152
GRA04	2-152
GRA05	2-152
GRA06	2-153
GRA07	2-153
GRA08	2-153
GRPH8_DATA	2-151
GRPH8_IDX	2-151
HEADER	2-76
I2C_CNTL_0	2-97
I2C_CNTL_1	2-97
I2C_DATA	2-98
INTERRUPT_LINE	2-77
INTERRUPT_PIN	2-77
LATENCY	2-76
LINK_CAP	2-81
LINK_CAP2	2-83
LINK_CNTL	2-82
LINK_CNTL2	2-84
LINK_STATUS	2-82
LINK_STATUS2	2-84
LVTMA_2ND_CRC_RESULT	2-324
LVTMA_BIT_DEPTH_CONTROL	2-321
LVTMA_BL_MOD_CNTL	2-331
LVTMA_CNTL	2-319
LVTMA_COLOR_FORMAT	2-320
LVTMA_CONTROL_CHAR	2-322

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
LVTMA_CONTROL0_FEEDBACK	2-322
LVTMA_CRC_CNTL	2-323
LVTMA_CRC_SIG_MASK	2-323
LVTMA_CRC_SIG_RGB	2-324
LVTMA_CTL_BITS	2-325
LVTMA_CTL0_1_GEN_CNTL	2-326
LVTMA_CTL2_3_GEN_CNTL	2-328
LVTMA_DATA_SYNCHRONIZATION	2-326
LVTMA_DCBALANCER_CONTROL	2-326
LVTMA_DEBUG	2-325
LVTMA_DITHER_RAND_SEED	2-335
LVTMA_FORCE_DATA	2-321
LVTMA_FORCE_OUTPUT_CNTL	2-320
LVTMA_LOAD_DETECT	2-333
LVTMA_LVDS_DATA_CNTL	2-332
LVTMA_MACRO_CONTROL	2-333
LVTMA_MODE	2-332
LVTMA_PREEMPHASIS_CONTROL	2-336
LVTMA_PWRSEQ_CNTL	2-329
LVTMA_PWRSEQ_DELAY1	2-329
LVTMA_PWRSEQ_DELAY2	2-329
LVTMA_PWRSEQ_REF_DIV	2-329
LVTMA_PWRSEQ_STATE	2-331
LVTMA_RANDOM_PATTERN_SEED	2-325
LVTMA_RED_BLUE_SWITCH	2-326
LVTMA_REG_TEST_OUTPUT	2-335
LVTMA_SOURCE_SELECT	2-320
LVTMA_SPLIT_LOAD_DETECT	2-337
LVTMA_SPLIT_PLL_ADJUST	2-337
LVTMA_SPLIT_TRANSMITTER_ADJUST	2-338
LVTMA_SPLIT_TRANSMITTER_CONTROL	2-337
LVTMA_SPLIT_TRANSMITTER_ENABLE	2-336
LVTMA_STEREOSYNC_CTL_SEL	2-322
LVTMA_SYNC_CHAR_PATTERN_0_1	2-322
LVTMA_SYNC_CHAR_PATTERN_2_3	2-323
LVTMA_SYNC_CHAR_PATTERN_SEL	2-322
LVTMA_TEST_PATTERN	2-324
LVTMA_TRANSMITTER_ADJUST	2-336
LVTMA_TRANSMITTER_CONTROL	2-334
LVTMA_TRANSMITTER_DEBUG	2-335
LVTMA_TRANSMITTER_ENABLE	2-332
MAX_LATENCY	2-77
MC_CONFIG	2-2

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
MC_IMP_CNTL	2-10
MC_IMP_DEBUG	2-10
MC_IMP_STATUS	2-10
MC_IO_A_PAD_CNTL_D0_I0	2-33
MC_IO_A_PAD_CNTL_D0_I1	2-34
MC_IO_A_PAD_CNTL_D1_I0	2-45
MC_IO_A_PAD_CNTL_D1_I1	2-45
MC_IO_CK_PAD_CNTL_D0_I0	2-31
MC_IO_CK_PAD_CNTL_D0_I1	2-31
MC_IO_CK_PAD_CNTL_D1_I0	2-43
MC_IO_CK_PAD_CNTL_D1_I1	2-43
MC_IO_CMD_PAD_CNTL_D0_I0	2-31
MC_IO_CMD_PAD_CNTL_D0_I1	2-32
MC_IO_CMD_PAD_CNTL_D1_I0	2-43
MC_IO_CMD_PAD_CNTL_D1_I1	2-43
MC_IO_DQ_PAD_CNTL_D0_I0	2-32
MC_IO_DQ_PAD_CNTL_D0_I1	2-32
MC_IO_DQ_PAD_CNTL_D1_I0	2-44
MC_IO_DQ_PAD_CNTL_D1_I1	2-44
MC_IO_PAD_CNTL	2-11
MC_IO_PAD_CNTL_D0	2-25
MC_IO_PAD_CNTL_D0_I0	2-28
MC_IO_PAD_CNTL_D0_I1	2-28
MC_IO_PAD_CNTL_D1	2-25
MC_IO_PAD_CNTL_D1_I0	2-39
MC_IO_PAD_CNTL_D1_I1	2-40
MC_IO_QS_PAD_CNTL_D0_I0	2-33
MC_IO_QS_PAD_CNTL_D0_I1	2-33
MC_IO_QS_PAD_CNTL_D1_I0	2-44
MC_IO_QS_PAD_CNTL_D1_I1	2-45
MC_IO_RD_DQ_CNTL_D0_I0	2-29
MC_IO_RD_DQ_CNTL_D0_I1	2-29
MC_IO_RD_DQ_CNTL_D1_I0	2-41
MC_IO_RD_DQ_CNTL_D1_I1	2-41
MC_IO_RD_QS_CNTL_D0_I0	2-30
MC_IO_RD_QS_CNTL_D0_I1	2-30
MC_IO_RD_QS_CNTL_D1_I0	2-41
MC_IO_RD_QS_CNTL_D1_I1	2-42
MC_IO_RD_QS2_CNTL_D0_I0	2-30
MC_IO_RD_QS2_CNTL_D0_I1	2-30
MC_IO_RD_QS2_CNTL_D1_I0	2-42
MC_IO_RD_QS2_CNTL_D1_I1	2-42

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
MC_IO_RD_STR_NCNTL_B0_D0	2-35
MC_IO_RD_STR_NCNTL_B0_D1	2-46
MC_IO_RD_STR_NCNTL_B1_D0	2-35
MC_IO_RD_STR_NCNTL_B1_D1	2-46
MC_IO_RD_STR_NCNTL_B2_D0	2-35
MC_IO_RD_STR_NCNTL_B2_D1	2-47
MC_IO_RD_STR_NCNTL_B3_D0	2-35
MC_IO_RD_STR_NCNTL_B3_D1	2-47
MC_IO_RD_STR_NCNTL_B4_D0	2-36
MC_IO_RD_STR_NCNTL_B4_D1	2-47
MC_IO_RD_STR_NCNTL_B5_D0	2-36
MC_IO_RD_STR_NCNTL_B5_D1	2-47
MC_IO_RD_STR_NCNTL_B6_D0	2-36
MC_IO_RD_STR_NCNTL_B6_D1	2-48
MC_IO_RD_STR_NCNTL_B7_D0	2-37
MC_IO_RD_STR_NCNTL_B7_D1	2-48
MC_IO_WR_CNTL_D0_I0	2-30
MC_IO_WR_CNTL_D0_I1	2-31
MC_IO_WR_CNTL_D1_I0	2-42
MC_IO_WR_CNTL_D1_I1	2-42
MC_IO_WR_DQ_CNTL_D0_I0	2-34
MC_IO_WR_DQ_CNTL_D0_I1	2-34
MC_IO_WR_DQ_CNTL_D1_I0	2-46
MC_IO_WR_DQ_CNTL_D1_I1	2-46
MC_IO_WR_QS_CNTL_D0_I0	2-34
MC_IO_WR_QS_CNTL_D0_I1	2-34
MC_IO_WR_QS_CNTL_D1_I0	2-46
MC_IO_WR_QS_CNTL_D1_I1	2-46
MC_NPL_STATUS	2-50
MC_PMG_CFG	2-9
MC_PMG_CMD	2-8
MC_SEQ_A_PAD_CNTL_D0_I0	2-27
MC_SEQ_A_PAD_CNTL_D0_I1	2-27
MC_SEQ_A_PAD_CNTL_D1_I0	2-39
MC_SEQ_A_PAD_CNTL_D1_I1	2-39
MC_SEQ_CAS_TIMING_B	2-5
MC_SEQ_CAS_TIMING_C	2-7
MC_SEQ_CAS_TIMING_P	2-4
MC_SEQ_CAS_TIMING_S	2-6
MC_SEQ_CK_PAD_CNTL_D0_I0	2-25
MC_SEQ_CK_PAD_CNTL_D0_I1	2-25
MC_SEQ_CK_PAD_CNTL_D1_I0	2-37
MC_SEQ_CK_PAD_CNTL_D1_I1	2-37

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
MC_SEQ_CMD	2-8
MC_SEQ_CMD_PAD_CNTL_D0_I0	2-26
MC_SEQ_CMD_PAD_CNTL_D0_I1	2-26
MC_SEQ_CMD_PAD_CNTL_D1_I0	2-37
MC_SEQ_CMD_PAD_CNTL_D1_I1	2-38
MC_SEQ_CNTL	2-2
MC_SEQ_DQ_PAD_CNTL_D0_I0	2-26
MC_SEQ_DQ_PAD_CNTL_D0_I1	2-26
MC_SEQ_DQ_PAD_CNTL_D1_I0	2-38
MC_SEQ_DQ_PAD_CNTL_D1_I1	2-38
MC_SEQ_DRAM	2-3
MC_SEQ_GENERAL_CONFIG	2-48
MC_SEQ_IO_CTL_D0	2-23
MC_SEQ_IO_CTL_D1	2-23
MC_SEQ_IO_CTL_UNUSED	2-24
MC_SEQ_MISC_TIMING_B	2-5
MC_SEQ_MISC_TIMING_C	2-7
MC_SEQ_MISC_TIMING_P	2-4
MC_SEQ_MISC_TIMING_S	2-6
MC_SEQ_MISC_TIMING2_B	2-5
MC_SEQ_MISC_TIMING2_C	2-8
MC_SEQ_MISC_TIMING2_P	2-4
MC_SEQ_MISC_TIMING2_S	2-7
MC_SEQ_NPL_CTL_D0	2-24
MC_SEQ_NPL_CTL_D1	2-24
MC_SEQ_QS_PAD_CNTL_D0_I0	2-27
MC_SEQ_QS_PAD_CNTL_D0_I1	2-27
MC_SEQ_QS_PAD_CNTL_D1_I0	2-38
MC_SEQ_QS_PAD_CNTL_D1_I1	2-39
MC_SEQ_RAS_TIMING_B	2-5
MC_SEQ_RAS_TIMING_C	2-7
MC_SEQ_RAS_TIMING_P	2-4
MC_SEQ_RAS_TIMING_S	2-6
MC_SEQ_RD_CTL_D0_B	2-14
MC_SEQ_RD_CTL_D0_C	2-20
MC_SEQ_RD_CTL_D0_P	2-11
MC_SEQ_RD_CTL_D0_S	2-17
MC_SEQ_RD_CTL_D1_B	2-15
MC_SEQ_RD_CTL_D1_C	2-21
MC_SEQ_RD_CTL_D1_P	2-12
MC_SEQ_RD_CTL_D1_S	2-18
MC_SEQ_RS_CNTL	2-48

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
MC_SEQ_STATUS_M	2-49
MC_SEQ_STATUS_S	2-49
MC_SEQ_WR_CTL_D0_B	2-17
MC_SEQ_WR_CTL_D0_C	2-22
MC_SEQ_WR_CTL_D0_P	2-14
MC_SEQ_WR_CTL_D0_S	2-19
MC_SEQ_WR_CTL_D1_B	2-17
MC_SEQ_WR_CTL_D1_C	2-23
MC_SEQ_WR_CTL_D1_P	2-14
MC_SEQ_WR_CTL_D1_S	2-20
MCIF_CONTROL	2-253
MCLK_PWRMGT_CNTL	2-91
MIN_GRANT	2-77
MM_DATA	2-51
MM_INDEX	2-51
MPLL_CNTL_MODE	2-90
MPLL_FUNC_CNTL	2-89
MPLL_TIME	2-93
MSI_CAP_LIST	2-84
MSI_MSG_ADDR_HI	2-85
MSI_MSG_ADDR_LO	2-85
MSI_MSG_CNTL	2-84
MSI_MSG_DATA	2-85
MSI_MSG_DATA_64	2-85
PCIE_ADV_ERR_CAP_CNTL	2-87
PCIE_ADV_ERR_RPT_ENH_CAP_LIST	2-85
PCIE_CAP	2-79
PCIE_CAP_LIST	2-78
PCIE_CI_CNTL	2-54
PCIE_CORR_ERR_MASK	2-87
PCIE_CORR_ERR_STATUS	2-87
PCIE_DATA	2-54
PCIE_ERR_CNTL	2-66
PCIE_HDR_LOG0	2-87
PCIE_HDR_LOG1	2-88
PCIE_HDR_LOG2	2-88
PCIE_HDR_LOG3	2-88
PCIE_INDEX	2-54
PCIE_LC_CNTL	2-69
PCIE_LC_CNTL2	2-70
PCIE_LC_LINK_WIDTH_CNTL	2-71
PCIE_LC_N_FTS_CNTL	2-71
PCIE_LC_STATE0	2-72

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
PCIE_LC_STATE1	2-72
PCIE_LC_STATE10	2-56
PCIE_LC_STATE11	2-56
PCIE_LC_STATE2	2-72
PCIE_LC_STATE3	2-72
PCIE_LC_STATE4	2-72
PCIE_LC_STATE5	2-73
PCIE_LC_STATE6	2-55
PCIE_LC_STATE7	2-55
PCIE_LC_STATE8	2-55
PCIE_LC_STATE9	2-55
PCIE_P_BUF_STATUS	2-57
PCIE_P_CNTL	2-56
PCIE_P_DECODE_ERR_CNT_0	2-63
PCIE_P_DECODE_ERR_CNT_1	2-63
PCIE_P_DECODE_ERR_CNT_10	2-64
PCIE_P_DECODE_ERR_CNT_11	2-64
PCIE_P_DECODE_ERR_CNT_12	2-65
PCIE_P_DECODE_ERR_CNT_13	2-65
PCIE_P_DECODE_ERR_CNT_14	2-65
PCIE_P_DECODE_ERR_CNT_15	2-65
PCIE_P_DECODE_ERR_CNT_2	2-63
PCIE_P_DECODE_ERR_CNT_3	2-63
PCIE_P_DECODE_ERR_CNT_4	2-63
PCIE_P_DECODE_ERR_CNT_5	2-63
PCIE_P_DECODE_ERR_CNT_6	2-64
PCIE_P_DECODE_ERR_CNT_7	2-64
PCIE_P_DECODE_ERR_CNT_8	2-64
PCIE_P_DECODE_ERR_CNT_9	2-64
PCIE_P_DECODE_ERR_CNTL	2-62
PCIE_P_DECODER_STATUS	2-58
PCIE_P_IMP_CNTL_STRENGTH	2-61
PCIE_P_IMP_CNTL_UPDATE	2-62
PCIE_P_MISC_DEBUG_STATUS	2-60
PCIE_P_PAD_MISC_CNTL	2-62
PCIE_P_STR_CNTL_UPDATE	2-62
PCIE_P_SYMSYNC_CTL	2-61
PCIE_RX_CNTL	2-67
PCIE_RX_CREDITS_ALLOCATED_CPL	2-68
PCIE_RX_CREDITS_ALLOCATED_NP	2-67
PCIE_RX_CREDITS_ALLOCATED_P	2-67
PCIE_RX_CREDITS_RECEIVED_CPL	2-68

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
PCIE_RX_CREDITS_RECEIVED_NP	2-68
PCIE_RX_CREDITS_RECEIVED_P	2-68
PCIE_RX_NUM_NACK	2-54
PCIE_RX_NUM_NACK_GENERATED	2-54
PCIE_TX_CNTL	2-65
PCIE_TX_REPLAY	2-66
PCIE_TX_SEQ	2-66
PCIE_UNCORR_ERR_MASK	2-86
PCIE_UNCORR_ERR_SEVERITY	2-86
PCIE_UNCORR_ERR_STATUS	2-86
PLL_BYPASSCLK_SEL	2-89
PLL_TEST_CNTL	2-94
PMI_CAP	2-78
PMI_CAP_LIST	2-78
PMI_STATUS_CNTL	2-78
PROG_INTERFACE	2-75
REVISION_ID	2-75
ROM_CNTL	2-137
ROM_DATA	2-138
ROM_INDEX	2-138
ROM_START	2-138
ROM_STATUS	2-137
SCLK_PWRMGT_CNTL	2-91
SEQ00	2-142
SEQ01	2-142
SEQ02	2-143
SEQ03	2-143
SEQ04	2-143
SEQ8_DATA	2-142
SEQ8_IDX	2-142
SPLL_CNTL_MODE	2-89
SPLL_TIME	2-93
STATUS	2-74
SUB_CLASS	2-75
TMDSA_2ND_CRC_RESULT	2-306
TMDSA_BIT_DEPTH_CONTROL	2-303
TMDSA_CNTL	2-301
TMDSA_COLOR_FORMAT	2-302
TMDSA_CONTROL_CHAR	2-303
TMDSA_CONTROL0_FEEDBACK	2-304
TMDSA_CRC_CNTL	2-304
TMDSA_CRC_SIG_MASK	2-305
TMDSA_CRC_SIG_RGB	2-305

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
TMDSA_CTL_BITS	2-307
TMDSA_CTL0_1_GEN_CNTL	2-308
TMDSA_CTL2_3_GEN_CNTL	2-309
TMDSA_DATA_SYNCHRONIZATION	2-308
TMDSA_DCBALANCER_CONTROL	2-307
TMDSA_DEBUG	2-307
TMDSA_DITHER_RAND_SEED	2-313
TMDSA_FORCE_DATA	2-302
TMDSA_FORCE_OUTPUT_CNTL	2-302
TMDSA_LOAD_DETECT	2-311
TMDSA_PLL_ADJUST	2-311
TMDSA_RANDOM_PATTERN_SEED	2-306
TMDSA_RED_BLUE_SWITCH	2-307
TMDSA_REG_TEST_OUTPUTA	2-313
TMDSA_REG_TEST_OUTPUTB	2-314
TMDSA_SOURCE_SELECT	2-301
TMDSA_STEREOSYNC_CTL_SEL	2-304
TMDSA_SYNC_CHAR_PATTERN_0_1	2-304
TMDSA_SYNC_CHAR_PATTERN_2_3	2-304
TMDSA_SYNC_CHAR_PATTERN_SEL	2-304
TMDSA_TEST_PATTERN	2-306
TMDSA_TRANSMITTER_ADJUST	2-314
TMDSA_TRANSMITTER_CONTROL	2-311
TMDSA_TRANSMITTER_DEBUG	2-313
TMDSA_TRANSMITTER_ENABLE	2-310
VENDOR_ID	2-73
VGA_CACHE_CONTROL	2-163
VGA_DEBUG_READBACK_DATA	2-168
VGA_DEBUG_READBACK_INDEX	2-168
VGA_DISPBUF1_SURFACE_ADDR	2-162
VGA_DISPBUF2_SURFACE_ADDR	2-162
VGA_HDP_CONTROL	2-162
VGA_HW_DEBUG	2-165
VGA_INTERRUPT_CONTROL	2-165
VGA_INTERRUPT_STATUS	2-166
VGA_MAIN_CONTROL	2-166
VGA_MEM_READ_PAGE_ADDR	2-169
VGA_MEM_WRITE_PAGE_ADDR	2-169
VGA_MEMORY_BASE_ADDRESS	2-162
VGA_MODE_CONTROL	2-161
VGA_RENDER_CONTROL	2-159
VGA_SEQUENCER_RESET_CONTROL	2-160

Table A-18 All Registers Sorted by Name (Continued)

Register Name	Page
VGA_STATUS	2-165
VGA_STATUS_CLEAR	2-166
VGA_SURFACE_PITCH_SELECT	2-161
VGA_TEST_CONTROL	2-168
VID_BUFFER_CONTROL	2-116
VIPH_CH0_ABCNT	2-113
VIPH_CH0_ADDR	2-111
VIPH_CH0_DATA	2-111
VIPH_CH0_SBCNT	2-112
VIPH_CH1_ABCNT	2-113
VIPH_CH1_ADDR	2-112
VIPH_CH1_DATA	2-111
VIPH_CH1_SBCNT	2-112
VIPH_CH2_ABCNT	2-113
VIPH_CH2_ADDR	2-112
VIPH_CH2_DATA	2-111
VIPH_CH2_SBCNT	2-113
VIPH_CH3_ABCNT	2-113
VIPH_CH3_ADDR	2-112
VIPH_CH3_DATA	2-111
VIPH_CH3_SBCNT	2-113
VIPH_CONTROL	2-114
VIPH_DMA_CHUNK	2-115
VIPH_DV_INT	2-115
VIPH_DV_LAT	2-114
VIPH_REG_ADDR	2-110
VIPH_REG_DATA	2-111
VIPH_TIMEOUT_STAT	2-115
VIPPAD_A	2-134
VIPPAD_EN	2-135
VIPPAD_MASK	2-133
VIPPAD_STRENGTH	2-136
VIPPAD_Y	2-136

Appendix B
Revision History

Rev 1.01o (December 07)

- Open source release.

This page intentionally left blank.