GLINT R5®

Reference Guide Volume II-

Hardware Registers

PROPRIETARY AND CONFIDENTIAL INFORMATION

3D*labs*®

GLINT R5®

Reference Guide Volume II-

Hardware Registers

PROPRIETARY AND CONFIDENTIAL INFORMATION

Issue 3

Proprietary Notice

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

3D*labs* Ltd.
Meadlake Place
Thorpe Lea Road, Egham
Surrey, TW20 8HE
United Kingdom
Tel: +44 (0) 1784 470555
Fax: +44 (0) 1784 470699

3D*labs* GmbH
Breckenheimer Weg 29
65205 Wiesbaden
Deutschland
Tel: +49 6122 916 778
Fax: +49 6122 919 646

3D*labs* K.K.
Shiroyama JT Mori Bldg 16F
40301 Toranomon
Minato-ku, Tokyo, 105, Japan
Tel: +81-3-5403-4653
Fax: +91-3-5403-4646

3D*labs* Inc.
480 Potrero Avenue
Sunnyvale, CA 94086,
United States
Tel: +1 (408) 530-4700
Fax: +1 (408) 530-4701

Change History

Document	Issue	Date	Change
172.1.2	1	15/02/2000	Creation
172.1.2	2	15/09/2000	First full release following engineering review.
172.1.2	3	05/03/2001	Add DMA Arbiter PCI Interrupt signals, corrected bitfiled options in RDKClkControl and RDMClkControl, added DMAArbiter registers, rddacontrol sync on green reserved, Interleave note in LocalMemControl and use of MemCount for delay timing, font for tickmarks corrected, added bookmarks, face normals removed, minor changes to format and fonts.

Table of Contents

4

Hardware Registers

This chapter lists GLINT R5 hardware registers by region and functional offset group. Within each group, the registers are listed alphanumerically. Exceptionally, graphics core "software" registers (offset 8000-9FFF) are shown in chapter 5. Global cross-reference listings in alphanumeric and offset order are available in chapter 6.

Register details have the following format information:

Name	The register's name.
Type	The region in which the register functions.
Offset	The offset of this register from the base address of the region.
Format	Can be bitfield or integer.
Bit	Bit Name
Read	Indicates whether the register bit can be read from. A ✓ mark indicates the register can be read from, a ✕ indicates the register bit is not readable.
Write	Indicates whether the register bit can be written to. A ✓ mark indicates the register can be written to, a ✕ indicates the register bit is not writable.
Reset	The value of the register following hardware reset.
Description	In the register descriptions:
Reserved Bits	Indicates bits that may be used in future members of the Permedia family. To ensure upwards compatibility software should not assume a value for these bits when read.
Not Used/ Unused Bits	Indicates bits that are adjacent to numeric fields. These may be used in future members of the Permedia family, but only to extend the dynamic range of these fields. The data returned from a read of these bits is undefined. When a Not Used field resides in the most significant position, a good convention to follow is to sign extend the numeric value, rather than masking the field to zero before writing the register. This will ensure compatibility if the dynamic range is increased in future.
Reserved Registers	Write accesses to reserved registers are accepted by the bus interface but the data is discarded. Read accesses return 0. Data written to reserved registers is never forwarded.

For enumeration fields that do not specify the full range of possible values use only the specified values. An example of an enumeration field is the comparison field in the **DepthMode** register. Future chips may define a meaning for the unused values.

4.1 PCI Bus Interface Registers

The bus interface contains a number of PCI Configuration Registers, and also various Control Status registers for the chip, with accesses to these registers being handled entirely by the bus interface unit.

The bus interface also accesses the read-back path from the Graphics Processor. This is used to access registers in the GP without passing through the pipeline. The read-back operates by sending a tag to the read-back port and waiting a set number of clocks. When the delay has expired, data from the register corresponding to the tag will be present on the read-back port.

A separate port is provided to forward accesses to the RAMDAC Interface. The VGA unit is accessed via the Bypass FIFO. A separate port is also provided to access the EEPROM and VMI interface. The related registers are described below.

4.1.1 Reset

During soft reset the PCI Bus Region 0 registers are reset with the GP input and output FIFOs. However the bus master and slave state machines continue to run, which can result in the PCI trying to load the GPInFIFO during a reset. For details on Configuration Pins and the Reset process see Volume IV, Reset.

Driver software is required to write to a PCI configuration register to disable the bus master before asserting a software reset; this ensures that the master is not trying to load the GP Input FIFO during such a reset.

Note: When bus retries are disabled, the current implementation accepts and then discards all write accesses to the GP Input FIFO — this is different from the manner in which Bypass accesses are handled. The situation only occurs if driver software performs a soft reset and does not check that it has completed before writing to the FIFO.

4.2 PCI Configuration Region (0x00-0x30)

The configuration registers are accessed and modified by the use of PCI Configuration Read and Write commands, and will normally be initialised by BIOS or similar low-level code at system power-up and reset.

Sixty four bytes of the Configuration Registers are predefined within the PCI Specification and are supported by R5. These are defined below and are all implemented within the PCI Bus Interface. Registers are provided for device identification, PCI control and status, and as base address registers for the relocatable memory regions.

The base address registers allow the boot software to relocate PCI devices in the address spaces. At system power-up, device-independent software must be able to determine what devices are present, build a consistent address map, and determine if a device has an Expansion ROM. These base registers allow power-up software to build a consistent address map, by determining the size of each region and setting its base address within the memory or I/O map. Only offsets 0x10 to 0x1B are populated with Base Address Register information. Offsets 0x1C to 0x27 are reserved.

AGP Command

Name	Type	Offset	Format
CFGAG1 Command	Config	0x48	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	DataRate	✓	✓	000b	0 = AGP disabled 1 = 1X transfer rate 2 = 2X transfer rate 4 = 4X transfer rate Setting this field to any other value will disable AGP mastering.
3	Reserved	✓	✕	0b	
4	FWEnable	✓	✓	0	0 = Fast Write disabled 1 = Fast Write enabled
5	4GEnable	✓	✓	0	0 = 4G Addressing disabled 1 = 4G Addressing enabled
6..7	Reserved	✓	✓	00b	
8	AGPEnable	✓	✓	0	0 = AGP Mastering disabled 1 = AGP Mastering enabled
9	SBAEnable	✓	✓	0	0 = sideband addressing disabled 1 = sideband addressing enabled
10..23	Reserved	✓	✕	00.000 0.0000 .0000b	
24..31	RQDepth	✓	✓	0	Maximum number of AGP requests that can be queued. The RQDepth set in this field should never exceed the value in the CFGAGPStatus register. The maximum *RQDepth* used internally is the lower of these two *RQDepth* fields in case this field has been programmed incorrectly.

Notes: This register controls the operation of the AGP interface.

- If *AGPCapable* is not set, writes to this register should be discarded.
- If *SBACapable* is not set and SBAEnable is set, AGP accesses should be disabled.
- *AGPCapable* is a term used to express the logical OR of AGP1X Capable with AGP2X Capable with AGP4X Capable.

CFGAGPRev

Name	Type	Offset	Format
CFGAG1 Rev	Configuration	0x042	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...3	Minor Rev	✓	✕	0	Configured by AGP Capbable 0x00 if AGP Capable = 0. or 1. *(sic)*
4...7	Major Rev	✓	✕	See Desc.	Configured by AGP Capable • 0x00 when AGP Capable = 0. • 0x02 when AGP Capable = 1. *(sic)*

Notes: This register reports the revision of the AGP specification to which the device conforms. AGP Capable is a term used to express the logical OR of AGP1XCapable with AGP2XCapable with AGP4XCapable. The revision level is currently 2.0, reflecting the 2.0 AGP specification.

CFGAGPStatus

Name	Type	Offset	Format
CFGAG1 Status	Configuration	0x044	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Rate	✓	✕	see Desc.	Configured by AGP 1X Capable, Configured by AGP 2X Capable, Configured by AGP 4X Capable 0 = Configured by AGP 1X Capable 1 = Configured by AGP 2X Capable 2 = Configured by AGP 4X Capable
3	Reserved	✓	✕	0b	
4	FW	✓	✓	0b	Fast writes are not supported on R5
5	4G	✓	✓	0b	0 if AGP Capable = 0 1 if AGP Capable = 1
6...8	Reserved	✕	✕	000b	
9	SBA	✓	✕	see Desc.	Configured by AGP Capable Sideband Addressing 0 when AGP Capable = 0 1 when AGPCapable = 1 and SBACapable = 1
10..23	Reserved	✓	✕	00.000 0.0000 .0000b	
24..31	RQ	✓	✕	see Desc.	Maximum number of AGP requests supported Configured by AGP Capable 0x00 if AGP Capable = 0 0x1F if AGP Capable= 1 (=32 outstanding requests)

Notes: This register describes the AGP capabilities of the device. AGP Capable is a term used to express the logical OR of AGP1XCapable with AGP2XCapable with AGP4XCapable.

CFGBase Address 0

Name	Type	Offset	Format
CFGBase Addr0	Configuration	0x10	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Memory Space Indicator	✓	✕	0	0 = Region is in PCI memory space.
1..2	Address Type	✓	✕	0	0 = Memory Space, not prefetchable, in 32 bit address space
3	Prefetchable	✓	✕	0	0 = Region is not prefetchable.
4..16	Size Indication	✓	✕	0	0 = Control registers must be mapped into 128 Kbyte region.
17..20	Size Indication	✓	✓	0	Allow bits 17...31 to define the base address.
21...31	Base Address	✓	✓	X	Loaded at boot time to set base address of PCI Region 0

Notes: Base Address 0 Register contains the GLINT R5 control space offset. The control registers are in memory space. They are not prefetchable and can be located anywhere in 32 bit address space.

CFGBase Address 1

Name	Type	Offset	Format
CFGBase Addr1	Configuration	0x14	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Memory Space Indicator	✓	✕	0	0 = Region is in PCI memory space.
1..2	Address Type	✓	✕	0	00b = Locate anywhere in 32 bit address space
3	Prefetchable	✓	✕	0	0 = Region is not prefetchable if *PrefetchEnable* is 0. 1= Region is prefetchable if *PrefetchEnable* is 1.
4..N	Size Indication	✓	✕	0	These bits are 0. N is defined by the *BaseAddress1Size* configuration pin value..
N..31	BaseAddress	✓	✓	0	Loaded at boot time to set base address of PCI Region 1

Notes: The **BaseAddress1** Register contains the R5 aperture 1 memory offset and defines the size and type of the region. Typically the aperture is configured for 128Mbytes.

CFGBase Address 2

Name	Type	Offset	Format
CFGBase Addr2	Configuration	0x18	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Memory Space Indicator	✓	✕	0	0 = Region is in PCI memory space.
1..2	Address Type	✓	✕	0	00b = Locate anywhere in 32 bit address space
3	Prefetchable	✓	✕	0	0 = Region is not prefetchable if *PrefetchEnable* = 0. 1= Region is prefetchable if *PrefetchEnable* = 1.
4..N	Size Indication	✓	✕	0	0 = Region size of 128Mbytes.
N..31	Base Address	✓	✓	0	Loaded at boot time to set offset of the memory space for aperture two.

Notes: The **BaseAddress2** Register contains the R5 aperture 2 memory offset, and defines the size and type of the region. Typically the aperture is configured for 128Mbytes.

CFGBuilt In Self-Test

Name	Type	Offset	Format
CFGBIST	Configuration	0x0F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...7	BIST	✓	✕	0x00	00 = BIST is unsupported by R5 over the PCI interface

Notes: Optional register used for control and status of Built-In Self Test (BIST).

CFGCacheLine

Name	Type	Offset	Format
CFGCacl eLine	Configuration	0x0C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Cache Line Size	✓	✕	0x00	00= Cache line size (not supported on R5)

Notes: This register specifies the cache line size in units of 32 bit words. It is only implemented for PCI bus masters that use the "memory write and invalidate" command. R5 does not use this command.

CFGCapID

Name	Type	Offset	Format
CFGCap' D	Configuration	0x040	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Capability ID	✓	✕	see desc.	Configured by AGP Capable 0x00 when AGP Capable = 0 0x02 when AGP Capable = 1

Notes: This register specifies that the device has AGP capability. AGP Capable is a term used to express the logical OR of AGP1XCapable with AGP2XCapable with AGP4XCapable

CFGCapPtr

Name	Type	Offset	Format
CFGCapPtr	Configuration	0x34	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Capability Ptr	✓	✗	0x4C	Pointer to Power Management capability, address 0x4C.
8..31	Reserved	✗	✗	0	

Notes: This register is an eight bit register used to provide an offset into the configuration space for the first item in a capabilities list. It is used to point to the Power Management Capability that starts at offset 0x48

CFGCardBus

Name	Type	Offset	Format
CFGCardBus	Configuration	0x28	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	CardBus CIS Pointer	✗	✗	0	0x0000.0000 = Not implemented

Notes: This register is optional and not supported in R5.

CFGClassCode[BaseClass]

Name	Type	Offset	Format
CFGClas Code[BaseClass]	Configuration	0x0B	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..7	BaseClass	✔	✕	configured	Upper byte of ClassCode register, determines function type

Notes: The **ClassCode** register is read-only, and is used to identify the generic function of the R5. The register is best viewed as three byte-sized sub-registers, detailed below The reset value of this register is determined by the *BaseClassZero* bit in the **ChipConfig** register, and also by whether fixed VGA addressing has been enabled (see table below).

CFGClassCode[SubClass]

Name	Type	Offset	Format
CFGClas Code[SubClass]	Configuration	0x0A	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..7	SubClass	✔	✕	Configured	Middle byte of ClassCode register

Notes: The reset value of this register is determined by the *BaseClassZero* bit in the **ChipConfig** register, and also by whether fixed VGA addressing has been enabled.

CFGClassCode[InterfaceClass]

Name	Type	Offset	Format
CFGClas Code[Interface ClassCod :]	Configuration	0x09	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Interface	✔	✕	Configured	Lower byte of ClassCode register. 00 = VGA or Other Display Controller

Notes: The reset value of this register is determined by the *BaseClassZero* bit in the **ChipConfig** register, and also by whether fixed VGA addressing has been enabled (see table below).

If the *BaseClassZero* bit in the **ChipConfig** register is zero then the Base Class will be reported as 03h, since the R5 is a PCI display controller. If this bit is one then the Base Class will be reported as 00h (which will allow Windows to boot, even though it does not interpret display controller class codes correctly). Fixed VGA addressing will be enabled if the *VGAFixed* and *VGAEnable* bits in the **ChipConfig** register are both one:

Configuration Pins					
BaseClass Zero (Config Bit)	Fixed SVGA Addressing	Base Class	Sub Class	Device Class	Generic Function
0	Disabled	0x03	0x80	0x00	"Other" display controller
0	Enabled	0x03	0x00	0x00	VGA Compatible Controller
1	Disabled	0x00	0x00	0x00	Non-VGA Compatible Controller
1	Enabled	0x00	0x01	0x00	VGA Compatible Device

CFGCommand

Name	Type	Offset	Format
CFGCommand	Configuration	0x04	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	I/O Space Enable	✓	✗	0	0 = Disable Accesses 1 = Enable Accesses If fixed SVGA addressing is disabled, and indirect I/O region is disabled, this bit is 0
1	Memory Space Enable	✓	✓	0	0 = Disable Accesses 1 = Enable Accesses
2	Bus Master Enable	✓	✓	0	0 = Disable access 1 = Enable access
3	Special Cycle Enable	✓	✗	0	0 = GLINT R5 never responds to special cycle accesses
4	Memory Write and Invalidate Enable	✓	✗	0	0 = "Memory Write and Invalidate" is never generated.
5	SVGA Palette Snoop Enable	✓	✓	0	0 = Treat palette accesses like other SVGA accesses 1 = Enable SVGA Palette snooping
6	Parity Error Response enable	✓	✗	0	0 = GLINT R5 does not support parity error reporting
7	Address/Data stepping enable	✓	✗	0	0 = GLINT R5 does not perform stepping
8	SERR driver enable	✓	✗	0	0 = GLINT R5 does not support parity error reporting
9	Master Fast Back-to-Back Enable	✓	✗	0	0 = GLINT R5 master does not do fast back-to-back accesses 1 = Enable fast back-to-back accesses
10..15	Reserved	✓	✗	0	

Notes: The command register provides control over a device's ability to generate and respond to PCI cycles. It contains sufficient control bits to fulfill the GLINT R5's PCI functionality. Writing 0 to this register disconnects the device from the PCI for all except configuration accesses.
VGA palette snooping when enabled, monitors writes to 0x3C6, 0x3C8 and 0x3C9 and posts the results to Bypass FIFO (if space in the FIFO permits).

CFGDeviceID

Name	Type	Offset	Format
CFGDev ceID	Configuration	0x02	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..15	DeviceID	✓	✕		Device identification number: 0x0012 = 3Dlabs R5 device identification number If alternateDeviceID=1 then reset = 0x0013
16..31	Reserved	✕	✕		

CFGHeaderType

Name	Type	Offset	Format
CFGHea lerType	Configuration	0x0E	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..6	Header Type.	✓	✕	0x00	0x00 = reserved
7	Multifunction	✓	✕	0	PCI Definition: 0 = Single Function Device 1 = Multifunction device

Notes: The register identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space) and whether r not the device contains multiple functions. The value is always 0 for R5.

CFGIndirectAddress

Name	Type	Offset	Format
CFGIndirectAddress	Configuration *Control register*	0x0F8	Bitfield

Bits	Name	Read	Write	Reset	Description
0..26	Offset	✓	✓	0x000 00000	Offset within the region.
27, 28	Reserved	✓	✗	0	
29...31	Base Address Select	✓	✓	0	0 = Base Address 0 1 = Base Address 1 2 = Base Address 2 3= ROM region

Notes: The Reserved Base Address Select values can be written to or read from the register, but in this case, indirect accesses are treated as if to Base Address 0.

1. Reading the indirect trigger register **CFGIndirectTrigger** returns the value at the location pointed to by the indirect address register. Indirect data register **CFGIndirectData** is written to the location pointed to by the indirect address register **CFGIndirectAddress** when the indirect trigger register is written.

CFGIndirect Data

Name	Type	Offset	Format
CFGIndirectData	Configuration *Control register*	0xF4	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Data	✓	✓	0x000 00000	Data to be written indirectly

Notes:

1. This register is used to access regions 0 to 2 and the ROM region indirectly through the config space. The region to be accessed and the offset into that region are programmed into the **CFGIndirectAddress** register. Data written to the **CFGIndirectData** register will be written to the location pointed to by the **CFGIndirectAddress** register when the **CFGIndirectTrigger** register is written.
2. Reading the **CFGIndirectTrigger** register returns the value at the location pointed to by the **CFGIndirectAddress** register.

CFGIndirect Access Trigger

Name	Type	Offset	Format
CFGIndirectTrigger	Configuration	0xFC	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	Trigger	✓	✓	0x000 00000	

Notes: This register is used to trigger indirect accesses as specified by the indirect address and data registers, **CFGIndirectAddress** and **CFGIndirectData**

CFGIntLine

Name	Type	Offset	Format
CFGIntLine	Configuration	0x3C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Interrupt Line	✓	✓	0	Not read or written by the R5 device itself.

Notes: The Interrupt Line register in an 8-bit register used to communicate interrupt line routing information. It is available for use by device drivers and the operatiing system but is not used by R5.

CFGIntPin

Name	Type	Offset	Format
CFGIntPin	Configuration	0x3D	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Interrupt Pin	✓	✗	0x01	0x01 = R5 uses Interrupt pin A

Notes: The Interrupt Pin register specifies the interrupt line that GLINT R5 uses.

CFGLatTimer

Name	Type	Offset	Format
CFGLatTimer	Configuration	0x0D	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Latency Timer Count	✓	✓	0x00	Sets the max number of PCI Clock cycles for master burst accesses

Notes: This register specifies, in PCI bus clocks, the value of the latency timer for this PCI bus master

CFGMaxLat

Name	Type	Offset	Format
CFGMaxLat	Configuration	0x3F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...5	Maximum latency	✓	✗	0	00.0000b - always set to zero
6	Maximum Latency[6]	✓	✗		This is set to 1
7	Maximum Latency[7]	✓	✗		This is set to 1

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

CFGMinGrant

Name	Type	Offset	Format
CFGMinGrant	Configuration	0x3E	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0-5	Minimum grant[5:0]	✓	✕	0	00.0000b - always set to zero
6	MinimumGrant [6]	✓	✕		This is set to 1
7	MinimumGrant [7]	✓	✕		This is set to 1

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

CFGNextPtr

Name	Type	Offset	Format
CFGNex Ptr	Configuration	0x041	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Next Ptr	✓	✕	0	0x00 = no further capabilities in list

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

CFGPower Management Capabilities

Name	Type	Offset	Format
CFGPM(	Configuration	0x4E	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Version	✓	✗	1	1=complies with Rev 1.0 of the PCI Power Management Interface Spec.
3	PME Clock	✓	✗	0	0=PME# not supported in any state
4	Aux Power source	✓	✗	0	0 = PME# is not supported in D3(cold)
5	DSI	✓	✗	1	1 = R5 requires special initialization following transition to the D0 uninitialized state
6...8	Reserved	✓	✗	0	
9	D1_Support	✓	✗	1	1 = D1 power level is supported
10	D2_Support	✓	✗	0	0 = D2 power level is not supported
11...15	PME_Support	✓	✗	0	0 = PME# signal is not asserted in any power state

Notes: The power management registers support states D0, D1 and D3. When the device is in any state other than D0 disable decoding of slave I/O and memory accesses. Configuration accesses must be decoded at all times. This requirement conforms with PCI Power Management spec version 1.1

CFGPM Capability ID

Name	Type	Offset	Format
CFGPM(apID	Configuration	0x4C	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Power Management Capability ID	✓	✗	0x1	0x01 = Power Management Capability
8..31	reserved	✓	✗	0	

Notes: This register specifies that the device has Power Management capability

CFGPower Management Control/Status

Name	Type	Offset	Format
CFGPMCSR	Configuration	0x50	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..1	PowerState	✓	✓	00b	Valid values are 0,1 and 3. If 2 is written to the register, the write is discarded (D2 is not supported) 0 = D0 1 = D1 (This drives the "Low Power" bit internally) 3 = D3(hot)
2..7	Reserved	✓	✕	0	
8	PME_EN	✓	✕	0	0 = PME# signal is not asserted in D3(cold)
9..12	Data_Select	✓	✕	0	0 = Data register not supported
13, 14	Data_scale	✓	✕	0	0 = Data register not supported
15	PME_Status	✓	✕	0	0 = PME# signal is not asserted in D3(cold)

Notes:

CFGPMCSR_BSE

Name	Type	Offset	Format
CFGPMCSR_BSE	Configuration	0x52	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Power Management Bridge support	✓	✕	0	0x00 = R5 is not a bridge.

Notes: This register specifies the Power Management PCI-PCI bridge support

CFGPMData

Name	Type	Offset	Format
CFGPMData	Configuration	0x53	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...7	PMData	✓	✗	0x00	0x00 = The register is reserved but not implemented

Notes: This register is the optional Power Management Data register

CFGPMNextPtr

Name	Type	Offset	Format
CFGPMNextPtr	Configuration	0x4D	

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Next Ptr	✓	✗	See Desc.	0 = no further capabilities in list if AGP Capable = 0 0x40 = point to AGP Capability if AGP Capable = 1
8...31	Reserved	✓	✗	0	

Notes: This register specifies the device has next capability item. AGP Capable is a term used to express the logical OR of AGP1XCapable with AGP2XCapable with AGP4XCapable.

CFGRevisionID

Name	Type	Offset	Format
CFGRevisionID	Configuration	0x08	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	RevisionID	✓	✕	0x1	Revision Identification Number - 0x01 = Revision R0

Notes:

CFGRom Base Address

Name	Type	Offset	Format
CFGRom Addr	Configuration	0x30	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Access Decode Enable	✓	✓	0	0= Expansion ROM accesses disabled 1= Expansion ROM accesses enabled
1..10	Reserved	✓	✕	0.0000.0000b	PCI Reserved register bits
11..15	Size Indication	✓	✕	0.0000b	0 = Indicates that Expansion ROM must be mapped into 64Kbytes.
16..31	Expansion Rom Base Address	✓	✓	0	Loaded at boot time to set base adress of the expansion ROM.

Notes: The expansion ROM base register is the base address offset for the expansion ROM.

CFGStatus

Name	Type	Offset	Format
CFGStatus	Configuration	0x06	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...3	Reserved	✓	✕	0000b	
4	Cap_List	✓	✕	0x1	1 = R5 can accept additional capabilities beyond PCI 2.1. These are power management and AGP (if *AGPCapable* is set in **CFGCapID**)
5	66MHz Capable	✓	✕	Configured	0 = R5 is 33MHz capable only 1 = R5 is 66MHz capable
6	UDF Supported	✓	✕	0	0 = R5 does not support user-definable configurations
7	Fast back-to-back capable	✓	✕	1	1 = R5 can accept fast back-to-back PCI transactions
8	Data Parity Error Detected	✓	✕	0	0 = R5 does not implement parity checking
9...10	DEVSEL Timing	✓	✕	01b	1 = R5 asserts DEVSEL# at medium speed
11	Signaled Target Abort	✓	✕	0	0 = R5 never signals Target-Abort
12	Received Target Abort	✓	✓	0	This bit is set by the R5 bus master whenever its transaction is terminated with Target-Abort
13	Received Master Abort	✓	✓	0	This bit is set by the R5 bus master whenever its transaction is terminated with Master-Abort
14	Signalled System Error	✓	✕	0	0 = R5 never asserts a system error
15	Detected Parity Error	✓	✕	0	0 = R5 does not implement parity checking

Notes: Writes to this register causes bits to be reset, but not set. A bit is reset whenever the register is loaded with the corresponding bit position set to one. AGP Capable is a term used to express the logical OR of AGP1XCapable with AGP2XCapable with AGP4XCapable

CFGSubsystemId

Name	Type	Offset	Format
CFGSubsystemId	Configuration	0x02E	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..15	SubsystemId	✕	✓ once	see text	See CFGSubsystemVendorID

Notes: This register is used to identify the add-in board on which the R5 resides. The reset value is the Device ID selected by the *AlternateDeviceID* configuration bit. The register can subsequently be written to only once, and further writes are discarded. If the SubsystemFromROM configuration bit is set, the register is loaded with the contents of ROM locations FFFE (low) and FFFF (high byte) after reset, and subsequent writes are discarded.

CFGSubsystemVendorId

Name	Type	Offset	Format
CFGSubsystemVendorId	Configuration	0x02C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..15	Subsystem VendorID	✕	✓ once	see text	

Notes: This register is used to identify the vendor of the add-in board on which the R5 resides. The rest value is the R5 Vendor ID. The register can subsequently be written to only once, and further writes are discarded. If the SubsystemFromROM configuration bit is set, the register is loaded with the contents of ROM locations FFFC (low) and FFFD (high byte) after reset, and subsequent writes are discarded.

CFGVendorID

Name	Type	Offset	Format
CFGVendorID	Configuration *Control register*	0x00	Integer

Bits	Name	Read	Write	Reset	Description
0..15	Vendor ID	✓	✕	0x3D3D	3Dlabs Company Code
16..31					See CFGDeviceID

Notes: Vendor Identification Number, 3D3D = **3D**Labs Company code

4.3 Region 0 Control Status and Test (0x0000-0x01FF)

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

A number of Control Status Registers are implemented within the PCI Bus Interface, including registers for interrupt and error handling, test, reporting graphics processor FIFO status, and DMA control. Mode control registers are provided for Memory Apertures One and Two.

Region Zero also contains Memory and Video Control registers, which are accessed using the bypass interface, and RAMDAC and VGA Control registers, which are accessed using their own particular interfaces.

Core graphics registers can be accessed either via the region 0 FIFO (0000.2000 -> 0000.2FFF) or memory-mapped registers (0000.8000 -> 0000.FFFF). This is discussed in more detail in the *GLINT R5 Programmer's Guide,* volume I

AGPControl

Name	Type	Offset	Format
AGPCon rol	Control Status	0x078	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0..2	Reserved	✓	✕	0		
3	AGP Long Read Disable	✓	✓	0	0 = AGP Long Read Requests may be generated.	1 = AGP Long Read Requests disabled.
4	Reserved	✓	✕	0		
5	AGP Data Fifo throttle	✓	✓	0	0 = RBF# throttle start of data transfer for low priority reads.	1 = Only request data when space is available in AGP data fifo to start receiving the burst (RBF# never asserted)
6	AGP High Priority	✓	✓	0	0 = Use AGP Low Priority reads.	1 = Use AGP High Priority reads
7..31	Reserved	✓	✕	0		

Notes: The AGP control register sets up the AGP master.

ApertureOne
ApertureTwo

Name	Type	Offset	Format
ApertureOne	Control Status	0x50	Bitfield
ApertureTwo	Control Status	0x58	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0..7	Reserved	✓	✕	0		
8	VGA Access	✓	✓	0	0 = Address memory controller directly.	1 = Address memory through SVGA subsystem.
9	ROM Access	✓	✓	0	0 = Use this aperture to access memory (SVGA or direct).	1 = Use this aperture to access the Expansion ROM.
10..31	Reserved	✓	✕	0		

Notes: Two memory apertures are provided, each being a PCI region with a fixed size of 64 MBytes. A variety of different access modes are possible - these are now controlled in the Bypass controller registers. The **ApertureOne** and **ApertureTwo** registers allow the Apertures to be used to access the VGA or ROM instead of the memory controller. When the *VGAAccess* bit in either of the **ApertureOne** or **ApertureTwo** registers is set, then all accesses to the relevant aperture will be forwarded to the VGA Unit, rather than directly to the memory controller.

The address for the VGA Unit is formed from bits 16 down to 2 of the incoming bus address, as for Fixed Memory Addresses. This results in the 128 KByte VGA memory space being aliased within the 16 MByte total region size. No byte swapping or other data formatting is performed when accessing the VGA memory in this manner.

AutoCalCount

Name	Type	Offset	Format
AutoCalCount	Control Status	0x00F8	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...12	AutoCalCount	✓	✗	0x007F.FFFF	Fixed part, read only
13....31	AutoCalCount	✓	✓		Programmable part

Notes: Controls the Auto Calibration period for the AGP 4X. – number of clocks between calibrations . In order to avoid a zero count, the bottom 12 bits are always set

ChipConfig

Name	Type	Offset	Format
ChipConfig	Control Status	0x70	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	BaseClassZero	✓	✓	X	0 = Use the correct PCI Base Class Code 1 = Force PCI Base Class Code to be zero
1	VGAEnable	✓	✓	X	0 = Disable internal SVGA subsystem 1 = Enable internal SVGA subsystem
2	VGAFixed	✓	✓	X	0 = Disable SVGA fixed address decoding 1 = Enable SVGA fixed address decoding
3	VGANoAlias	✓	✓	X	0 = decode only 10 bits of VGA I/O Addresses 1 = decode all 32 bits of VGA I/O Addresses
4	AGP4G	✓	✓	X	0 = only 32-bit AGP addresses 1 = AGP can generate >4GB addresses
5	RetryDisable	✓	✓	X	0 = Enable PCI Retry using "Disconnect-Without-Data" 1 = Disable PCI Retry using "Disconnect-Without-Data"
6	Reserved	✓	✗	0	
7	ShortReset	✓	✓	X	0 = Generate normal "AReset" pulse to rest of the chip 1 = Generate short "AReset" pulse (BusReset+ 64 clocks)
8	SBA Capable	✓	✓	X	0 = AGP sideband Addressing Disable 1 = AGP sideband Addressing Enable

9	AGP 1X Capable	✓	✓	X	0 = Not AGP 1X Capable 1 = AGP 1X Capable
10	AGP 2X Capable	✓	✓	X	0 = Not 2X Capable 1 = 2X Capable
11	AGP 4X Capable	✓	✓	X	0 = Not 4X Capable 1 = 4X Capable
12	Subsystem FromRom	✓	✓	X	0 = Leave subsystem registers with reset values 1 = Load subsystem registers from ROM after reset
13	reserved	✓	✕	0	
14	WC Enable	✓	✓	config ured	0 = Upper half of region zero is a byte swapped version of lower half 1 = Upper half of region zero is flagged as a Write combined version of the lower half
15	Prefetch Enable	✓	✓	Config ured	0 = Regions 1 and 2 marked as not prefetchable 1 = Regions 1 and 2 marked as prefetchable
16	Alternate Device IDEnable	✓	✕	Config ured	0=Device ID=0x12 1=Device ID=0x13
17	AutoCal Enable	✓	✕	Config ured	0=Disable 1=Enable
18..20	Max Function ID	✓	✕	Config ured	reserved
21...23	ChipID	✓	✕	Config ured	The hardcoded ID of this chip in a multichip adapter
24	Common Strobe	✓	✕	Config ured	0 = don't use common memory strobes 1 = use common memory strobes
25	Internal Strobes	✓	✕	Config ured	0 = don't use internal memory strobes 1 = use internal memory strobes
26	DDR Enable	✓	✕	Config ured	0 = DDR memory not fitted 1 = DDR memory fitted
27	reserved	✓	✕		
28..31	Mask rev	✓	✕	See Desc.	Value gives the Mask Revision. The initial revision is 0x0.

Notes: Most of the sampled values from the configuration pins are loaded into the ChipConfig register on the trailing edge of reset. This register can then be read back over the PCI bus, to allow the host to determine how the GLINT R5 chip has been configured, and to modify various fields of the configuration if required.

ControlDMAAddress

Name	Type	Offset	Format
ControlDMAAddress	Control Status	0x28	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	Control DMA Start Address	✓	✓	0	PCI start address for PCI master read transfer to the graphics processor input fifo.

Notes: When using the GPIn FIFO DMA controller to load the graphics processor, the Control DMA Start Address register should be loaded with the PCI address of the first word in the buffer to be transferred. Writing to the Control DMA Start Address register loads the address into the Control DMA address counter. Once a DMA has been set off, the next Control DMA start address may be loaded. A read of this register returns the last start value loaded even if the DMA is already underway.

ControlDMAControl

Name	Type	Offset	Format
ControlDMAControl	Control Status	0x60	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	ControlDMA Byte Swap Control	✓	✓	0	This field should only be changed when the ControlDMA controller is idle: 0 = Standard. 1 = Byte Swapped
1	ControlDMA using AGP	✓	✓	0	0 = DMA uses PCI Master 1 = DMA uses AGP Master
2	Sideband	✓	✓	0	0 = Don't use sideband command pins 1 = Use sideband command pins
3	Alignment	✓	✓	0	0 = use 4-byte aligned address 1 = use 64-byte aligned address
4...6	Burst size	✓	✓	0	Dwords: 0, 1, 2 = 4 dwords 3 = 8 dwords 4 = 16 dwords 5 = 32 dwords 6, 7 = 64 dwords
2..31	Reserved	✓	✕	0	

Notes: The DMA control register sets up the data transfer modes for the DMA controller. Data transfer can be set to 'byte swapped' for big endian hosts.

ControlDMACount

Name	Type	Offset	Format
ControlDMACount	Control Status *Control register*	0x30	Integer

Bits	Name	Read	Write	Reset	Description
0..23	Control DMA Count	✓	✓	0	Number of words to be transferred in the DMA operation.
16..31	Reserved	✓	✗	0x0000	

Notes:
1. Loading a word greater than zero triggers the DMA. The value read back indicates the current number of words left to transfer. If written to while non-zero and mastering is enabled, the write is discarded and the DMACountOverwrite error flag is set.
2. Mastering is enabled if:
   - *ControlDMAUseAGP* = 0 and PCI Bus Master Enabled, or
   - *ControlDMAUseAGP* = 1 and AGP Master is Enabled.
   See **DMAControlRegister**.
3. When using the GPIn FIFO DMA controller to load the graphics processor, the Control DMA Start Address register should be loaded with the PCI address of the first word in the buffer to be transferred. Writing to the Control DMA Start Address register loads the address into the Control DMA address counter. Once a DMA has been set off, the next Control DMA start address may be loaded.

CoreErrorFlagsOne

Name	Type	Offset	Format
CoreErrc ·FlagsOne	Control Status	0x00B0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..15	Command Packet Error 15-0	✓	To clear	0x0000	Non-packet data before the packet is complete Bit 15 is error 15, down to bit 0 is error 0. 0 = No error 1 = Error outstanding
16..31	Command BB TestError15-0	✓	To clear	0x0000	Incorrect use of bounding box test. Bit 15 is error 15, down to bit 0 is error 0. 0 = No error 1 = Error outstanding

Notes: For error flags generated by the core pipeline. If an error occurs the Error Flag bit is set in **IntFlags** If the error interrupt is enabled in **IntFlags** then an interrupt is generated. Reset flag bits by writing to the register with the corresponding bit set to 1. Writing 1 to a bit currently set to 0 leaves the bit set to 0.

CoreErrorFlagsTwo

Name	Type	Offset	Format
CoreErrc ·FlagsTwo	Control Status	0x00C0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Vertex Data Error	✓	✓	0	Incomplete Vertex. 0 = No error 1 = Error outstanding
1	Index Cache Error	✓	✓	0	Attempt to Cache uncachable data 0 = No error 1 = Error outstanding
2	Command Nesting Error	✓	✓	0	Attempt to nest DMA more than 2 levels deep 0 = No error 1 = Error outstanding
3	Command WC Error	✓	✓	0	Incorrect use of write combining 0 = No error 1 = Error outstanding
4..31	reserved	✓	✗	0x0000	

Notes: For error flags generated by the core pipeline. If an error occurs the Error Flag bit is set in **IntFlags** If the error interrupt is enabled in **IntFlags** then an interrupt is generated. Reset flag bits by writing to the register with the corresponding bit set to 1. Writing 1 to a bit currently set to 0 leaves the bit set to 0.

ErrorFlags

Name	Type	Offset	Format
ErrorFlags	Control Status	0x0038	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Input FIFO Error Flag	✓	✓	0	Flag set on write to full input FIFO. 0 = No error. 1 = Error outstanding.
1	Output FIFO Error Flag	✓	✓	0	Flag set on read from empty output FIFO. 0 = No error. 1 = Error outstanding.
2	Reserved	✓	✗	0	
3	Control DMA Error Flag	✓	✓	0	Flag set for direct or register access to input FIFO while DMA is in progress (i.e. when the Control DMACount register is not zero). 0 = No error. 1 = Error outstanding.
4	Video Fifo Underflow Error Flag	✓	✓	0	Flag set when video FIFO underflows 0 = No error 1 = Error outstanding
5	Reserved	✓	✓	0	
6	Reserved	✓	✓	0	
7	PCI Master Error Flag	✓	✓	0	Flag set when either Master abort or Target abort occurs while PCI Master access in progress. - The CFGStatus register can be read to determine the type of error. 0 = No error. 1 = Error outstanding.
8	GPOutDMA Error Flag	✓	✓	0	Flag set for slave access to output FIFO while DMA is in progress 0 = No error. 1 = Error outstanding.
9	Control DMA Count Overwrite Error Flag	✓	✓	0	Flag set if an attempt is made to write the Control DMACount register when it is not zero. 0 = No error. 1 = Error outstanding.
10	GPOutDMA Feedback Error Flag	✓	✓	0	Flag set if a feedback error occurs. 0 = No error. 1 = Error outstanding.
11	Texture DMA Error Flag	✓	✓	0	
12	Texture DMA FIFO Error Flag	✓	✓	0	
13	reserved	✓	✗	0	

14	Matrix Unit Error	✓	✓	0	Flag set if Matrix Unit error occurs 0 = No error 1 = Error Outstanding
15	VideoOverlay Underflow Error	✓	✓	0	Flag set if an underflow error occurs in the Video Overlay unit: 0 = No error 1 = Error Outstanding
16..31	Reserved	✓	✕	0	All bits 0

Notes: The Error Flags register shows which errors are outstanding in GLINT R5. Flag bits are reset by writing to this register with the corresponding bit set to a one. Flags at positions where the bits are set to zero will be unaffected by the write.

FIFODiscon

Name	Type	Offset	Format
FIFODis on	Control Status *Control register*	0x68	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Input FIFO Disconnect Enable	✓	✓	0	0 = Disabled 1 = Enabled
1	Output FIFO Disconnect Enable	✓	✓	0	0 = Disabled 1 = Enabled
2	Texture FIFO Disconnect Enable	✓	✓	0	0 = Disabled 1 = Enabled
3	FBSync FIFO Disconnect	✓	✓	0	0 = Disabled 1 = Enabled
4..31	Reserved	✓	✕	0	

Notes: The **FIFODiscon** register enables the input and output FIFO disconnect signals, which drive two physical pins. Disconnects are disabled at reset. It also allows protocol disconnects to be enabled for the Texture FIFO.

GPOutDMAAddress

Name	Type	Offset	Format
GPOutDMAAddress	Control Status	0x080	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	GPOutDMAAddress	✓	to clear	0x0000	Next address to be issued to the DMA Arbiter.

Notes: The **GPOutDMA** Address register can be used to monitor the progress of the GPOutDMA controller. It returns the next address to be issued to the DMA arbiter. It is cleared by writing any value to it, which allows software to determine when DMA has started.

HostTextureAddress

Name	Type	Offset	Format
HostTextureAddress	Control Status	0x0100	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	0	.
4..31	HostTexture Address	✓	✓	X	

Notes: Used in "Slave Download Mode" to supply the address of the first word of a texture

InFIFOSpace

Name	Type	Offset	Format
InFIFOSpace	Control Status	0x0018	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	Input FIFO Space	✓	✗	0x0000.0080	The number of empty words in the input FIFO. This number of words can be updated before checking InFIFOSpace again.

Notes: The **InFIFOSpace** register shows the number of words that can currently be written to the input FIFO. This register can be read at any time. If the DMA controller for the FIFO is in use, the value read is a snapshot of the current FIFO status. (This value is returned from the graphics processor core and should be zero extended to 32 bits.)

IntEnable

Name	Type	Offset	Format
IntEnable	Control Status	0x08	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Control DMA Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
1	Sync Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
2	Reserved	✓	✗	0	
3	Error Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
4	Vertical Retrace Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable Interrupt .
5	Scanline Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable Interrupt
6	Texture DownLoad Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt

7	Bypass DMA Read Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
8	Reserved	✓	✓	0	
9	Reserved	✓	✓	0	
10	VS Serial Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
11	VidDDC Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
12	Flat Panel Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
13	Bypass DMA Write Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
14	Command Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
15	Reserved	✓	✕	0	Reserved
16	Bypass Read DMA Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
17	Bypass Write DMA Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
18	GP Out Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
19	GP In Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
20	Input FIFO Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.

21	Memory Unit Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt 1 = Enable interrupt
22	Rasterizer Page Fault Interrupt enable	✓	✓	0	0 = Disable interrupt 1 = Enable interrupt
23	Texture DMA Complete Interrupt enable	✓	✓	0	0 = Disable interrupt 1 = Enable interrupt
24..31	Reserved	✓	✕	0	Read Only, all bits 0.

Notes: The **IntEnable** register selects which internal conditions are permitted to generate a bus interrupt. At reset all interrupt sources are disabled

IntFlags

Name	Type	Offset	Format
IntFlags	Control Status	0x10	Bitfield

Control register

Bits	Flag Name	Read	Write	Reset	Description
0	Control DMA	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding.
1	Sync	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding
2	Reserved	✓	✗	0	
3	Error	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding.
4	Vertical Retrace	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding.
5	Scanline	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding
6	Texture Download	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding
7	Bypass Read DMA	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding.
8	Reserved	✓	✓	0	
9	Reserved	✓	✓	0	
10	VS Serial	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding
11	VidDDC	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding
12	Flat Panel Flag	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding.
13	Bypass Write DMA	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding.
14	HostIn Command DMA	✓	✓	0	0 = No interrupt. 1 = Interrupt outstanding
15	Reserved	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
16	Bypass Read DMA Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
17	Bypass Write DMA Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
18	GP Out Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
19	GP In Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
20	Input FIFO Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
21	Memory Unit Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding

22	Rasterizer Page Fault	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
23	Texture DMA Complete Interrupt	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
24..30	Reserved	✓	✕	0	
31	VGA Interrupt Line	✓	✕	0	0 = No interrupt. 1 = Interrupt asserted.

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

LastTextureReadAddress

Name	Type	Offset	Format
LastTextureReadAddress	Control Status	0x0120	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0.31	TexDMA Address	✓	✕	X	

Notes: Found in the TexturePage unit, this register returns the address of the last data returned to the Texture Read unit.

LogicalTexturePage

Name	Type	Offset	Format
LogicalTexturePage	Control Status	0x118	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..15	LogicalTexture Page	✓	✕	X	
16..31	Reserved	✓	✕	0	

Notes: Used with Slave Download Mode to complete the Texture FIFO protocol..

Miscellaneous Control

Name	Type	Offset	Format
MiscCtl	Control Status	0x0048	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	DMA Context Save	✓	✓	0	Connects to command, vertex and rectangle DMA units: 0 = Don't save context 1 = Save context
1	PMSlave DecodeDisable	✓	✓	0	Controls whether slave decodes occur in a power state other than D0: 0 = Decode slave accesses in all power states 1 = Decode slave accesses only in state D0
2	PMMaster Disable	✓	✓	0	Controls whether bus master accesses can occur in a power state other than D0: 0 = Allow PCI & AGP mastering in all power states 1 = Allow PCI & AGP mastering only in state D0
3..16	Reserved	✓	✕	0x000 0.0000	

Notes:

OutFIFOWords

Name	Type	Offset	Format
OutFIFC Words	Control Status *Control register*	0x0020	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Output FIFO Words	✓	✕	0x000 0.0000	The number of valid words in the output FIFO. This number of words can be read before checking "OutFIFOWords" again.

Notes: The **OutFIFOWords** register shows the number of words currently in the output FIFO. This register can be read at any time.

PCIAbortAddress

Name	Type	Offset	Format
PCIAbor Address	Control Status *Control register*	0x0098	Integer

Bits	Name	Read	Write	Reset	Description
0..31	PCIAbort Address	✓	✕	0	

Notes: The **PCIAbortAddress** register contains the first PCI Address issued by the PCI Master to cause an Abort.
N.B. the following 32 bits (offset 0x00A0) are also reserved for future expansion of **PCIAbortAddress** to 64 bits.

PCIAbortStatus

Name	Type	Offset	Format
PCIAbor Status	Control Status	0x090	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..22	ReadSource	✓	×	0	The Read source in the DMA Arbiter that caused the Abort.
23	ReadStatus	✓	×	0	0 = No read abort 1 = Read abort
24...30	WriteSource	✓	×	0	The Write source in the DMA Arbiter which caused the Abort.
31	WriteStatus	✓	×	0	0 = No Write abort 1 = Write abort.

Notes: The **PCIAbortStatus** register reports whether a PCI Master read or write operation has caused an abort (either a Master Abort or Target Abort.) . The **PCIAbortAddress** register can be read to determine the first PCI Address issued which caused an abort. The register can be cleared by writing any value to it.

PCIFeedbackCount

Name	Type	Offset	Format
PCIFeedl ackCount	Control Status	0x0088	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	PCI Feedback Count	✓	×	0x0000	Number of words that have been transferred in the DMA operation.

Notes: The **PCIFeedbackCount** register can be read to monitor the progress of a Feedback DMA. The value returned is the number of double words transferred in the current DMA

PCIPLLStatus

Name	Type	Offset	Format
PCIPLLS tatus	Control Status	0x00F0	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..8	PCIPLLSetup	✓	✓	tbd	Provides 9 bits of setup for the deskew PLL.
9..11	PCIPLL PostScale	✓	✓	0x1	Divide by 2
12	PCIPLL Enable	✓	✓	0x1	reset = 1
13..30	Reserved	✓	✗	0	0
31	Deskew PLL Lock	✓	✗	0	Deskew lock

Notes: The PCIPLLStatus register controls the PCI deskew PLL status bits.

PclkProfCount0

Name	Type	Offset	Format
PclkProf ount0	Control Status	0x0260	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	PclkProf Count0	✓	✓	0x0	

Notes: Counts the evet flags determined by A0 and B0 inverts and masks..

PclkProfCount1

Name	Type	Offset	Format
PclkProf ount1	Control Status	0x0288	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	PclkProf Count1	✓	✓	0x0	

Notes: Counts the evet flags determined by A0 and B0 inverts and masks..

PclkProfInvertA0

Name	Type	Offset	Format
PclkProfInvertA0	Control Status	0x0240	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	PclkProfInvertA0	✓	✓	0x0	

Notes: Bitwise inverts the "A" set of event flags to profiling counter 0. Flags and bit positions are defined in **PClkProfMaskA0** below.

PclkProfInvertA1

Name	Type	Offset	Format
PclkProfInvertA1	Control Status	0x0270	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	PclkProfInvertA1	✓	✓	0x0	

Notes: Bitwise inverts the "A" set of event flags to profiling counter 1.

PclkProfInvertB0

Name	Type	Offset	Format
PclkProfInvertB0	Control Status	0x0248	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	PclkProfInvertB0	✓	✓	0x0000.0000	

Notes: Bitwise inverts the "B" set of event flags to profiling counter 0. Flags and bit positions are defined in **PClkProfMaskA0** below.

PclkProfInvertB1

Name	Type	Offset	Format
PclkProfInvertB1	Control Status	0x0270	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	PclkProfInvertB1	✓	✓	0x0	

Notes: Bitwise inverts the "B" set of event flags to profiling counter 1.

PclkProfMaskA0

Name	Type	Offset	Format
PclkProfMaskA0	Control Status	0x0250	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Always	✓	✓	X	
1	SlaveWriteAddr	✓	✓	X	
2	SlaveWritebusy	✓	✓	X	
3	SlaveWriteWait	✓	✓	X	
4	SlaveWriteData	✓	✓	X	
5	SlaveReadAddr	✓	✓	X	
6	SlaveReadBusy	✓	✓	X	
7	SlaveReadWait	✓	✓	X	
8	SlaveReadData	✓	✓	X	
9	SlaveTgtStop	✓	✓	X	
10	MasterRequest	✓	✓	X	
11	MasterWrite Addr	✓	✓	X	
12	MasterWrite Wait	✓	✓	X	
13	MasterWrite Data	✓	✓	X	
14	MasterRead Addr	✓	✓	X	
15	MasterRead Wait	✓	✓	X	
16	MasterRead Data	✓	✓	X	
17	MasterTgt Stop	✓	✓	X	
18	AgpData BusBusy	✓	✓	X	
19	AgpData MasDelay	✓	✓	X	
20	AgpData Transfer	✓	✓	X	
21	AgpArb FifoFull	✓	✓	X	
22	AgpCtrl FifoFull	✓	✓	X	
23	AgpAddr FifoFull	✓	✓	X	

24	AgpData FifoEmpty	✓	✓	X	
25	MasInCtrl FifoFull	✓	✓	X	
26	MasInData FifoEmpty	✓	✓	X	
27	MasOutCtrl FifoFull	✓	✓	X	
28	MasOutData FifoFull	✓	✓	X	
29	Reserved	✓	✗	X	
30	Mode	✓	✓	X	0=AND 1=OR
31	Reserved	✓	✗	X	

Notes: Masks the "A" set of event flags to profile counter 0 (1=masked).

PclkProfMaskA1

Name	Type	Offset	Format
PclkProfl IaskA1	Control Status	0x0278	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Always	✓	✓	X	
1	SlaveWriteAddr	✓	✓	X	
2	SlaveWritebusy	✓	✓	X	
3	SlaveWriteWait	✓	✓	X	
4	SlaveWriteData	✓	✓	X	
5	SlaveReadAddr	✓	✓	X	
6	SlaveReadBusy	✓	✓	X	
7	SlaveReadWait	✓	✓	X	
8	SlaveReadData	✓	✓	X	
9	SlaveTgtStop	✓	✓	X	
10	MasterRequest	✓	✓	X	
11	MasterWrite Addr	✓	✓	X	
12	MasterWrite Wait	✓	✓	X	
13	MasterWrite Data	✓	✓	X	
14	MasterRead Addr	✓	✓	X	
15	MasterRead Wait	✓	✓	X	
16	MasterRead Data	✓	✓	X	
17	MasterTgt Stop	✓	✓	X	
18	AgpData BusBusy	✓	✓	X	
19	AgpData MasDelay	✓	✓	X	
20	AgpData Transfer	✓	✓	X	
21	AgpArb FifoFull	✓	✓	X	
22	AgpCtrl FifoFull	✓	✓	X	
23	AgpAddr FifoFull	✓	✓	X	

24	AgpData FifoEmpty	✓	✓	X	
25	MasInCtrl FifoFull	✓	✓	X	
26	MasInData FifoEmpty	✓	✓	X	
27	MasOutCtrl FifoFull	✓	✓	X	
28	MasOutData FifoFull	✓	✓	X	
29	Reserved	✓	✗	0	
30	Mode	✓	✓	X	0=AND 1=OR
31	Reserved	✓	✗	0	

Notes: Masks the "A" set of event flags to profile counter 1 (1=masked).

PclkProfMaskB0

Name	Type	Offset	Format
PclkProf! IaskA0	Control Status	0x258	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	ByInFifo Full	✓	✓	X	
1	ByInFifoEmpty	✓	✓	X	
2	ByDMA Write Running	✓	✓	X	
3	ByDMA WriteEmpty	✓	✓	X	
4	ByDMARead Running	✓	✓	X	
5	ByDMA ReadFull	✓	✓	X	
6	GpIn Fifo Full	✓	✓	X	
7	GpIn FifoEmpty	✓	✓	X	
8	GpInDMA Running	✓	✓	X	
9	GpInDMA Ctrl Full	✓	✓	X	
10	GpInDMA DataFull	✓	✓	X	
11	TXDMA Running	✓	✓	X	
12	TXDMA Ctrl Full	✓	✓	X	
13	TXDMA DataFull	✓	✓	X	
14…17	Reserved	✓	✗	0	
18	MemDMA Running	✓	✓	X	
19	MemDMA CtrlFull	✓	✓	X	
20	MemDMA Data Empty	✓	✓	X	
21	GpOut FifoFull	✓	✓	X	
22	GpOut FifoEmpty	✓	✓	X	
23	GpOutDMA Running	✓	✓	X	

24	GpOutDMA Ctrl Empty	✓	✓	X	
25	GpOutDMA CtrlFull	✓	✓	X	
26	GpOutDMA DataEmpty	✓	✓	X	
27	GpOutDMA DataFull	✓	✓	X	
28...31	Reserved	✓	✗	0	

Notes: Masks the "B" set of event flags to profile counter 0 (1=masked).

PclkProfMaskB1

Name	Type	Offset	Format
PclkProfl IaskB1	Control Status	0x0258	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	ByInFifo Full	✓	✓	X	
1	ByInFifoEmpty	✓	✓	X	
2	ByDMA Write Running	✓	✓	X	
3	ByDMA WriteEmpty	✓	✓	X	
4	ByDMARead Running	✓	✓	X	
5	ByDMA ReadFull	✓	✓	X	
6	GpIn Fifo Full	✓	✓	X	
7	GpIn FifoEmpty	✓	✓	X	
8	GpInDMA Running	✓	✓	X	
9	GpInDMA Ctrl Full	✓	✓	X	
10	GpInDMA DataFull	✓	✓	X	
11	TXDMA Running	✓	✓	X	
12	TXDMA Ctrl Full	✓	✓	X	
13	TXDMA DataFull	✓	✓	X	
14…17	reserved	✓	✓	0	
18	MemDMA Running	✓	✓	X	
19	MemDMA CtrlFull	✓	✓	X	
20	MemDMA Data Empty	✓	✓	X	
21	GpOut FifoFull	✓	✓	X	
22	GpOut FifoEmpty	✓	✓	X	
23	GpOutDMA Running	✓	✓	X	

24	GpOutDMA Ctrl Empty	✓	✓	X	
25	GpOutDMA CtrlFull	✓	✓	X	
26	GpOutDMA DataEmpty	✓	✓	X	
27	GpOutDMA DataFull	✓	✓	X	
28...31	Reserved	✓	✗	0	

Notes: Masks the "B" set of event flags to profile counter 0 (1=masked).

ResetStatus

Name	Type	Offset	Format
ResetStatus	Control Status *Control register*	0x00	Integer

Bits	Name	Read	Write	Reset	Description
0..30	Reserved	✓	✗	0x000 0.0000	
31	Software Reset Flag	✓	✓	0x000 0.0000	0 = GP is ready for use. 1 = GP is being reset and must not be used

Notes: Writing to the reset status register causes a software reset of the graphics processor (GP). The software reset does not reset the bus interface. The reset takes a number of cycles to complete during which the graphics processor should not be used. A flag in the register shows that the software reset is still in progress.

SyncDataFIFOSpace

Name	Type	Offset	Format
SyncDataFIFOSpace	Control Status	0x00D0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...5	SyncDataFIFO Space	✓	✕	0x20	Number of 32-bit spaces in the FB Sync Data FIFO. The FIFO is 32 spaces deep.
6...31	reserved	✓	✕	0	

Notes: The depth of the FIFO is specified by the Rectangle DMA core unit. The reset value is the FIFO depth, i.e. 32 bits

TestInputControl0

Name	Type	Offset	Format
TestInpu Control	Control Status	0x0200	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Test Input DeltaSwitch	✓	✓	0	
1	Test Input Delta N	✓	✓	0	
2	Test Input DeltaMux N	✓	✓	0	
3	Test Input TwoD	✓	✓	0	
4	Test Input Rasterizer	✓	✓	0	
5	Test Input Scissor	✓	✓	0	
6	Test Input Router	✓	✓	0	
7	TestInput Read Monitor	✓	✓	0	
8	Test Input LBRead	✓	✓	0	
9	Test Input Stencil	✓	✓	0	

10	Test Input LBWrite	✓	✓	0	
11	Test Input ColorDDA	✓	✓	0	
12	Test Input TX Coord	✓	✓	0	
13	Test Input TX Index	✓	✓	0	
14	Test Input TX Read	✓	✓	0	
15	Test Input TX LUT	✓	✓	0	
16	Test Input TX Filter	✓	✓	0	
17	Test Input TX Comp	✓	✓	0	
18	Test Input TX App	✓	✓	0	
19	Test Input YUV	✓	✓	0	
20	Test Input Fog	✓	✓	0	
21	Test Input AlphaTest	✓	✓	0	
22	Test Input FBRead	✓	✓	0	
23	Test Input Ablend	✓	✓	0	
24	Test Input Dither	✓	✓	0	
25	Test Input Logicop	✓	✓	0	
26	Test Input FBWrite	✓	✓	0	
27	Test Input Hostout	✓	✓	0	
28	Test Input GPOut	✓	✓	0	
29, 30	Which Delta	✓	✓	0	Selects which Delta or Delta Mux input FIFO to use.
31	Readback Disable	✓	✓	0	

Notes: This register sets the FIFO to which test writes are sent. The register must be enabled (bit 31) before any test writes or reads can take place.

TestInputControl1

Name	Type	Offset	Format
TestInpu Control1	Control Status	0x0290	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Test Input Command Unit	✓	✓	0	
1	Test Input Vertex Array Cache Unit	✓	✓	0	
2	Test Input Vertex Array Data Unit	✓	✓	0	
3	Test Input Rectangler DMA Unit	✓	✓	0	
4	Test Input Vertex Array Tag Unit	✓	✓	0	
5	Test Input Matrix Unit	✓	✓	0	
6	Test Input Pipe Manager Unit	✓	✓	0	
7	TestInput Read Vertex Machine Unit	✓	✓	0	
8	Test Input Programmable Vector Unit	✓	✓	0	
9	Test Input Transform Unit	✓	✓	0	
10	Test Input Cull Unit	✓	✓	0	
11	Test Input Geometry Unit	✓	✓	0	
12	Test Input TX Normalization Unit	✓	✓	0	
13	Test Input Texture Fog	✓	✓	0	
14	Test Input Texture Format	✓	✓	0	

15	Test Input Light Switch Unit	✓	✓	0	
16	Test Input Lighting Unit N	✓	✓	0	
17	Test Input Light Mux Unit N	✓	✓	0	
18	Test Input Material	✓	✓	0	
19	reserved	✓	✗	0	
20...22	Which Lighting	✓	✓	0	
23...31	reserved	✓	✗	0	

Notes: This register sets the FIFO to which test writes are sent. The register must be enabled (bit 31) before any test writes or reads can take place.

TestInputRdy0

Name	Type	Offset	Format
TestInpu Rdy	Control Status	0x0208	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Input Delta Switch Ready	✓	✓	0	
1	Test Input Delta N Ready	✓	✓	1	
2	Test Input Delta Mux N Ready	✓	✓	1	
3	Test Input TwoD Ready	✓	✓	1	
4	Test Input Rasterizer Ready	✓	✓	1	
5	Test Input Scissor Ready	✓	✓	1	
6	Test Input Router Ready	✓	✓	1	

7	Test Input Read Monitor Ready	✓	✓	1	
8	Test Input LBRead Ready	✓	✓	1	
9	Test Input Stencil Ready	✓	✓	1	
10	Test Input LBWrite Ready	✓	✓	1	
11	Test Input ColorDDA Ready	✓	✓	1	
12	Test Input TX Coord Ready	✓	✓	1	
13	Test Input TX Index Ready	✓	✓	1	
14	Test Input TX Read Ready	✓	✓	1	
15	Test Input TX LUT Ready	✓	✓	1	
16	Test Input TX Filter Ready	✓	✓	1	
17	Test Input TX Comp Ready	✓	✓	1	
18	Test Input TX App Ready	✓	✓	1	
19	Test Input YUV Ready	✓	✓	1	
20	Test Input Fog Ready	✓	✓	1	
21	Test Input AlphaTest Ready	✓	✓	1	
22	Test Input FBRead Ready	✓	✓	1	
23	Test Input Ablend Ready	✓	✓	1	
24	Test Input Dither Ready	✓	✓	1	
25	Test Input Logicop Ready	✓	✓	1	
26	Test Input FBWrite Ready	✓	✓	1	

27	Test Input Hostout Ready	✓	✓	1	
28	Test Input GPOut Ready	✓	✓	1	
29...31	Reserved	✓	✗	0	

Notes: This register shows the input status of all the core FIFOs when read. A write to this register sets a write pulse to the FIFO pointed at by the **TestInputControl**.

TestInputRdy1

Name	Type	Offset	Format
TestInpu Rdy1	Control Status	0x0298	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Input Command Unit Ready	✓	✓	0	
1	Test Input Vertex Array Cache Unit Ready	✓	✓	1	
2	Test Input Vertex Array Data Unit Ready	✓	✓	1	
3	Test Input Rectangle DMA Ready	✓	✓	1	
4	Test Input Vertex Array Tag Ready	✓	✓	1	
5	Test Input Matrix Ready	✓	✓	1	
6	Test Input Pipe Manager Ready	✓	✓	1	
7	Test Input Vertex Machine Ready	✓	✓	1	
8	Reserved	✓	✗	X	

9	Test Input Transform Unit Ready	✓	✓	1	
10	Test Input Cull Unit Ready	✓	✓	1	
11	Test Input Geometry Unit Ready	✓	✓	1	
12	Test Input Normalization Unit Ready	✓	✓	1	
13	Test Input Texture FOG Ready	✓	✓	1	
14	Test Input Texture Format Ready	✓	✓	1	
15	Test Input Light Switch Unit Ready	✓	✓	1	
16	Test Input Lighting Unit N Ready	✓	✓	1	
17	Test Input Light Mux Unit N Ready	✓	✓	1	
18	Test Input Material Unit Ready	✓	✓	1	
19	Reserved	✓	✗	0	
20...31	Test Input Fog Ready	✓	✓	1	
21	Test Input AlphaTest Ready	✓	✓	1	
22	Test Input FBRead Ready	✓	✓	1	
23	Test Input ABlend Ready	✓	✓	1	
24	Test Input Dither Ready	✓	✓	1	
25	Test Input Logicop Ready	✓	✓	1	

26	Test Input FBWrite Ready	✓	✓	1	
27	Test Input Hostout Ready	✓	✓	1	
28	Test Input GPOut Ready	✓	✓	1	
29...31	Reserved	✓	✗	0	

Notes: This register shows the input status of all the core FIFOs when read. A write to this register sets a write pulse to the FIFO pointed at by the **TestInputControl**.

TestOutputControl0

Name	Type	Offset	Format
TestOutputControl0	Control Status	0x0210	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Output Delta Switch	✓	✓	0	
1	Test Output Delta N	✓	✓	0	
2	Test Output Delta Mux N	✓	✓	0	
3	Test Output TwoD	✓	✓	0	
4	Test Output Rasterizer	✓	✓	0	
5	Test Output Scissor	✓	✓	0	
6	Test Output Router	✓	✓	0	
7	Test Output ReadMonitor	✓	✓	0	
8	Test Output LBRead	✓	✓	0	
9	Test Output Stencil	✓	✓	0	
10	Test Output LBWrite	✓	✓	0	
11	Test Output ColorDDA	✓	✓	0	

12	Test Output TX Coord	✓	✓	0	
13	Test Output TX Index	✓	✓	0	
14	Test Output TX Read	✓	✓	0	
15	Test Output TX LUT	✓	✓	0	
16	Test Output TX Filter	✓	✓	0	
17	Test Output TX Comp	✓	✓	0	
18	Test Output TX App	✓	✓	0	
19	Test Output YUV	✓	✓	0	
20	Test Output Fog	✓	✓	0	
21	Test Output AlphaTest	✓	✓	0	
22	Test Output FBRead	✓	✓	0	
23	Test Output ABlend	✓	✓	0	
24	Test Output Dither	✓	✓	0	
25	Test Output Logicop	✓	✓	0	
26	Test Output FBWrite	✓	✓	0	
27	Test Output Hostout	✓	✓	0	
28	Test Output GPOut	✓	✓	0	
29, 30	Which Delta	✓	✓	0	
31	reserved	✓	✗	0	

Notes: Sets the source location for FIFO test reads.. The individual Delta or Delta Mux FIFO to use is selected by the *WhichDelta* field

TestOutputControl1

Name	Type	Offset	Format
TestOutputControl1	Control Status	0x02a0	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Test Output GP InFIFO	✓	✓	0	
1	Test Output Command Unit	✓	✓	0	
2	Test Output Vertex Array Cache Unit	✓	✓	0	
3	Test Output Vertex Aray Data	✓	✓	0	
4	Test Output Rectangle DMA Unit	✓	✓	0	
5	Test Output Vertex Array Tag Unit	✓	✓	0	
6	Test Output Matrix Unit	✓	✓	0	
7	Test Output Pipe Manager Unit	✓	✓	0	
8	Test Output Vertex Machine Unit Ready	✓	✓	0	
9	Test Output Programmable Vector Unit	✓	✓	0	
10	Test Output Transform Unit	✓	✓	0	
11	Test Output Cull Unit	✓	✓	0	
12	Test Output Geometry Ready	✓	✓	0	
13	Test Output Normalization Unit Ready	✓	✓	0	

14	Test Output Texture Fog Unit	✓	✓	0	
15	Test Output Texture Format Unit	✓	✓	0	
16	Test Output Light Switch Unit	✓	✓	0	
17	Test Output Lighting Unit N	✓	✓	0	
18	Light Mux Unit N	✓	✓	0	
19	Test Output Material Unit	✓	✓	0	
20	reserved	✓	✗	0	
21...23	Which Lighting	✓	✓	0	
24...31	reserved	✓	✗	0	

TestOutputRdy0

Name	Type	Offset	Format
TestOutputRdy0	Control Status	0x0218	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Test Output DeltaSwitch Ready	✓	✓	0	
1	Test Output Delta N Ready	✓	✓	0	
2	Test Output Delta Mux N Ready	✓	✓	0	
3	Test Output TwoD Ready	✓	✓	0	
4	Test Output Rasterizer Ready	✓	✓	0	
5	Test Output Scissor Ready	✓	✓	0	
6	TestOutput Router Ready	✓	✓	0	

7	Test Output ReadMonitor Ready	✓	✓	0	
8	Test Output LBRead Ready	✓	✓	0	
9	Test Output Stencil Ready	✓	✓	0	
10	Test Output LBWrite Ready	✓	✓	0	
11	Test Output ColorDDA Ready	✓	✓	0	
12	Test Output TX Coord Ready	✓	✓	0	
13	Test Output TX Index Ready	✓	✓	0	
14	Test Output TX Read Ready	✓	✓	0	
15	Test Output TX LUT Ready	✓	✓	0	
16	Test Output TX Filter Ready	✓	✓	0	
17	Test Output TX Comp Ready	✓	✓	0	
18	Test Output TX App Ready	✓	✓	0	
19	Test Output YUV Ready	✓	✓	0	
20	Test Output Fog Ready	✓	✓	0	
21	Test Output AlphaTest Ready	✓	✓	0	
22	Test Output FBRead Ready	✓	✓	0	
23	Test Output ABlend Ready	✓	✓	0	
24	Test Output Dither Ready	✓	✓	0	

25	Test Output Logicop Ready	✓	✓	0	
26	Test Output FBWrite Ready	✓	✓	0	
27	Test Output Hostout Ready	✓	✓	0	
28	Test Output GP Out Ready	✓	✓	0	
29, 30	Reserved	✓	✕	0	
31	Test Output TagData Ready	✓	✓	0	

Notes: Shows the output status of all core FIFOs when read. A write to the register sets a read pulse to the FIFO pointed to by **TestOutputControl**.

TestOutputRdy1

Name	Type	Offset	Format
TestOutputRdy1	Control Status	0x0218	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Test Output GP InFIFO Ready	✓	✓	0	
1	Test Output Command Unit Ready	✓	✓	0	
2	Test Output vertex Array Cache Unit Ready	✓	✓	0	
3	Test Output Vertex Array Data Unit Ready	✓	✓	0	
4	Test Output Rectangle DMA Unit Ready	✓	✓	0	
5	Test Output Vertex Array Tag Unit Ready	✓	✓	0	
6	Test Output Matrix Unit Ready	✓	✓	0	
7	Test Output Pipe Manager Unit Ready	✓	✓	0	
8	Test Output Vertex Machine Unit Ready	✓	✓	0	
9	Reserved	✓	✗	0	
10	Test Output Transform Unit Ready	✓	✓	0	
11	Test Output Cull Unit Ready	✓	✓	0	

12	Test Output Geometry Unit Ready	✓	✓	0	
13	Test Output Normalization Unit Ready	✓	✓	0	
14	Test Output Texture Fog Unit Ready	✓	✓	0	
15	Test Output Texture Format Unit Ready	✓	✓	0	
16	Test Output Light Switch Unit Ready	✓	✓	0	
17	Test Output Lighting Unit N Ready	✓	✓	0	
18	Test Output Light Mux Unit N Ready	✓	✓	0	
19	Test Output Material Unit Ready	✓	✓	0	
20…30	reserved	✓	✕	0	
31	Test Output Tag Data Ready	✓	✓	0	

TestReadData

Name	Type	Offset	Format
TestRead Data	Control Status	0x0238	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..4	TestReadData	✓	✕	0.0000 b	Data

Notes: This register reads data from the register selected by **TestReadSelect**.

TestReadSelect

Name	Type	Offset	Format
TestWrite Select	Control Status	0x0230	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..4	TestReadSelect	✓	✓	0.0000 b	0=FIFO Tag 1=FIFO Data (31:0) 2=FIFO Data (63:32) 3=FIFO Data (95:64) 4=FIFO Data (127:96) 5= FIFO Data (159:128) 6= FIFO Data (191:160) Other values have undefined behaviour.

Notes: This register controls which portion of the data/Tag is read from the selected FIFO when the **TestReadData** register is read. The register post-increments after the data is read, i.e. Write 0 to **TestReadSelect** selects the FIFOtag, read data (**TestReadData**) comes from the FIFO Tag, and **TestReadSelect** auto-increments to 1 so the next **TestReadData** comes from FIFO Data bits 0...31.

TestWriteSelect

Name	Type	Offset	Format
TestWrite Select	Control Status	0x220	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..4	TestWriteSelect	✓	✓	0.0000 b	0=FIFO Tag 1=FIFO Data (31:0) 2=FIFO Data (63:32) 3=FIFO Data (95:64) 4=FIFO Data (127:96) 5= FIFO Data (159:128) 6= FIFO Data (191:160)

Notes: This register controls which portion of the data/Tag is written to in the selected FIFO when the **TestWriteData** register is written to. The register post-increments after the data is written, i.e. Write 0 to **TestWriteSelect** selects the FIFOtag, written data (**TestWriteData**) goes to the FIFO Tag, and **TestWriteSelect** auto-increments to 1 so the next **TestWriteData** goes to FIFO Data bits 0...31.

TestWriteData

Name	Type	Offset	Format
TestWriteData	Control Status	0x0228	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	TestWriteData	✓	✓	0x0000.0000	Data

Notes: Writes to this register are sent to the FIFO selected by **TestWriteSelect**

TexFIFOSpace

Name	Type	Offset	Format
TexFIFOSpace	Control Status	0x0128	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	TexFIFOSpace	✓	✗	0x10	

Notes: This register returns number of 128-bit spaces in the Texture Data FIFO. space is decremented by 1 after four 32-bit writes to the FIFO region. Software must always write in multiples of four 32-bit words.

TextureDMAAddress

Name	Type	Offset	Format
TextureDMAAddress	Control Status	0x0130	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...3	reserved	✓	✗	0	
4..31	TexDMA Address	✓	✗	X	128-bit aligned address

Notes: This register returns the address of the last data returned in response to a texture read operation.

TextureDMAControl

Name	Type	Offset	Format
TextureDMAControl	Control Status *Control register*	0x0140	Bitfield

Bits	Name	Read	Write	Reset	Description
0	reserved	✓	✗	0	
1	Protocol	✓	✓	0	0 = use PCI 1 = use AGP
2	Sideband	✓	✓	0	0 = use Register/Automatic mode 1 = use Sideband mode
3	Alignment	✓	✓	0	0 = Off 1 = Align to 64-bit boundaries where possible
4...6	Burst size	✓	✓	0	0-6 = log2 of size of DMA bursts in DWORDS.
7	RegHeader	✓	✓	0	0 = Header in host memory 1 = Read header from **TextureDMAHeader** registers
8...31	Reserved	✓	✗	X	

Notes:

TextureDMACount

Name	Type	Offset	Format
TextureDMACount	Control Status *Control register*	0x0138	Integer

Bits	Name	Read	Write	Reset	Description
0...27	Count	✓	✓	X	Count of 128-bit items to transfer
28...31	reserved	✓	✗	X	

Notes: Writing a count of 128-bit items to this register starts the texture DMA transfer from the **TextureDMAAddress** register. Reading the register returns the number of remaining unread items.

TextureDMAHeaderHigh

Name	Type	Offset	Format
TextureDMAHeaderHigh	Control Status	0x0150	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...31	Header 32-63	✓	✓	0x0000	Bits 32-63 of texture header

Notes: Bits 0-31 of the texture header, which normally contains the *LogicalTexturePage*..

TextureDMAHeaderLow

Name	Type	Offset	Format
TextureDMAHeaderLow	Control Status	0x0148	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...31	Header 0-31	✓	✓	0x0000	Bits 0-31 of texture header

Notes: Bits 32-63 of the texture header, which normally contains the *LogicalTexturePage*..

TextureDownloadControl

Name	Type	Offset	Format
TextureDownloadControl	Control Status *Control register*	0x108	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Texture Download Enable	✓	✓	0	0 = Disabled 1 = Enabled
1	Texture Download Busy	✓	✗	0	
2...7	reserved	✓	✗	X	
8..12	reserved	✓	✗	X	
13	SlaveTexture Download	✓	✓	0	0 = Use Texture DMA for downloads - Slave Writes to the FIFO are discarded. 1 = Use Slave writes into the FIFO. (Slave Reads of FIFO return zero)
14..31	Reserved	✓	✗	0	

Notes:

TextureOperation

Name	Type	Offset	Format
TextureOperation	Control Status *Control register*	0x110	Integer

Bits	Name	Read	Write	Reset	Description
0..8	Length	✓	✗	X	
9..10	Memory Pool	✓	✗	X	
11	Host Virt	✓	✗	X	
12..31	Reserved	✓	✗	0	

Notes: Required in Slave Download Mode to complete the Texture FIFO protocol.

TextureRBLightNumber

Name	Type	Offset	Format
TextureR 3LightNumber	Control Status	0x02B0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...5	TextureRB LightNumber	✓	✓	0x0000	
6...31	reserved	✓	✗	0	

Notes: Selects which light to read back from the texture format unit..

TextureRBSelect

Name	Type	Offset	Format
TextureR 3Select	Control Status	0x02B0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...2	TextureRB Select	✓	✓	0x0000	
3...31	reserved	✓	✗		

Notes: Selects the texture to read back from the texture format unit..

VClkRDacCtl

Name	Type	Offset	Format
VClkRDa :Ctl	Control Status	0x40	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	VidCtl(0) pin	✓	✓	0x0000.0000	00 = RDDClk set 0 01 = RDDClk set 1 10 = RDDClk set 2 11 = RDDClk set 3
1	VidCtl(1) pin	✓	✓	0	
2...31	Reserved	✓	✗	0	All bits 0

Notes: This 2 bit register is used to select which set of RAMDAC control registers is to be used to control the DClk PLL.

4.4 Region 0 Bypass Controls (0x0300-0x03FF)

The bypass unit is used to access the memory, the memory control registers, the video unit, and the VGA. It holds two DMA engines, one for reading from system memory and writing to local memory (DMARead) and one for writing to system memory and reading from local memory (DMAWrite). There is also byte swapping for upload and download, and conversion to and from YUV planar data format.

The DMA engines are controlled from a buffer of commands held in memory. DMARead takes commands from system memory, DMAWrite takes commands from local memory. Each command has the format:

Word 0: address of data in system memory, 128 bit aligned.

Word 1: address of data in local memory, 128 bit aligned.

Word 2: lower 16 bits = byte enable mask to apply to first transfer, upper 16 bits = mask to apply to last transfer.

Word 3: count of 128 bit items to transfer.

The command mechanism allows for full gather-scatter DMA, one important use is the ability to access non-contiguous system memory.

Arbitration for access to local memory is done in a strict priority, with any direct access being handled immediately. The second priority goes to DMARead (i.e. write to local memory) so that any data delivered by the bus is cleared immediately. If there are no other requests, the DMAWrite unit is allowed to read local memory and store the data ready for transfer over the PCI. The memory is used efficiently because request to the PCI are for bursts. The PCI will complete one burst before moving to the next.

ByAperture1Mode
ByAperture2Mode

Name	Type	Offset	Format
ByApertu :e1Mode	Bypass Control	0x0300	Bitfield
ByApertu :e2Mode	Bypass Control	0x0328	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..1	ByteSwap	✓	✓	0	Controls byte swapping on writing to or reading from local memory. 0 = ABCD (no swap) 1 = BADC (byte swapped) 2 = CDAB (half word swapped) 3 = DCBA (reversed)
2	PatchEnable	✓	✓	0	Organizes accesses to local memory to fit 2 dimensional patch. 0 = Off 1 = On
3..4	Format	✓	✓	0	Pixel format. YUV formats are converted from planar 420 to 422 format on writing, and from 422 to planar 420 on reads: 0 = Raw 1 = YUYV 2 = UYVY 3 = Reserved
5..6	PixelSize	✓	✓	0	0 = 8 bits 2 = 32 bits 1 = 16 bits 3 = Reserved
7..8	EffectiveStride	✓	✓	0	Stride used to calculate patched address. Should always be bigger or equal to the real stride of the display" 0 = 1024 1 = 2048 2 = 4096 3 = 8192
9..15	PatchOffsetX	✓	✓	0	Adjusts X position within patch.
16..20	PatchOffsetY	✓	✓	0	Adjusts Y position within patch.
21	Buffer	✓	✓	0	0 = Framebuffer 1 = Localbuffer
22, 23	SizeofMemA	✓	✓	0	Set the size of memory A: 0 = 8 Mbytes 1 = 16 Mbytes 2 = 32Mbytes 3 = 64Mbytes

24..26	DoubleWrite	✓	✓	0	Do two writes for every one received. Defines the boundary on which the second write occurs. A write to an odd multiple of the segment specified causes a write to the corresponding even segment; a write to an even segment causes a write to the odd segment. 0 = Off 1 = 1 Mbyte 2 = 2 Mbytes 3 = 4 Mbytes 4 = 8 Mbytes 5 = 16 Mbytes 6 = 32 Mbytes 7 = Reserved
27..31	Reserved	✓	✗	0	

Notes: These registers allow the required byte swapping and memory packing mode to be selected for each of the **CFGBaseAddr** memory apertures.

ByAperture1UStart
ByAperture2UStart

Name	Type	Offset	Format
ByApertu :e1UStart	Bypass Control	0x0318	Integer
ByApertu :e2UStart	Bypass Control	0x0340	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..23	UStart	✓	✓	X	Number of 128 bit transfers before interpreting data as U.
24..31	Reserved	✓	✗	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as U.

ByAperture1VStart
ByAperture2VStart

Name	Type	Offset	Format
ByApertu :e1VStart	Bypass Control	0x0320	Integer
ByApertu :e2VStart	Bypass Control	0x0348	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..23	VStart	✓	✓	X	Number of 128 bit transfers before interpreting data as V.
24..31	Reserved	✓	✕	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as V.

ByAperture1Ystart
ByAperture2YStart

Name	Type	Offset	Format
ByApertu :e1YStart	Bypass Control	0x0310	Integer
ByApertu :e2YStart	Bypass Control	0x0338	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..23	YStart	✓	✓	X	Number of 128 bit transfers before interpreting data as Y.
24..31	Reserved	✓	✕	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as Y.

ByAperture1Stride
ByAperture2Stride

Name	Type	Offset	Format
ByApertu :e1Stride	Bypass Control	0x0308	Integer
ByApertu :e2Stride	Bypass Control	0x0330	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..11	Stride	✓	✓	X	Number of pixels per line.
12..31	Reserved	✓	✗	0	

Notes: Sets the stride of the buffer in local memory. Only used when patching or doing YUV format conversions.

ByDMAReadCommandBase

Name	Type	Offset	Format
ByDMAI eadCommand Base	Bypass Control	0x0378	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✕	0	
4..31	Address	✓	✓	X	Base address of command buffer for DMA transfers from system memory to local memory. Always in system memory. Address is 128 bit aligned.

Notes:

ByDMAReadCommandCount

Name	Type	Offset	Format
ByDMAI eadCommand Count	Bypass Control	0x0380	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	Count	✓	✓	X	Number of command packets to transfer.

Notes:

ByDMAReadMode

Name	Type	Offset	Format
ByDMAI eadMode	Bypass Control *Control register*	0x0350	Bitfield

Bits	Name	Read	Write	Reset	Description
0..1	ByteSwap	✓	✓	0	Controls byte swapping on writing to or reading from local memory. 0 = ABCD (no swap) 1 = BADC (byte swapped) 2 = CDAB (half word swapped) 3 = DCBA
2	PatchEnable	✓	✓	0	Organizes accesses to local memory to fit 2 dimensional patch. 0 = Off 1 = On
3..4	Format	✓	✓	0	Pixel format. YUV formats are converted from planar 420 to 422 format on writing, and from 422 to planar 420 on reads. 0 = Raw 1 = YUYV 2 = UYVY 3 = reserved
5..6	PixelSize	✓	✓	0	0 = 8 bits 1 = 16 bits 2 = 32 bits 3 = reserved
7..8	EffectiveStride	✓	✓	0	0 = 1024 1 = 2048 2 = 4096 3 = 8192
9..15	PatchOffsetX	✓	✓	0	Adjusts X position within patch.
16..20	PatchOffsetY	✓	✓	0	Adjusts Y position within patch.
21	Buffer	✓	✓	0	0 = Framebuffer 1 = Localbuffer
22, 23	SizeofMemA	✓	✓	0	Set the size of memory A: 0 = 8 Mbytes 1 = 16 Mbytes 2 = 32Mbytes 3 = 64Mbytes
24	Active	✓	✓	0	Indicates the status of the DMA: 0 = DMA Idle 1 = DMA Running
25	MemType	✓	✓	0	Type of Bus Protocol to use for DMA: 0 = PCI 1 = AGP
26...28	Burst	✓	✓	0	Size of burst defined as log2 of burst size.
29	Align	✓	✓	0	Enables alignment of transfers to 64 byte boundaries. 0 = Off 1 = On
30..31	Reserved	✓	✗	0	Reserved

Notes: Controls the operation of the DMA controller reading data from system memory and writing it to local memory.

ByDMAReadStride

Name	Type	Offset	Format
ByDMAI eadStride	Bypass Control	0x0358	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..11	Stride	✓	✓	X	Number of pixels per line.
12..31	Reserved	✓	✗	0	

Notes: Sets the stride of the buffer in local memory. Only used when patching or doing YUV format conversions.

ByDMAReadUStart

Name	Type	Offset	Format
ByDMAI eadUStart	Bypass Control	0x0368	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..23	UStart	✓	✓	X	Number of 128 bit transfers before interpreting data as U.
24..31	Reserved	✓	✗	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as U.

ByDMAReadVStart

Name	Type	Offset	Format
ByDMAI eadVStart	Bypass Control *Control register*	0x0370	Integer

Bits	Name	Read	Write	Reset	Description
0..23	VStart	✓	✓	X	Number of 128 bit transfers before interpreting data as V.
24..31	Reserved	✓	✕	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as V.

ByDMAReadYStart

Name	Type	Offset	Format
ByDMAI eadYStart	Bypass Control *Control register*	0x0360	Integer

Bits	Name	Read	Write	Reset	Description
0..23	YStart	✓	✓	X	Number of 128 bit transfers before interpreting data as Y.
24..31	Reserved	✓	✕	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as Y.

ByDMAWriteCommand Base

Name	Type	Offset	Format
ByDMAWriteCommand Base	Bypass Control	0x03B0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✕	0	
4..31	Address	✓	✓	X	Base address of command buffer for DMA transfers from local memory to system memory. Always in local memory. Address is 128 bit aligned.

Notes:

ByDMAWriteCommandCount

Name	Type	Offset	Format
ByDMAWriteCommand Count	Bypass Control	0x03B8	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..31	Count	✓	✓	X	Number of command packets to transfer.

Notes:

ByDMAWriteMode

Name	Type	Offset	Format
ByDMAWriteMode	Bypass Control	0x0388	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..1	ByteSwap	✓	✓	0	Controls byte swapping on writing to or reading from local memory. 0 = ABCD (no swap) 1 = BADC (byte swapped) 2 = CDAB (half word swapped) 3 = DCBA
2	PatchEnable	✓	✓	0	Organizes accesses to local memory to fit 2 dimensional patch. 0 = Off 1 = On
3..4	Format	✓	✓	0	Pixel format. YUV formats are converted from planar 420 to 422 format on writing, and from 422 to planar 420 on reads. 0 = Raw 1 = YUYV 2 = UYVY 3 = reserved
5..6	PixelSize	✓	✓	0	0 = 8 bits 1 = 16 bits 2 = 32 bits 3 = reserved
7..8	EffectiveStride	✓	✓	0	0 = 1024 1 = 2048 2 = 4096 3 = 8192
9..15	PatchOffsetX	✓	✓	0	Adjusts X position within patch.
16..20	PatchOffsetY	✓	✓	0	Adjusts Y position within patch.
21	Buffer	✓	✓	0	0 = Framebuffer 1 = Localbuffer
22, 23	SizeofMemA	✓	✓	0	Set the size of memory A: 0 = 8 Mbytes 1 = 16 Mbytes 2 = 32Mbytes 3 = 64Mbytes
24	Active	✓	✓	0	Indicates the status of the DMA: 0 = DMA Idle 1 = DMA Running
25	MemType	✓	✓	0	Type of Bus Protocol to use for DMA: 0 = PCI 1 = AGP
26...28	Burst	✓	✓	0	Size of burst defined as log2 of burst size.
29	Align	✓	✓	0	Enables alignment of transfers to 64 byte boundaries. 0 = Off 1 = On
28..31	Reserved	✓	✗	0	

Notes: Controls the operation of the DMA controller reading data from local memory and writing it to system memory.

ByDMAWriteStride

Name	Type	Offset	Format
ByDMAWriteStride	Bypass Control *Control register*	0x0390	Integer

Bits	Name	Read	Write	Reset	Description
0..11	Stride	✓	✓	X	Number of pixels per line.
12..31	Reserved	✓	✗	0	

Notes: Sets the stride of the buffer in local memory. Only used when patching or doing YUV format conversions.

ByDMAWriteUStart

Name	Type	Offset	Format
ByDMAWriteUStart	Bypass Control *Control register*	0x03A0	Integer

Bits	Name	Read	Write	Reset	Description
0..23	UStart	✓	✓	X	Number of 128 bit transfers before interpreting data as U.
24..31	Reserved	✓	✗	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as U.

ByDMAWriteVStart

Name	Type	Offset	Format
ByDMAWriteVStart	Bypass Control *Control register*	0x03A8	Integer

Bits	Name	Read	Write	Reset	Description
0..23	VStart	✓	✓	X	Number of 128 bit transfers before interpreting data as V.
24..31	Reserved	✓	✗	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as V.

ByDMAWriteYStart

Name	Type	Offset	Format
ByDMAWriteYStart	Bypass Control *Control register*	0x0398	Integer

Bits	Name	Read	Write	Reset	Description
0..23	YStart	✓	✓	X	Number of 128 bit transfers before interpreting data as Y.
24..31	Reserved	✓	✗	0	

Notes: Used to control the conversion of planar YUV to packed YUV, this register sets the number of transfers to do before interpreting the data as Y.

4.5 Region 0 Memory Control (0x1000-0x1BFF)

Memory Control registers are accessed using the Bypass FIFO with an appropriate select line to indicate which set of registers is being addressed (Memory or Video). The Bypass address is formed from bits 11 to 2 of the incoming bus address. The bottom two bits are discarded to align the address to a 32-bit boundary. The resulting 10 bits of address are placed into the Bypass Input FIFO.

LocalMemACaps

Name	Type	Offset	Format
LocalMemACaps	Memory Control Command register	0x1018	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	Column Address	✓	✓	0xf	Address bits to use for column address. 0 = 8 bits 1 = 9 bits 2 = 10 bits 3 = 11 bits
4..7	RowAddress	✓	✓	0xf	Address bits to use for row address. 0 = 10 bits 1 = 11 bits 2 = 12 bits
8..11	BankAddress	✓	✓	0	Address bits to use for bank address. 0 = 1 bit 1 = 2 bits
12..15	ChipSelect	✓	✓	0	Address bits to use for chip select.
16..19	PageSize	✓	✓	0	Page size (units = full width of memory) 0 = 32 units 1 = 64 units, etc
20..23	RegionSize	✓	✓	0xf	Region size (units = full width of memory) 0 = 32 units 1 = 64 units, etc
24	NoPrecharge Opt	✓	✓	0	0 = off 1 = on
25	SpecialMode Opt	✓	✓	0	0 = off 1 = on
26	TwoColor BlockFill	✓	✓	0	0 = off 1 = on
27	Combine Banks	✓	✓	0	0 = off 1 = on
28	NoWriteMask	✓	✓	0x1	0 = off 1 = on
29	NoBlockFill	✓	✓	0x1	0 = off 1 = on
30, 31	Reserved	✓	✗	0	

Notes:
1. The *ColumnAddress*, *RowAddress*, *BankAddress*, and *ChipSelect* fields select the bits of the absolute physical address that are to be used to define corresponding parameters. Each value follows on from the previous one, so the *ChipSelect* value starts at *ColumnAddress* + *RowAddress* + *BankAddress* and continues for *ChipSelect* bits.
2. The *PageSize* field defines the size of the page, and the *RegionSize* field defines the size of the region of memory that each of the four page detectors should be assigned to (so that it is set to one quarter of the memory size).

LocalMemAControl

Name	Type	Offset	Format
LocalMemAControl	Memory Control Command register	0x1028	Bitfield

Bits	Name	Read	Write	Reset	Description	
0..2	CASLatency	✓	✓	110b	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
3	Interleave	✓	✔	0	1 = On	0 = Off
4...6	Address Extension	✓	✓	0	0 = 0 bits 2 = 2 bits 4 = 4 bits 6 = 6 bits	1 = 1 clock 3 = 3 bits 5 = 5 bits 7 = 7 bits
7...21	Reserved	✓	✕	0		
22..31	Mode	✓	✓	00001 10000 b	Mode register value used to configure memory. Bit 22 coresponds to bit 0 of register, bit 31 corresponds to bit 9 of register.	

Notes:
1. Latency values are for delays from the current operation to the next. If the delay is set to zero the next operation can follow the current one in the next CLK cycle.
   This generally means that the value loaded into the register is the corresponding data sheet value minus one. For example, the data sheet may specify the block write cycle time to be 2 clocks, so the register value would be one because there has to be a one clock delay between block writes.
2. Bits 22 and 31 of *LocalMemAControl* register correspond respectively to bits 0 and 9 of the mode register in the memory device.
3. *Interleave* with 64MB RAM creates a 128MB framebuffer which is dispersed into 16MB chunks totalling 64MB. The chunks can be used for a variety of purposes depending on the API. Most boards require a supplementary resistor to work properly in Interleave mode.
4. For bitfield details see Volume IV, Chapter 9 – *Memory Systems*

LocalMemADelayIn0

Name	Type	Offset	Format
LocalMemADelayIn0	Memory Control Command register	0x1088	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay0	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert0	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay1	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert1	✓	✓	0	0 = no invert 1 = invert
10..13	StrobeDelay2	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
14	StrobeInvert2	✓	✓	0	0 = no invert 1 = invert
15..18	StrobeDelay3	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
19	StrobeInvert3	✓	✓	0	0 = no invert 1 = invert
20..23	StrobeDelay4	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
24	StrobeInvert4	✓	✓	0	0 = no invert 1 = invert
25..28	StrobeDelay5	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
29	StrobeInvert5	✓	✓	0	0 = no invert 1 = invert
30..31	Reserved	✓	✗	0	

Notes:

LocalMemADelayIn1

Name	Type	Offset	Format
LocalMemADelayIn1	Memory Control Command register	0x1090	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay6	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert6	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay7	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert7	✓	✓	0	0 = no invert 1 = invert
10..31	Reserved	✓	✗	0	

Notes:

LocalMemADelayOut0

Name	Type	Offset	Format
LocalMemADelayOut0	Memory Control Command register	0x1098	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay0	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert0	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay1	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert1	✓	✓	0	0 = no invert 1 = invert
10..13	StrobeDelay2	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
14	StrobeInvert2	✓	✓	0	0 = no invert 1 = invert
15..18	StrobeDelay3	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
19	StrobeInvert3	✓	✓	0	0 = no invert 1 = invert
20..23	StrobeDelay4	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
24	StrobeInvert4	✓	✓	0	0 = no invert 1 = invert
25..28	StrobeDelay5	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
29	StrobeInvert5	✓	✓	0	0 = no invert 1 = invert
30..31	Reserved	✓	✕	0	

Notes:

LocalMemADelayOut1

Name	Type	Offset	Format
LocalMemADelayOut1	Memory Control Command register	0x10A0	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay6	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert6	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay7	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert7	✓	✓	0	0 = no invert 1 = invert
10..13	ClockDelay	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
14	ClockInvert	✓	✓	0	0 = no invert 1 = invert
15..31	Reserved	✓	✕	0	

Notes:

LocalMemAPowerDown

Name	Type	Offset	Format
LocalMemPowerDown	Memory Control Command register	0x1038	Bitfield

Bits	Name	Read	Write	Reset	Description	
0	Enable	✓	✓	0	0 = Off	1 = On
1..16	Reserved	✓	✗	0		
17..31	Delay	✓	✓	0	Timeout in 32 clock units	

Notes: Delay = timeout between reseting memory to low power mode in 32 clock units.

LocalMemARefresh

Name	Type	Offset	Format
LocalMemARefresh	Memory Control Command register	0x1030	Bitfield

Bits	Name	Read	Write	Reset	Description	
0	Enable	✓	✓	1	0 = Off	1 = On
1..7	RefreshDelay	✓	✓	0		
8..31	Reserved	✓	✗	0	Delay in 32 clock units	

Notes: Delay between refresh cycles in 32 clock units.

LocalMemATiming

Name	Type	Offset	Format
LocalMemTiming	Memory Control *Command register*	0x1020	Bitfield

Bits	Name	Read	Write	Reset	Description	
0..1	TurnOn	✓	✓	0x3	0 = 0 clocks 3 = 3 clock	2 = 2 clocks 1 = 1 clock
2..3	TurnOff	✓	✓	0x3	0 = 0 clocks 2 = 2 clocks	1 = 1 clock 3 = 3 clock
4..5	RegisterLoad	✓	✓	0x3	0 = 0 clocks 2 = 2 clocks	1 = 1 clock 3 = 3 clock
6..7	BlockWrite	✓	✓	0x3	0 = 0 clocks 2 = 2 clocks	1 = 1 clock 3 = 3 clock
8..10	ActivateTo Command	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
11..13	Precharge ToActivate	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks

14..16	BlockWriteTo Precharge	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
17..19	WriteTo Precharge	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
20..23	ActivateTo Precharge	✓	✓	0xF	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks 8 = 8 clocks 10 = 10 clocks 12 = 12 clocks 14 = 14 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks 9 = 9 clocks 11 = 11 clocks 13 = 13 clocks 15 = 15 clocks
24..27	RefreshCycle	✓	✓	0xF	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks 8 = 8 clocks 10 = 10 clocks 12 = 12 clocks 14 = 14 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks 9 = 9 clocks 11 = 11 clocks 13 = 13 clocks 15 = 15 clocks
28..31	ActivateTo Activate	✓	✓	0xf	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks 8 = 8 clocks 10 = 10 clocks 12 = 12 clocks 14 = 14 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks 9 = 9 clocks 11 = 11 clocks 13 = 13 clocks 15 = 15 clocks

Notes: Values are for delays from the current operation to the next. If the delay is set to zero the next operation can follow the curent one in the next clock cycle. This generally means that the value loaded into the register is the corresponding data sheet value minus one. For example, the data sheet may specify the block write cycle time to be 2 clocks, so the register value would be 1 because there has to be a one clock delay between block writes.

LocalMemBCaps

Name	Type	Offset	Format
LocalMemBCaps	Memory Control Command register	0x1040	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	Column Address	✓	✓	1110b	Address bits to use for column address. 0 = 8 bits 1 = 9 bits 2 = 10 bits 3 = 11 bits
4..7	RowAddress	✓	✓	1111b	Address bits to use for row address. 0 = 10 bits 1 = 11 bits 2 = 12 bits
8..11	BankAddress	✓	✓	0	Address bits to use for bank address. 0 = 1 bit 1 = 2 bits
12..15	ChipSelect	✓	✓	0	Address bits to use for chip select.
16..19	PageSize	✓	✓	0	Page size (units = full width of memory) 0 = 32 units 1 = 64 units, etc
20..23	RegionSize	✓	✓	1111b	Region size (units = full width of memory) 0 = 32 units 1 = 64 units, etc
24	NoPrecharge Opt	✓	✓	0	0 = off 1 = on
25	SpecialMode Opt	✓	✓	0	0 = off 1 = on
26	TwoColor BlockFill	✓	✓	0	0 = off 1 = on
27	Combine Banks	✓	✓	0	0 = off 1 = on
28	NoWriteMask	✓	✓	1	0 = off 1 = on
29	NoBlockFill	✓	✓	1	0 = off 1 = on
30, 31	Reserved	✓	✕	0	

Notes:
1. The reset state guarantees access to the lower 512 bytes of memory.
2. The *ColumnAddress*, *RowAddress*, *BankAddress*, and *ChipSelect* fields select the bits of the absolute physical address that are to be used to define corresponding parameters. Each value follows on from the previous one, so the *ChipSelect* value starts at *ColumnAddress* + *RowAddress* + *BankAddress* and continues for *ChipSelect* bits.
3. The *PageSize* field defines the size of the page, and the *RegionSize* field defines the size of the region of memory that each of the four page detectors should be assigned to (so that it is set to one quarter of the memory size).

LocalMemBControl

Name	Type	Offset	Format
LocalMemBControl	Memory Control Command register	0x1050	Bitfield

Bits	Name	Read	Write	Reset	Description
0..2	CASLatency	✓	✓	110b	0 = 0 clocks 1 = 1 clock 2 = 2 clocks 3 = 3 clocks 4 = 4 clocks 5 = 5 clocks 6 = 6 clocks 7 = 7 clocks
3	Interleave	✓	✔	0	0 = Off 1 = On
4...6	Address Extension	✓	✓	0	0 = 0 bits 1 = 1 clock 2 = 2 bits 3 = 3 bits 4 = 4 bits 5 = 5 bits 6 = 6 bits 7 = 7 bits
7...21	Reserved	✓	✕	0	
22..31	Mode	✓	✓	00001 10000 b	Mode register value used to configure memory. Bit 22 coresponds to bit 0 of register, bit 31 corresponds to bit 9 of register.

Notes:
1. Latency values are for delays from the current operation to the next. If the delay is set to zero the next operation can follow the current one in the next CLK cycle.
   This generally means that the value loaded into the register is the corresponding data sheet value minus one. For example, the data sheet may specify the block write cycle time to be 2 clocks, so the register value would be one because there has to be a one clock delay between block writes.
2. Bits 22 and 31 of *LocalMemBControl* register correspond respectively to bits 0 and 9 of the mode register in the memory device.
3. For bitfield details see Volume IV, Chapter 9 – *Memory Systems*

LocalMemBDelayIn0

Name	Type	Offset	Format
LocalMemBDelayIn0	Memory Control Command register	0x10A8	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay0	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert0	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay1	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert1	✓	✓	0	0 = no invert 1 = invert

10..13	StrobeDelay2	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
14	StrobeInvert2	✓	✓	0	0 = no invert 1 = invert
15..18	StrobeDelay3	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
19	StrobeInvert3	✓	✓	0	0 = no invert 1 = invert
20..23	StrobeDelay4	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
24	StrobeInvert4	✓	✓	0	0 = no invert 1 = invert
25..28	StrobeDelay5	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
29	StrobeInvert5	✓	✓	0	0 = no invert 1 = invert
30..31	Reserved	✓	✕	0	

Notes:

LocalMemBDelayIn1

Name	Type	Offset	Format
LocalMemBDelayIn1	Memory Control Command register	0x10B0	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay6	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert6	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay7	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert7	✓	✓	0	0 = no invert 1 = invert
10..31	Reserved	✓	✕	0	

Notes:

LocalMemBDelayOut0

Name	Type	Offset	Format
LocalMemBDelayOut0	Memory Control Command register	0x10B8	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay0	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert0	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay1	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert1	✓	✓	0	0 = no invert 1 = invert
10..13	StrobeDelay2	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
14	StrobeInvert2	✓	✓	0	0 = no invert 1 = invert
15..18	StrobeDelay3	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
19	StrobeInvert3	✓	✓	0	0 = no invert 1 = invert
20..23	StrobeDelay4	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
24	StrobeInvert4	✓	✓	0	0 = no invert 1 = invert
25..28	StrobeDelay5	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
29	StrobeInvert5	✓	✓	0	0 = no invert 1 = invert
30..31	Reserved	✓	✗	0	

Notes:

LocalMemBDelayOut1

Name	Type	Offset	Format
LocalMemBDelayOut1	Memory Control Command register	0x10C0	Bitfield

Bits	Name	Read	Write	Reset	Description
0..3	StrobeDelay6	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
4	StrobeInvert6	✓	✓	0	0 = no invert 1 = invert
5..8	StrobeDelay7	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
9	StrobeInvert7	✓	✓	0	0 = no invert 1 = invert
10..13	ClockDelay	✓	✓	0	Bit code of delay: 0 = 0 taps, 15 = 15 taps
14	ClockInvert	✓	✓	0	0 = no invert 1 = invert
15..31	Reserved	✓	✗	0	

Notes:

LocalMemBPowerDown

Name	Type	Offset	Format
LocalMemPowerDown	Memory Control Command register	0x1060	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Enable	✓	✓	0	0 = Off 1 = On
1..16	Reserved	✓	✗	0	
17..31	Delay	✓	✓	0	Timeout in 32 clock units

Notes: Delay = timeout between reseting memory to low power mode in 32 clock units.

LocalMemBRefresh

Name	Type	Offset	Format
LocalMemBRefresh	Memory Control Command register	0x1058	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Enable	✓	✓	1	0 = Off 1 = On
1..7	RefreshDelay	✓	✓	0	
8..31	Reserved	✓	✗	0	Delay in 32 clock units

Notes: Delay between refresh cycles in 32 clock units.

LocalMemBTiming

Name	Type	Offset	Format
LocalMemTiming	Memory Control *Command register*	0x1048	Bitfield

Bits	Name	Read	Write	Reset	Description	
0..1	TurnOn	✓	✓	0x3	0 = 0 clocks 3 = 3 clock	2 = 2 clocks 1 = 1 clock
2..3	TurnOff	✓	✓	0x3	0 = 0 clocks 2 = 2 clocks	1 = 1 clock 3 = 3 clock
4..5	RegisterLoad	✓	✓	0x3	0 = 0 clocks 2 = 2 clocks	1 = 1 clock 3 = 3 clock
6..7	BlockWrite	✓	✓	0x3	0 = 0 clocks 2 = 2 clocks	1 = 1 clock 3 = 3 clock
8..10	ActivateTo Command	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
11..13	Precharge ToActivate	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks

14..16	BlockWriteTo Precharge	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
17..19	WriteTo Precharge	✓	✓	0x7	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks
20..23	ActivateTo Precharge	✓	✓	0xF	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks 8 = 8 clocks 10 = 10 clocks 12 = 12 clocks 14 = 14 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks 9 = 9 clocks 11 = 11 clocks 13 = 13 clocks 15 = 15 clocks
24..27	RefreshCycle	✓	✓	0xF	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks 8 = 8 clocks 10 = 10 clocks 12 = 12 clocks 14 = 14 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks 9 = 9 clocks 11 = 11 clocks 13 = 13 clocks 15 = 15 clocks
28..31	ActivateTo Activate	✓	✓	0xf	0 = 0 clocks 2 = 2 clocks 4 = 4 clocks 6 = 6 clocks 8 = 8 clocks 10 = 10 clocks 12 = 12 clocks 14 = 14 clocks	1 = 1 clock 3 = 3 clocks 5 = 5 clocks 7 = 7 clocks 9 = 9 clocks 11 = 11 clocks 13 = 13 clocks 15 = 15 clocks

Notes: Values are for delays from the current operation to the next. If the delay is set to zero the next operation can follow the curent one in the next clock cycle. This generally means that the value loaded into the register is the corresponding data sheet value minus one. For example, the data sheet may specify the block write cycle time to be 2 clocks, so the register value would be 1 because there has to be a one clock delay between block writes.

LocalMemProfileCount0

Name	Type	Offset	Format
LocalMemProfileCount0	Memory Control Command register	0x1070	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Mask	✓	✓	0	Count of profile events

Notes: This register determines the bits that get written to memory by way of the bypass.

LocalMemProfileCount1

Name	Type	Offset	Format
LocalMemProfileCount1	Memory Control Command register	0x1080	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Mask	✓	✓	0	Count of profile events

Notes: This register determines the bits that get written to memory by way of the bypass.

LocalMemProfileMask0

Name	Type	Offset	Format
LocalMemProfileMask0	Memory Control Command register	0x1068	Bitfield

Bits	Name	Read	Write	Reset	Description	
0	Always	✓	✓	x	0 = Exclude	1 = Include
1	IdleA	✓	✓	x	0 = Exclude	1 = Include
2	Page-breakA	✓	✓	x	0 = Exclude	1 = Include
3	ReadA	✓	✓	x	0 = Exclude	1 = Include
4	WriteA	✓	✓	x	0 = Exclude	1 = Include
5	OtherA	✓	✓	x	0 = Exclude	1 = Include
6	IdleB	✓	✓	x	0 = Exclude	1 = Include
7	Page-breakB	✓	✓	x	0 = Exclude	1 = Include
8	ReadB	✓	✓	x	0 = Exclude	1 = Include
9	WriteB	✓	✓	x	0 = Exclude	1 = Include
10	OtherB	✓	✓	x	0 = Exclude	1 = Include
11…29	Reserved	✓	✗	x		
30	Mode	✓	✓	x	0 = And	1 = Or
31	Reserved	✓	✗	0		

Notes:

LocalMemProfileMask1

Name	Type	Offset	Format
LocalMemProfileMask1	Memory Control Command register	0x1078	Bitfield

Bits	Name	Read	Write	Reset	Description	
0	Always	✓	✓	x	0 = Exclude	1 = Include
1	IdleA	✓	✓	x	0 = Exclude	1 = Include
2	Page-breakA	✓	✓	x	0 = Exclude	1 = Include
3	ReadA	✓	✓	x	0 = Exclude	1 = Include
4	WriteA	✓	✓	x	0 = Exclude	1 = Include
5	OtherA	✓	✓	x	0 = Exclude	1 = Include
6	IdleB	✓	✓	x	0 = Exclude	1 = Include
7	Page-breakB	✓	✓	x	0 = Exclude	1 = Include
8	ReadB	✓	✓	x	0 = Exclude	1 = Include
9	WriteB	✓	✓	x	0 = Exclude	1 = Include
10	OtherB	✓	✓	x	0 = Exclude	1 = Include
11…29	Reserved	✓	✗	x		
30	Mode	✓	✓	x	0 = And	1 = Or
31	Reserved	✓	✗	0		

Notes:

MemBypassWriteMask

Name	Type	Offset	Format
MemBypassWriteMask	Memory Control Command register	0x1008	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Mask	✓	✓	0xFFFFFFFF	Per bit control: 0 = mask write, 1 = allow write

Notes: This register determines the bits that get written to memory by way of the bypass.

MemCounter

Name	Type	Offset	Format
MemCounter	Memory Control Command register	0x1000	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Count	✓	✕	0	

Notes: **MemCounter** can also be used as a delay timer for non-host backoff algorithms etc.

MemScratch

Name	Type	Offset	Format
MemScratch	Memory Control *Command register*	0x1010	Integer

Bits	Name	Read	Write	Reset	Description
0..31		✓	✓	0	Scratch memory

Notes: Scratch memory

RemoteMemControl

Name	Type	Offset	Format
RemoteMemControl	Memory Control Command register	0x1100	Integer

Bits	Name	Read	Write	Reset	Description	
0	TxReadType	✓	✓	0	0 = PCI	1 = AGP
1	LbReadType	✓	✓	0	0 = PCI	1 = AGP
2	LbWriteType	✓	✓	0	0 = PCI	1 = AGP
3	FbReadType	✓	✓	0	0 = PCI	1 = AGP
4	FbWriteType	✓	✓	0	0 = PCI	1 = AGP
5..31	Reserved	✓	✕	0		

Notes: The Remote Memory Controller reads and writes over the PCI bus using either PCI or AGP protocols. Requests from the units are sent to an arbiter which selects the source. The access is passed to/from the PCI bus via a FIFO protocol (which may or may not involve an actual FIFO).

4.6 Region 0 FB Sync Data FIFO (0x1C00-0x1FFF)

The rectangle DMA core unit has a data FIFO which allows "sideband" synchronization data to be inserted into the command stream. This FIFO is accessible through the control space at offset 0x1C00. It is 256 dwords in length. Any dword write to this region causes an entry to be put into the FIFO. A read from this region return 0. If the FIFO is full then a slave write should be discarded unless the *FBSync Disconnect* signal is set in the **FIFODisconnect** Register. In this case, a retry should be asserted on the PCI bus.

4.7 Region 0 GP FIFO (0x2000-0x2FFF)

The graphics processor FIFO access area is a direct route to the GP Input and Output FIFOs. Writing to any address in this region causes the data to be sent to the Input FIFO; reading from any address accesses data from the Output FIFO. When this area is written to directly, it should be in the form tag followed by data, or one of the efficient transfer mechanisms (such as index mode DMA). The bus interface is responsible for decoding these transfers, and writing the appropriate tag/data pairs into the Input FIFO.

4.8 Region 0 Video Control (0x3000-0x33FF)

Video Control registers are accessed using the Bypass FIFO with an appropriate select line to indicate which set of registers is being addressed (Memory or Video). The Bypass address is formed from bits 11 to 2 of the incoming bus address. The bottom two bits are discarded to align the address to a 32-bit boundary. The resulting 10 bits of address are placed into the Bypass Input FIFO.

DisplayData

Name	Type	Offset	Format
DisplayData	Video Control Register	0x3068	Bitfield

Bits	Name	Read	Write	Reset	Description	
0	DataIn	✓	✗	X	0 = Data line is low	1 = Data line is high
1	ClkIn	✓	✗	X	0 = Clock line is low	1 = Clock line is high
2	DataOut	✓	✓	1	0 = Drive data line low	1 = Tri-state data line
3	ClkOut	✓	✓	1	0 = Drive clock line low 1 = Tri-state clock line	
4	LatchedData	✓	✗	0	0 = Data latched at 0	1 = Data latched at 1
5	DataValid	✓	✓	0	0 = DataIn not valid	1 = DataIn valid

6	Start	✓	✓	0	0 = Has not passed through start state 1 = Has passed through start state
7	Stop	✓	✓	0	0 = Has not passed through stop state 1 = Has passed through stop state
8	Wait	✓	✓	0	0 = Do not insert wait states 1 = Insert wait states
9	UseMonitorID	✓	✓	0	0 = Use DDC 1 = Use MonitorID
10..11	MonitorIDIn[1..0]	✓	✕	X	0 = Data line is low, clock line is low 1 = Data line is high, clock is high
12	Reserved	✓	✕	0	Read back as zeros.
13..14	MonitorIDOut[1..0]	✕	✓	0x3	0 = Drive data line low 1 = Tri-state data line
15..31	Reserved	✓	✕	0	Read back as zeros.

Notes:
- Some bits in this register are set during operation and cleared by writing to the register with those bits set. The bits are *DataValid*, *Start* and *Stop*.
- *UseMonitorID* (bit 9) allows the programmer to chose whether to drive the VIDDDC clock and data pins (AG33, AG34) by setting bits 0...3 or bits 10, 11, 13 and 14 - the latter being primarily intended for Macintosh compatibility.
- Reset value = 000.000.000.000.111X.XX00.0000.11XX

FifoControl

Name	**Type**	**Offset**	**Format**
FifoControl	Video Control Register	0x3078	Bitfield

Bits	**Name**	**Read**	**Write**	**Reset**	**Description**
0..4	LowThreshold	✓	✕	0x10	Request data from memory with low priority when there are this many spaces in the fifo.
5..7	Reserved	✓	✕	0	
8..12	High Threshold	✓	✕	0x10	Request data from memory with high priority when there are this many spaces in the fifo.
13..15	Reserved	✓	✕	0	
16	Underflow	✓	✓	0	This bit is set by the by the behavioural code. It is cleared by writing a 1 to this bit. 0 = underflow has not occurred 1 = underflow has occured
17..31	Reserved	✓	✕	0	

Notes: If the FIFO underflows bit 16 is set - this can be used to help set the best threshold values for the FIFO. Register Reset = 0000.0000.0000.0000.0001.0000.0001.0000

HbEnd

Name	Type	Offset	Format
HbEnd	Video Control *Control register*	0x3020	Integer

Bits	Name	Read	Write	Reset	Description
0..10	HbEnd	✓	✓	x	First 128 bit unit out of horizontal blank
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

HgEnd

Name	Type	Offset	Format
HgEnd	Video Control *Control register*	0x3018	Integer

Bits	Name	Read	Write	Reset	Description
0..10	HgEnd	✓	✓	X	Last 128 bit unit in gate period
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

HsEnd

Name	Type	Offset	Format
HsEnd	Video Control *Control register*	0x3030	Integer

Bits	Name	Read	Write	Reset	Description
0..10	HsEnd	✓	✓	X	First 128 bit unit out of horizontal sync.
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

HsOffset

Name	Type	Offset	Format
HsOffset	Video Control *Control register*	0x3098	Integer

Bits	Name	Read	Write	Reset	Description
0..10	HsEnd	✓	✓	X	First 128 bit unit out of horizontal sync.
11..31	Reserved	✓	✗	0	

Notes: Used to compensate for clocking in Genlock - the start y of the locked Hsync is HsStart + HsOffset.
See *Multi-rasterizer Setup* in the *GLINT R5 Programmer's Guide*, Volume I, chapter 5.
Reset = 000.000.000.000.000.0XXX.XXX.XXX

HsStart

Name	Type	Offset	Format
HsStart	Video Control *Control register*	0x3028	Integer

Bits	Name	Read	Write	Reset	Description
0..10	HsStart	✓	✓	X	First 128 bit unit in horizontal sync.
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

HTotal

Name	Type	Offset	Format
HTotal	Video Control *Control register*	0x3010	Integer

Bits	Name	Read	Write	Reset	Description
0..10	HTotal	✓	✓	X	Last 128 bit unit (including horizontal blank period) on screen
11..31	Reserved	✓	✗	0	

Notes: Reset value = 000.000.000.000.000.0XXX.XXX.XXX

InterruptLine

Name	Type	Offset	Format
Interrupt Line	Video Control	0x3060	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..10	InterruptLine	✓	✓	X	Generate interrupt at start of this line
11..31	Reserved	✓	✗	0	

Notes:

MiscControl

Name	Type	Offset	Format
MiscControl	Video Control	0x3088	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	StripeEnable	✓	✓	0	0 = off 1 = primary
1..3	Size	✓	✓	0	Height of stripe in log2 of lines
4..5	Count	✓	✓	0	Number of rasterizers: 0 = 1 1 = 2 2 = 4 3 = 8
6..8	StripeID	✓	✓	0	The stripe number this chip should display
9..16	StripeOffset	✓	✓	0	Offset to stripe (in lines) for panning in Y
17	Reserved	✓	✕	X	
18	CompactFB	✓	✓	0	Use compact framebuffer 0 = Off 1 = On
19	ByteDouble	✓	✓	0	Output each byte twice to allow fine control over timing for 8-bit pixels. 0 = off 1 = on
20…31	Reserved	✓	✕	0	

Notes: MiscControl controls digital striping. It should not be confused with RDStripe which controls analogue striping. *CompactFB* removes address space allocation for stripes owned by other rasterizers.

- Reset = 000.000.00XX.XXXX.XXXX.XXXX.XXXX.XXXX

ScreenBase

Name	Type	Offset	Format
ScreenBase	Video Control	0x3000	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..21	ScreenBase	✓	✓	X	Base address of screen in 128 bit units.
22...29	Reserved	✓	✕	0	
30	MemType	✓	✓	0	0 = mem type A 1 = Mem type B
31	Reserved	✓	✕	0	

Notes: Reset = 000.000.00XX.XXXX.XXXX.XXXX.XXXX.XXXX

ScreenBaseRight

Name	Type	Offset	Format
ScreenBa eRight	Video Control *Control register*	0x3080	Integer

Bits	Name	Read	Write	Reset	Description
0..21	ScreenBase Right	✓	✓	X	Base address of right screen in 128 bit units.
22..29	Reserved	✓	✗	0	
30	MemType	✓	✓	X	0 = MemA 1 = MemB
31	Reserved	✓	✗	0	

Notes: Reset = 000.000.00XX.XXXX.XXXX.XXXX.XXXX.XXXX

ScreenStride

Name	Type	Offset	Format
ScreenStri de	Video Control *Control register*	0x3008	Integer

Bits	Name	Read	Write	Reset	Description
0..21	ScreenStride	✓	✓	X	Stride between stripes in 128 bit units.
22..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.0000.XXXX.XXXX.XXXX.XXXX.XXXX

StripeStride

Name	Type	Offset	Format
StripeStri le	Video Control *Control register*	0x3090	Integer

Bits	Name	Read	Write	Reset	Description
0..21	ScreenStride	✓	✓	X	Stride between stripes in 128 bit units.
22..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.0000.XXXX.XXXX.XXXX.XXXX.XXXX

VbEnd

Name	Type	Offset	Format
VbEnd	Video Control	0x3040	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..10	VbEnd	✓	✓	X	First scanline out of vertical blank
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

VerticalLineCount

Name	Type	Offset	Format
VerticalLineCount	Video Control	0x3070	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..10	VerticalLineCount	✓	✗	X	Current vertical line.
11..31	Reserved	✓	✗	0	

Notes: Reset = 0000.0000.0000.0000.0000.0XXX.XXXX.XXXX

VideoControl

Name	Type	Offset	Format
VideoCo1 trol	Video Control *Control register*	0x3058	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Enable	✓	✓	0	0 = GP video disabled 1 = GP video enabled
1	BlankCtl	✓	✓	0	0 = Active High 1 = Active Low
2	LineDouble	✓	✓	0	0 = Off 1 = On
3..4	HSyncCtl	✓	✓	0	0 = Forced High 1 = Active High 2 = Forced Low 3 = Active Low
5..6	VSyncCtl	✓	✓	0	0 = Forced High 1 = Active High 2 = Forced Low 3 = Active Low
7	BypassPending	✓	✕	0	Read only bit set when ScreenBase register is loaded. It is cleared when new value in ScreenBase has been used (i.e. during VBlank) 0 = ScreenBase register data from bypass used 1 = ScreenBase register data from bypass not used yet.
8	Reserved	✓	✕	0	
9..10	BufferSwap	✓	✓	0	0 = SyncOnFrameBlank 1 = FreeRunning. 2 = LimitToFrameRate 3 = Reserved
11	Stereo	✓	✓	0	0 = Disabled 1 = Enabled.
12	RightEyeCtl	✓	✓	0	0 = Active high 1 = Active low
13	RightFrame	✓	✕	0	0 = Display left frame 1 = Display right frame
14	ExtControl	✓	✕	0	0 = low 1 = high.
15	Reserved	✓	✓	0	Reserved
16..17	SyncMode	✓	✓	0	0 = Independent 1 = Reserved 2 = Reserved 3 = Reserved
18	PatchEnable	✓	✓	0	0 = Off 1 = On
19..20	PixelSize	✓	✓	0	0 = 8 bits 1 = 16 bits 2 = 32 bits 3 = Reserved
21	DisplayDisable	✓	✓	0	0 = Off 1 = On
22..27	PatchOffsetX	✓	✓	0	
28..31	PatchOffsetY	✓	✓	0	

Notes: The *ExtControl* bit (14) drives the Video External Control pin (AT15) directly for use controlling external devices.

VideoOverlayBase0
VideoOverlayBase1
VideoOverlayBase2

Name	Type	Offset	Format
VideoOverlayBase0	Video Overlay Control	0x3120	Bitfield
VideoOverlayBase1	Video Overlay Control	0x3128	Bitfield
VideoOverlayBase2	Video Overlay Control	0x3130	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..25	Address	✓	✓	X	Pixel address.
26..29	Reserved	✓	✗	0	
30..31	MemoryType	✓	✓	X	0 = Framebuffer 1 = Localbuffer 2 = Reserved 3 = Reserved

Notes:

VideoOverlayFieldOffset

Name	Type	Offset	Format
VideoOverlayFieldOffset	Video Overlay Control	0x3170	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	0	
4..27	Offset	✓	✓	X	Scale factor as 12.12 2's complement fixed point value.
28..31	Reserved	✓	✗	0	

Notes:

VideoOverlayFIFOControl

Name	Type	Offset	Format
VideoOverlayFIFOControl	Video Overlay Control	0x3110	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...15	Low	✓	✓	0	Low threshold
16...31	High	✓	✓	0xFF	High threshold

Notes:

VideoOverlayHeight

Name	Type	Offset	Format
VideoOverlayHeight	Video Overlay Control	0x3148	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..11	Height	✓	✓	X	Height of overlay buffer in lines.
12..31	Reserved	✓	✗	0	

Notes:

VideoOverlayIndex

Name	Type	Offset	Format
VideoOverlayIndex	Video Overlay Control	0x3118	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..1	Index	✓	✓	X	Base address register to use when *BufferSync* is Manual
2..30	Reserved	✓	✗	0	
31	Field	✓	✓	X	0 = Odd 1 = Even

Notes:

VideoOverlayMode

Name	Type	Offset	Format
VideoOv rlayMode	Video Overlay Control	0x3108	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0	Enable	✓	✓	0	0 = Off	1 = On
1..3	BufferSync	✓	✓	0	0 = Manual 2 = VideoStreamB	1 = VideoStreamA 3..7 = Reserved
4	FieldPolarity	✓	✓	0	0 = Normal	1 = Invert
5..6	PixelSize	✓	✓	0	0 = 8 bits 2 = 32 bits	1 = 16 bits 3 = Reserved
7..9	ColorFormat	✓	✓	0	0 = RGB8888 2 = RGB551 4 = RGB332 6 = reserved	1 = RGB4444 3 = RGB565 5 = CI8 7 = reserved
10..11	YUV	✓	✓	0	0 = RGB 2 = YUV444	1 = YUV422 3 = Reserved
12	ColorOrder	✓	✓	0	0 = BGR	1 = RGB
13	LinearColor Extension	✓	✓	0	0 = Off	1 = On
14..15	Filter	✓	✓	0	0 = Off 2 = Partial (X with zoom)	1 = Full 3 = Reserved
16..17	DeInterlace	✓	✓	0	0 = Off 2..3 = Reserved	1 = Bob
18..19	PatchMode	✓	✓	0	0 = Off 2..3 = Reserved	1 = On
20..22	Flip	✓	✓	0	0 = Video	1..7 = reserved
23	MirrorX	✓	✓	0	0 = Off	1 = On
24	MirrorY	✓	✓	0	0 = Off	1 = On
25..31	Reserved	✓	✗	0		

Notes:

The following table shows the bit positions of each component in each color format. In Color Index (CI) 8-bit mode the color value is repeated for R, G and B channels.

			Internal Color Channels		
Color Format	**Color Order**	**Name**	**R**	**G**	**B**
0	0	8:8:8:8	8@0	8@8	8@16
1	0	4:4:4:4	4@0	4@4	4@8
2	0	5:5:5:1	5@0	5@5	5@10
3	0	5:6:5	5@0	6@5	5@11
4	0	3:3:2	3@0	3@3	2@6
0	1	8:8:8:8	8@16	8@8	8@0
1	1	4:4:4:4	4@8	4@4	4@0
2	1	5:5:5:1	5@10	5@5	5@0
3	1	5:6:5	5@11	6@5	5@0
4	1	3:3:2	3@5	3@2	2@0
5	1	C18	8@0	8@0	8@0

In YUV422 or YUV444 mode the *ColorFormat* field is ignored. The following bit positions are used:

			Internal Color Channels		
YUV	**Color Order**	**Name**	**Y**	**U**	**V**
0	0	RGB	-	-	-
1	0	YUV444	8@0	8@8	8@16
2	0	YUV422	8@0	8@8	8@8
3	0	Reserved	-	-	-
0	1	RGB	-	-	-
1	1	YUV444	8@16	8@8	8@0
2	1	YUV422	8@8	8@0	8@0
3	1	Reserved	-	-	-

In YUV422 mode the U and V components share the same bits in alternate pixels; U is always in the lower 16 bits and V in the upper 16 bits.

VideoOverlayOrigin

Name	Type	Offset	Format
VideoOverlayOrigin	Video Overlay Control	0x3150	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..11	XOrigin	✓	✓	X	X origin of data to display within source buffer.
12..15	Reserved	✓	✗	0	
16..27	YOrigin	✓	✓	X	Y origin of data to display within source buffer.
28..31	Reserved	✓	✗	0	

Notes:

VideoOverlayShrinkXDelta

Name	Type	Offset	Format
VideoOverlayShrinkXDelta	Video Overlay Control	0x3158	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	0	
4..27	Delta	✓	✓	X	Scale factor as 12.12 2's complement fixed point value.
28..31	Reserved	✓	✗	0	

Notes:

VideoOverlayStatus

Name	Type	Offset	Format
VideoOverlayStatus	Video Overlay Control	0x3178	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	FIFOUnderflow	✓	✓	0	Set by overlay unit, cleared by writing 1.
1..31	Reserved	✕	✕	0	

Notes:

VideoOverlayStride

Name	Type	Offset	Format
VideoOverlayStride	Video Overlay Control	0x3138	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..11	Stride	✓	✓	X	Stride of overlay buffer in pixels.
12..31	Reserved	✓	✕	0	

Notes:

VideoOverlayUpdate

Name	Type	Offset	Format
VideoOverlayUpdate	Video Overlay Control	0x3100	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Enable	✓	✓	0	Set to 1 to enable update, cleared following update.
1..31	Reserved	✓	✕	0	

Notes:

VideoOverlayWidth

Name	Type	Offset	Format
VideoOverlayWidth	Video Overlay Control	0x3140	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..11	Width	✓	✓	X	Width of overlay buffer in pixels.
12..31	Reserved	✓	✗	0	

Notes:

VideoOverlayYDelta

Name	Type	Offset	Format
VideoOverlayYDelta	Video Overlay Control	0x3168	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	0	
4..27	Delta	✓	✓	X	Scale factor as 12.12 2's complement fixed point value.
28..31	Reserved	✓	✗	0	

Notes:

VideoOverlayZoomXDelta

Name	Type	Offset	Format
VideoOverlayZoomXDelta	Video Overlay Control	0x3160	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	0	
4..16	Delta	✓	✓	X	Scale factor as 1.12 unsigned
17..31	Reserved	✓	✗	0	

Notes:

VsEnd

Name	Type	Offset	Format
VsEnd	Video Control *Control register*	0x3050	Integer

Bits	Name	Read	Write	Reset	Description
10..0	VsEnd	✓	✓	X	First scanline out of vertical sync - 1
31..11	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

VsStart

Name	Type	Offset	Format
VsStart	Video Control *Control register*	0x3048	Integer

Bits	Name	Read	Write	Reset	Description
0..10	VsStart	✓	✓	X	First scanline in vertical sync – 1.
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

VTotal

Name	Type	Offset	Format
Vtotal	Video Control *Control register*	0x3038	Integer

Bits	Name	Read	Write	Reset	Description
0..10	VTotal	✓	✓	X	Last scanline on screen, including vertical blank period.
11..31	Reserved	✓	✗	0	

Notes: Reset = 000.000.000.000.000.0XXX.XXX.XXX

4.9 DMA Arbiter

The DMA Arbiter unit has the following functions:

- Take requests to read data from or write data to system memory.
- Convert logical addresses to physical addresses.
- Arbitrate between requests.
- Issue a transaction to the PCI/AGP bus master interface.
- For write transactions, route data from the source to the bus interface.
- For read transactions, route data from the bus interface to its destination.

Read request sources include the BypassDMA controller, the RasterizerPage controller, the InputFifo DMA controller and the geometry processor core. Write sources are the MidStream and DownStream[1] controllers, more commonly called the Bypass Write and GPOut Write controllers.

No DMA controller is permitted to request read data unless it has space available to consume the data, nor to generate a write request unless all the data for the transfer is guaranteed to be available. This simplifies the design of the Arbiter since there is no need to track the available space or data for each request source. However the input and output FIFOs for the Arbiter make full use of flow control (with full/empty flags and write/read strobes) to ensure no data is lost or incorrectly presumed available. This is essential when clock speeds can be dynamically changed, for example to support power management.

The DMA Arbiter also provides 7 interrupt signals that are reflected in bits 16 to 22 of the **IntEnable** and **IntFlags** registers. These signals map as follows to the bits in the interrupt registers.

DMA Arbiter Interrupts

DMA Arbiter Interrupt Signal	Interrupt Register bit definition
RasteriserPageFault	Bit 22: Rasteriser Page Page Fault
MemoryUnitPageFault	Bit 21: Memory Unit Page Fault
InputFifoPageFault	Bit 20: Input FIFO Page Fault
GPPageFault	Bit 19: GP In Page Fault
GPOutWritePageFault (DownStreamPageFault)	Bit 18: GP Out Page Fault
BypassWritePageFault (MidStreamPageFault)	Bit 17: Bypass Write DMA Page Fault
BypassDMAPageFault	Bit 16: Bypass Read DMA Page Fault

[1] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

4.9.1 Direct Registers

ArbiterRequestControl

Name	Type	Offset	Format
ArbiterR(questControl	Address Direct	0x00	bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	BypassDMA Enable	✓	✓	0	0 = off; 1 = bus master requests enabled for BypassDMA reads.
1	RasterizerPage Enable	✓	✕	0	0 = off; 1 = bus master requests enabled for RasterizerPage reads.
2	InputFifo Enable	✓	✓	0	0 = off; 1 = bus master requests enabled for InputFifo reads.
3	Reserved [MemoryUnit Enable]	✓	✕	0	0 = off; 1 = bus master requests enabled for MemoryUnit reads.
4	MidStream Enable	✓	✓	0	0 = off; 1 = bus master requests enabled for writes from MidStream (Bypass) bus.
5	DownStream Enable	✓	✕	0	0 = off; 1 = bus master requests enabled for writes from DownStream bus.
6	GPEnable	✓	✓	0	0 = off; 1 = bus master requests enabled for Geometry Processor reads.
7	Enable	✓	✕	0	0 = bus master requests disabled; 1 = bus master requests enabled as above.
8	MidStream Protocol [Reserved]	✓	✓	0	0 = PCI; 1 = AGP. Defines protocol for writes from MidStream bus.
9	DownStream Protocol [Reserved]	✓	✕	0	0 = PCI; 1 = AGP. Defines protocol for writes from DownStream bus.
10-31	Reserved	✓	✓	0	0 = Disabled 1 = Enabled

Notes: The MidStream and DownStream[2] controllers are commonly called the Bypass Write and GPOut Write controllers

[2] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

AddressTranslationControl

Name	Type	Offset	Format
AddressTranslationControl	Arbiter Direct	0x08	bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	BypassDMAEnable	✓	✓	0	0 = off; 1 = address translation enabled for BypassDMA reads.
1	RasterizerPageEnable	✓	✗	0	0 = off; 1 = address translation enabled for RasterizerPage reads.
2	InputFifoEnable	✓	✓	0	0 = off; 1 = address translation enabled for InputFifo reads.
3	Reserved [MemoryUnitEnable]	✓	✗	0	0 = off; 1 = address translation enabled for MemoryUnit reads.
4	MidStreamEnable	✓	✓	0	0 = off; 1 = address translation enabled for writes from MidStream bus.
5	DownStreamEnable	✓	✗	0	0 = off; 1 = address translation enabled for writes from DownStream bus.
6	GPEnable	✓	✓	0	0 = off; 1 = address translation enabled for Geometry Processor reads.
7	Enable	✓	✗	0	0 = address translation disabled; 1 = address translation enabled as above.
8	PhysicalAddressFormat	✓	✓	0	0 = 32 bits; 1 = 64 bits.
9	TableFormat	✓	✗	0	0 = 32 bits; 1 = 64 bits.
10	PageReadProtocol	✓	✓	0	0 = PCI; 1 = AGP. Defines protocol for reading page table entries.
11-31	Reserved	✓	✗	0	

Notes: The MidStream and DownStream[3] controllers are commonly called the Bypass Write and GPOut Write controllers

[3] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

PhysicalAddressOffset

Name	Type	Offset	Format
PhysicalAddressOffset	Arbiter Direct	0x18	Address

Control register

Bits	Name	Read	Write	Reset	Description
0 - 31	Address	✓	✓	0	Bits 32-63 of address if translation disabled and 64-bit addressing enabled 0 = Disabled 1 = Enabled

Notes:

PageTableBaseAddressLower

Name	Type	Offset	Format
PageTablBaseAddress Lower	Arbiter Direct	0x20	Address

Control register

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✓	0	0 = Disabled 1 = Enabled
2-31	Address	✓	✗	0	Base address of page table in system memory.

Notes:

PageTableBaseAddressUpper

Name	Type	Offset	Format
PageTabl BaseAddress Upper	Arbiter direct	0x28	Address

Control register

Bits	Name	Read	Write	Reset	Description
0-31	Address	✓	✓	0	Upper 32 bits of address when **PhysicalAddressFormat** set to 64 bits.

Notes:

MinLogicalPage

Name	Type	Offset	Format
MinLogic ılPage	Arbiter direct	0x30	Address

Control register

Bits	Name	Read	Write	Reset	Description
0-11	Reserved	✓	✓	0	
12-31	Address	✓	✕	0	Lowest valid logical page address.

Notes:

MaxLogicalPage

Name	Type	Offset	Format
MaxLogicalPage	Arbiter direct *Control register*	0x38	address

Bits	Name	Read	Write	Reset	Description
0-11	Reserved	✓	✓	0	
12-31	Address	✓	✗	0	Highest valid logical page address.

Notes:

Flush

Name	Type	Offset	Format
Flush	Arbiter direct *Command*	0x40	

Bits	Name	Read	Write	Reset	Description
0…31	Reserved	✓	✓	0	Write to register initiates AGP Flush operation; read returns zero.

Notes:

InvalidateTLB

Name	Type	Offset	Format
Invalidate TLB	Arbiter direct *Control register*	0x48	Bitfield

Bits	Name	Read	Write	Reset	Description
0	BypassDMA	✓	✓	0	0 = no effect; 1 = invalidate TLB entry for BypassDMA reads.
1	RasterizerPage	✓	✗	0	0 = no effect; 1 = invalidate TLB entry for RasterizerPage reads.
2	InputFifo	✓	✓	0	0 = no effect; 1 = invalidate TLB entry for InputFifo reads.
3	Reserved [MemoryUnit]	✓	✗	0	0 = no effect; 1 = invalidate TLB entry for MemoryUnit reads.
4	MidStream	✓	✓	0	0 = no effect; 1 = invalidate TLB entry for MidStream writes.
5	DownStream	✓	✗	0	0 = no effect; 1 = invalidate TLB entry for DownStream writes.
6	GP	✓	✓	0	0 = no effect; 1 = invalidate all TLB entries for Geometry Processor reads.
7-31	Reserved	✓	✗	0	

Notes: The MidStream and DownStream[4] controllers are commonly called the Bypass Write and GPOut Write controllers

[4] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

Restart

Name	Type	Offset	Format
Restart	Arbiter direct *Control register*	0x50	bitfield

Bits	Name	Read	Write	Reset	Description
0	BypassDMA	✓	✓	0	Write to register with data bit 0 set restarts BypassDMA table lookup.
1	RasterizerPage	✓	✗	0	Write to register with data bit 1 set restarts RasterizerPage table lookup.
2	InputFifo	✓	✓	0	Write to register with data bit 2 set restarts InputFifo table lookup.
3	Reserved [MemoryUnit]	✓	✗	0	Write to register with data bit 3 set restarts MemoryUnit table lookup.
4	MidStream	✓	✓	0	Write to register with data bit 4 set restarts MidStream table lookup.
5	DownStream	✓	✗	0	Write to register with data bit 5 set restarts DownStream table lookup.
6	GP	✓	✓	0	Write to register with data bit 6 set restarts Geometry Processor table lookup.
7-31	Reserved	✓	✗	0	Read always returns zero for all 32 bits of this register.

Notes: The MidStream and DownStream[5] controllers are commonly called the Bypass Write and GPOut Write controllers

[5] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

PageFaultSource

Name	Type	Offset	Format
PageFaul Source	Arbiter direct	0x60	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	BypassDMA	✓	✓	0	1 = page fault signalled during BypassDMA address translation.
1	RasterizerPage	✓	✓	0	1 = page fault signalled during RasterizerPage address translation.
2	InputFifo	✓	✓	0	1 = page fault signalled during InputFifo address translation.
3	Reserved [MemoryUnit]	✓	✓	0	1 = page fault signalled during MemoryUnit address translation.
4	MidStream	✓	✓	0	1 = page fault signalled during MidStream address translation.
5	DownStream	✓	✓	0	1 = page fault signalled during DownStream address translation.
6	GP	✓	✓	0	1 = page fault signalled during Geometry Processor address translation.
7-12	GPID	✓	✓	0	ID of faulting Geometry Processor unit.
13-31	Reserved	✓	✕	0	

Notes: The MidStream and DownStream[6] controllers are commonly called the Bypass Write and GPOut Write controllers

[6] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

ArbiterStatus

Name	Type	Offset	Format
ArbiterSt tus	Arbiter direct *Control register*	0x68	bitfield

Bits	Name	Read	Write	Reset	Description
0	BypassDMA	✓	✓	0	0 = idle; 1 = BypassDMA read in progress.
1	RasterizerPage	✓	✓	0	0 = idle; 1 = RasterizerPage read in progress.
2	InputFifo	✓	✓	0	0 = idle; 1 = InputFifo read in progress.
3	Reserved [MemoryUnit]	✓	✓	0	0 = idle; 1 = MemoryUnit read in progress.
4	MidStream	✓	✓	0	0 = idle; 1 = MidStream write in progress.
5	DownStream	✓	✓	0	0 = idle; 1 = DownStream write in progress.
6	GP	✓	✓	0	0 = idle; 1 = Geometry Processor read in progress.
7	PCIWrite	✓	✓	0	0 = idle; 1 = PCI Master write in progress.
8	AGPWrite [Reserved]	✓	✓	0	0 = idle; 1 = AGP Master write in progress.
9	AGPFlush	✓	✓	0	0 = idle; 1 = AGP Flush operation in progress.
10-31	Reserved	✓	✗	0	

Notes:

ArbiterReadIndex

Name	Type	Offset	Format
ArbiterR adIndex	Arbiter direct *Control register*	0x70	integer

Bits	Name	Read	Write	Reset	Description
0-7	Index	✓	✓	0	Index of 32-bit Arbiter status register or TLB entry to read.
8-31	Reserved	✓	✗		

Notes:

ArbiterReadData

Name	Type	Offset	Format
ArbiterReadData	Arbiter direct *Control register*	0x78	integer

Bits	Name	Read	Write	Reset	Description
0…31	Data	✓	✓	0	Read returns data selected by the contents of the ArbiterReadIndex register.

Notes:

ProfileControl

Name	Type	Offset	Format
ProfileControl	Arbiter direct *Control register*	0x80	bitfield

Bits	Name	Read	Write	Reset	Description
0…2	PipeSelect	✓	✓	0	Select address translation pipe to be profiled: 0 = BypassDMA 1 = RasterizerPage 2 = InputFifo 3 = undefined [MemoryUnit] 4 = MidStream 5 = DownStream 6 = Geometry Processor 7 = undefined.
3…31	Reserved	✓	✕	0	

Notes:

4.9.2 Indirect Registers

The page fault registers and TLB entries are accessed indirectly. An index selecting which register to read is first loaded into the **ArbiterReadIndex** register, and then the data read back from the **ArbiterReadData** register.

Reading back an indirect register using an undefined value in the **ArbiterReadIndex** register will return zero.

BypassPageFaultAddress

Name	Type	Index	Format
BypassPageFaultAddress	Arbiter indirect *Control register*	0x00	address

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✗		
2-31	Address	✓	✓	0	Logical address that caused BypassDMA page fault.

Notes:

RasterizerPageFaultAddress

Name	Type	Index	Format
RasterizerPageFaultAddress	Arbiter indirect *Control register*	0x01	Address

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✗		
2-31	Address	✓	✓	0	Logical address that caused RasterizerPage fault.

Notes:

InputFifoFaultAddress

Name	Type	Index	Format
InputFifoFaultAddress	Arbiter indirect *Control register*	0x02	Address

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✗	0	Logical address that caused InputFifo fault.
2-31	Address	✓	✓		

Notes:

MemoryUnitFaultAddress

Name	Type	Index	Format
MemoryUnitFaultAddress	Arbiter indirect *Command*	0x03	

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✓	0	
2-31	Address	✓	✗		Logical address that caused MemoryUnit fault.

Notes:

MidStreamPageFaultAddress

Name	Type	Index	Format
MidStreamPageFault Address	Arbiter indirect *Control register*	0x04	Address

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✗	0	
2-31	Address	✓	✓		Logical address that caused MidStream page fault.

Notes: The MidStream and DownStream[7] controllers are commonly called the Bypass Write and GPOut Write controllers

[7] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

DownStreamPageFaultAddress

Name	Type	Index	Format
DownStr amPageFault Address	Arbiter indirect	0x05	Address

Control register

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✕	0	.
2-31	Address	✓	✓		Logical address that caused DownStream page fault.

Notes: The MidStream and DownStream[8] controllers are commonly called the Bypass Write and GPOut Write controllers

GPPageFaultAddress

Name	Type	Index	Format
GPPageF ultAddress	Arbiter indirect	0x08	Address

Control register

Bits	Name	Read	Write	Reset	Description
0-1	Reserved	✓	✓	0	
2-31	Address				Logical address that caused Geometry Processor page fault.

Notes:

[8] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

BypassLogicalPage

Name	Type	Index	Format
BypassLogicalPage	Arbiter indirect *Control register*	0x10	Address

Bits	Name	Read	Write	Reset	Description
0-11	Reserved	✓	✓	0	
12-31	LogicalPage				Logical page associated with TLB for BypassDMA source.

Notes:

BypassStatus

Name	Type	Index	Format
BypassStatus	Arbiter indirect *Control register*	0x11	Address

Bits	Name	Read	Write	Reset	Description
0	Loaded	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for BypassDMA source
1-31	Reserved				

Notes:

BypassPageEntryLower

Name	Type	Index	Format
BypassPageEntryLower	Arbiter indirect *Control register*	0x12	Address

Bits	Name	Read	Write	Reset	Description
0-31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry for BypassDMA source.

Notes:

BypassPageEntryUpper

Name	Type	Index	Format
BypassPa ;eEntryUpper	Arbiter indirect *Control register*	0x13	Address

Bits	Name	Read	Write	Reset	Description
0-31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry for BypassDMA source.

Notes:

RasterizerLogicalPage

Name	Type	Index	Format
Rasterize: LogicalPage	Arbiter indirect *Control register*	0x14	Address

Bits	Name	Read	Write	Reset	Description
0-11	Reserved	✓	✕	0	
12-31	LogicalPage	✓	✓		Logical page associated with TLB for RasterizerPage source.

Notes:

RasterizerPageStatus

Name	Type	Index	Format
Rasterize: PageStatus	Arbiter indirect *Control register*	0x15	Address

Bits	Name	Read	Write	Reset	Description
0	Loaded	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for RasterizerPage source
1-31	Reserved	✓	✕		

Notes:

RasterizerPageEntryLower

Name	Type	Index	Format
Rasterize: PageEntryLower	Arbiter indirect *Control register*	0x40	Address

Bits	Name	Read	Write	Reset	Description
0-31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry for RasterizerPage source.

Notes:

RasterizerPageEntryUpper

Name	Type	Index	Format
Rasterize: PageEntryUpper	Arbiter indirect *Control register*	0x40	Address

Bits	Name	Read	Write	Reset	Description
0-31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry for RasterizerPage source.

Notes:

InputFifoLogicalPage

Name	Type	Index	Format
InputFifo LogicalPage	Arbiter indirect *Control register*	0x40	Bitfield

Bits	Name	Read	Write	Reset	Description
0…11	Reserved	✓	✕	0	Write to register initiates AGP Flush operation; read returns zero.
12…31	LogicalPage	✓	✓		Logical page associated with TLB for InputFifo source

Notes:

InputFifoStatus

Name	Type	Index	Format
InputFifo Status	Arbiter indirect *Control register*	0x19	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Loaded	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for InputFifo source
1…31	Reserved	✓	✗	0	

Notes:

InputFifoEntryLower

Name	Type	Index	Format
InputFifo EntryLower	Arbiter indirect *Control register*	0x1A	0x22

Bits	Name	Read	Write	Reset	Description
0-31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry for InputFifo source.

Notes:

InputFifoEntryUpper

Name	Type	Index	Format
InputFifo EntryUpper	Arbiter indirect *Control register*	0x1B	0x23

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry for InputFifo source.

Notes:

MemoryUnitLogicalPage

Name	Type	Index	Format
MemoryUnitLogicalPage	Arbiter indirect *Command*	0x1C	Bitfield

Bits	Name	Read	Write	Reset	Description
0…11	Reserved	✓	✕	0	
12…31	Logical page	✓	✓	0	Logical page associated with TLB for MemoryUnit source.

Notes:

MemoryUnitStatus

Name	Type	Offset	Format
MemoryUnitStatus	Arbiter indirect *Control*	0x1D	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Loaded	✓	✓	0	0 = not loaded 1 = TLB entry loaded for MemoryUnit source
1…31	Reserved	✓	✕	0	

Notes:

MemoryUnitEntryLower

Name	Type	Offset	Format
MemoryUnitEntryLower	Arbiter indirect *Control*	0x1E	Address

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry for MemoryUnit source.

Notes:

MemoryUnitEntryUpper

Name	Type	Offset	Format
MemoryUnitEntryUpper	Arbiter indirect *Control*	0x1F	Address

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry for MemoryUnit source.

Notes:

MidStreamLogicalPage

Name	Type	Index	Format
MidStreamLogicalPage	Arbiter indirect *Command*	0x20	Bitfield

Bits	Name	Read	Write	Reset	Description
0…11	Reserved	✓	✗	0	
12…31	Logical page	✓	✓	0	Logical page associated with TLB for Midstream source.

Notes: The MidStream and DownStream[9] controllers are commonly called the Bypass Write and GPOut Write controllers

[9] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

MidStreamStatus

Name	Type	Offset	Format
MidStreamStatus	Arbiter indirect *Control*	0x21	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Loaded	✓	✓	0	0 = not loaded 1 = TLB entry loaded for Midstream source
1…31	Reserved	✓	✗	0	

Notes: The MidStream and DownStream[10] controllers are commonly called the Bypass Write and GPOut Write controllers

MidStreamPageEntryLower

Name	Type	Offset	Format
MidStreamPageEntryLower	Arbiter indirect *Control*	0x22	Address

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry for Midstream source.

Notes: The MidStream and DownStream[11] controllers are commonly called the Bypass Write and GPOut Write controllers

[10] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

[11] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

MidStreamPageEntryUpper

Name	Type	Offset	Format
MidStreamPageEntryUpper	Arbiter indirect *Control*	0x23	Address

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry for Midstream source.

Notes: The MidStream and DownStream[12] controllers are commonly called the Bypass Write and GPOut Write controllers

DownStreamLogicalPage

Name	Type	Index	Format
DownStreamLogicalPage	Arbiter indirect *Command*	0x24	Bitfield

Bits	Name	Read	Write	Reset	Description
0…11	Reserved	✓	✗	0	
12…31	Logical page	✓	✓	0	Logical page associated with TLB for source.

Notes:

[12] MidStream write requests and data are supplied by the Bypass DMA Controller, and DownStream write requests by the GPOut DMA controller.

DownStreamStatus

Name	Type	Offset	Format
DownStr amStatus	Arbiter indirect *Control*	0x25	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Loaded	✓	✓	0	0 = not loaded 1 = TLB entry loaded for Downstream source
1…31	Reserved	✓	✕	0	

Notes:

DownStreamPageEntryLower

Name	Type	Offset	Format
DownStr amPageEntryLow er	Arbiter indirect *Control*	0x26	Address

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry for Downstream source.

Notes:

DownStreamPageEntryUpper

Name	Type	Offset	Format
DownStr amPageEntryUpp er	Arbiter indirect *Control*	0x27	Address

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry for Downstream source.

Notes:

ProfileCountStateRun

Name	Type	Index	Format
ProfileCountStateRun	Arbiter indirect *Command*	0x30	Integer

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of cycles when address translation running normally.

Notes:

ProfileCountStateMiss

Name	Type	Offset	Format
ProfileCountStateMiss	Arbiter indirect *Control*	0x31	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Count	✓	✓	0	Count of cycles when selected address pipe in “miss” state.

Notes:

ProfileCountStateBlock

Name	Type	Offset	Format
ProfileCountStateBlock	Arbiter indirect *Control*	0x32	Address

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of cycles when next request held up behind a TLB miss.

Notes:

ProfileCountStateFault

Name	Type	Offset	Format
ProfileCountStateFault	Arbiter indirect *Control*	0x33	Integer

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of cycles when selected address pipe in "fault" state.

Notes:

ProfileCountHits

Name	Type	Offset	Format
ProfileCountHits	Arbiter indirect *Command*	0x34	Integer

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of translated requests issued following a TLB hit.

Notes:

ProfileCountMisses

Name	Type	Offset	Format
ProfileCountMisses	Arbiter indirect *Command*	0x35	Integer

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of page table reads issued following a TLB miss.

Notes:

ProfileCountPhysical

Name	Type	Offset	Format
ProfileCountPhysical	Arbiter indirect *Command*	0x36	Integer

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of direct physical requests issued for selected pipe

Notes:

ProfileCountCyclesEnabled

Name	Type	Offset	Format
ProfileCountCyclesEnabled	Arbiter indirect *Command*	0x37	Integer

Bits	Name	Read	Write	Reset	Description
0…31	Count	✓	✓	0	Count of cycles when the global request enable is true.

Notes:

GPStatusA

Name	Type	Offset	Format
GPStatus A	Arbiter indirect *Command*	0x3E	Bitfield

Bits	Name	Read	Write	Reset	Description
0	Loaded0	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 0
1	Loaded1	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 1
2…29	Loaded [n]	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit [n]
30	Loaded30	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 30
31	Loaded31	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 31

Notes: The indirect register map defines one TLB entry for each of the 64 possible Geometry Processor units. If there are fewer than 64 GP units, then registers or fields defined for non-existent units should read back as zero.

GPStatusB

Name	Type	Offset	Format
GPStatus B	Arbiter indirect *Command*	0x3F	Bitfield

Bits	Name	Read	Write	Reset	Description
32	Loaded0	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 0
33	Loaded1	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 1
34…29	Loaded [n]	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit [n]
30	Loaded30	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 30
31	Loaded63	✓	✓	0	0 = not loaded; 1 = TLB entry loaded for GP Unit 31

Notes: The indirect register map defines one TLB entry for each of the 64 possible Geometry Processor units. If there are fewer than 64 GP units, then registers or fields defined for non-existent units should read back as zero.

GPLogicalPage0 … GPLogicalPage63

Name	Type	Offset	Format
GPLogicalPage0 … GPLogicalPage63	Arbiter indirect *Command*	0x40…0x7F	Address

Bits	Name	Read	Write	Reset	Description
0…11	Reserved	✓	✕	0	
12…31	Logicalk page	✓	✓	0	Logical page associated with TLB for GP Unit0 … GP Unit63.

Notes:

GPPageEntryLower[0…63]

Name	Type	Offset	Format
GPPageEntryLower[0…63]	Arbiter indirect *Command*	0x80	

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Lower	✓	✓	0	Lower 32 bits of page table entry from TLB for GP Units 0…63.

Notes:

GPPageEntryUpper[0…63]

Name	Type	Offset	Format
GPPageEntryUpper[0…63]	Arbiter indirect *Command*	0x81	

Bits	Name	Read	Write	Reset	Description
0…31	PageEntry Upper	✓	✓	0	Upper 32 bits of page table entry from TLB for GP Units 0…63.

Notes:

4.10 Region 0 RAMDAC

Reads and writes to the RAMDAC are forwarded to the RAMDAC Interface Unit. The address for this unit is formed from bits 6 down to 3 of the incoming bus address. The bottom two bits are discarded to give a 5-bit address aligned to a 32-bit boundary. This address is accompanied by four byte enables, and 32 bits of write data if required.

4.10.1 Direct RAMDAC Registers (0x4000-0x4FFF)

RDIndexControl

Name	Type	Offset	Format
RDIndexControl	RAMDAC Control	0x4038	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	AutoIncrement	✓	✓	0	0 = Disabled 1 = Enabled
1..7	Reserved	✓	✗	0	

Notes: The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary.

RDIndexedData

Name	Type	Offset	Format
RDIndexedData	RAMDAC Control	0x4030	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Data	✓	✓	X	

Notes:
1. A read or write to this register will access the register pointed to by the **RDIndex** register. Following a read or write to this register, the index will be incremented if *AutoIncrement* is enabled in **RDIndexControl**.
2. The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary

RDIndexHigh

Name	Type	Offset	Format
RDIndex High	RAMDAC Control	0x4028	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Index	✓	✓	000.0xxx	
3..7	Reserved	✓	✗	0	

Notes:
1. This register, with RDIndexLow, selects the register that will be accessed when the RDIndexedData register is written or read.
2. The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary

RDIndexLow

Name	Type	Offset	Format
RDIndex Low	RAMDAC Control	0x4020	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Index	✓	✓	X	

Notes:
1. This register, with **RDIndexHigh**, selects the register that will be accessed when the **RDIndexedData** register is written or read.
2. The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary

RDPaletteData

Name	Type	Offset	Format
RDPalett Data	RAMDAC Control	0x4008	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Data	✓	✓	X	

Notes:
1. If the color resolution is 6 bits, bits 6 and 7 are returned as zero for reads and ignored for writes. In this mode, bits 0 to 5 are read from, or written to, bits 2 to 7 of the palette. A read auto-increments **RDPaletteReadAddress** and **RDPaletteWriteAddress**, whereas a write autoincrements the **RDPallettWriteAddress** only.
2. The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary.

RDPaletteReadAddress

Name	Type	Offset	Format
RDPalett ReadAddress	RAMDAC Control	0x4018	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Address	✓	✓	X	Index of palette entry to be read.

Notes: The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary.
If this register is read, its operation is determined by the state of the *LastReadAddress* bit in the **RDMiscControlA** register.

RDPaletteWriteAddress

Name	Type	Offset	Format
RDPalett WriteAddress	RAMDAC Control	0x4000	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Address	✓	✓	0x0000	

Notes: The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary.

RDPixelMask

Name	Type	Offset	Format
RDPixelMask	RAMDAC Control	0x4010	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Mask	✓	✓	X	

Notes:
1. The contents of this register is ANDed with the index into the color palette. The same mask is applied separately to red, green, and blue components.
2. The register is accessed directly by reading or writing to the defined address. It is a byte wide and set on an 8 byte boundary in the PCI address range. When accessed from the SVGA it is set on a byte boundary

4.10.2 Indirect RAMDAC Registers (0x200-0xFFF)

These registers are accessed indirectly by first loading the index into the **RDIndexLow** and **RDIndexHigh** registers, and then reading or writing the **RDIndexedData** register

RDCheckControl

Name	Type	Offset	Format
RDCheckControl	RAMDAC Control	0x018	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Pixel	✓	✓	0	Set to start checksum, cleared when complete. 0 = Disabled 1 = Enabled
1	LUT	✓	✓	0	Set to start checksum, cleared when complete. 0 = Disabled 1 = Enabled
2..7	Reserved	✓	✕	0	

Notes:

RDCheckLUTBlue

Name	Type	Offset	Format
RDCheckLUTBlue	RAMDAC Control	0x01E	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	CheckSum	✓	✕	X	Checksum for blue component after look-up table.

Notes:

RDCheckLUTGreen

Name	Type	Offset	Format
RDCheckLUTGreen	RAMDAC Control	0x01D	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	CheckSum	✓	✕	X	Checksum for green component after look-up table.

Notes:

RDCheckLUTRed

Name	Type	Offset	Format
RDCheckLUTRed	RAMDAC Control	0x01C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	CheckSum	✓	✕	X	Checksum for red component after look-up table.

Notes:

RDCheckPixelBlue

Name	Type	Offset	Format
RDCheckPixelBlue	RAMDAC Control	0x01B	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	CheckSum	✓	✕	X	Checksum for blue component after pixel processing.

Notes:

RDCheckPixelGreen

Name	Type	Offset	Format
RDCheckPixelGreen	RAMDAC Control	0x01A	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	CheckSum	✓	✕	X	Checksum for green component after pixel processing.

Notes:

RDCheckPixelRed

Name	Type	Offset	Format
RDCheckPixelRed	RAMDAC Control	0x019	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	CheckSum	✓	✕	X	Checksum for red component after pixel processing.

Notes:

RDColorFormat

Name	Type	Offset	Format
RDColorFormat	RAMDAC Control	0x004	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..4	ColorFormat	✓	✓	X	See table below
5	RGB	✓	✓	X	Color ordering, see table below.
6	LinearColorExtension	✓	✓	X	0 = Disabled - pad low order bits of components less than 8 bits with zeros. 1 = Enabled - linearly extend low order bits of components less than 8 bits.
7	Reserved	✓	✗	0	

Notes: The table below shows the bit positions for each color format specified. The color format is defined in the form number of bits @ bit position, where the bit position defines the first bit of the component with sucessive bits at incresing bit positions.

			Internal Color Channels			
ColorFormat	RGB	Name	R	G	B	O
0	0	8:8:8:8	8@0	8@8	8@16	8@24
1	0	5:5:5:1Front	5@0	5@5	5@10	1@15
2	0	4:4:4:4	4@0	4@4	4@8	4@12
3	0	Reserved	8@0	8@8	8@16	8@24
4	0	Reserved	8@0	8@8	8@16	8@24
5	0	3:3:2Front	3@0	3@3	2@6	0
6	0	3:3:2Back	3@8	3@11	2@14	0
7	0	Reserved	8@0	8@8	8@16	8@24
8	0	Reserved	8@0	8@8	8@16	8@24
9	0	2:3:2:1Front	2@0	3@2	2@5	1@7
10	0	2:3:2:1Back	2@8	3@10	2@13	1@15
11	0	2:3:2FrontOff	2@0	3@2	2@5	0
12	0	2:3:2BackOff	2@8	3@10	2@13	0
13	0	5:5:5:1Back	5@16	5@21	5@26	1@31
14	0	CI8	-	-	-	-
15	0	Reserved	8@0	8@8	8@16	8@24
16	0	5:6:5Front	5@0	6@5	5@11	0
17	0	5:6:5Back	5@16	6@21	5@27	0
18..31	0	Reserved	8@0	8@8	8@16	8@24
0	1	8:8:8:8	8@16	8@8	8@0	8@24
1	1	5:5:5:1Front	5@10	5@5	5@0	1@15

			Internal Color Channels			
ColorFormat	**RGB**	**Name**	**R**	**G**	**B**	**O**
2	1	4:4:4:4	4@8	4@4	4@0	4@12
3	1	Reserved	8@16	8@8	8@0	8@24
4	1	Reserved	8@16	8@8	8@0	8@24
5	1	3:3:2Front	3@5	3@2	2@0	0
6	1	3:3:2Back	3@13	3@10	2@8	0
7, 8	1	Reserved	8@16	8@8	8@0	8@24
9	1	2:3:2:1Front	2@5	3@2	2@0	1@7
10	1	2:3:2:1Back	2@13	3@10	2@8	1@15
11	1	2:3:2FrontOff	2@5	3@2	2@0	0
12	1	2:3:2BackOff	2@13	3@10	2@8	0
13	1	5:5:5:1Back	5@26	5@21	5@16	1@31
14	1	CI8	-	-	-	-
15	1	Reserved	8@16	8@8	8@0	8@24
16	1	5:6:5Front	5@11	6@5	5@0	0
17	1	5:6:5Back	5@27	6@21	5@16	0
18..31	1	Reserved	8@16	8@8	8@0	8@24

RDCursorControl

Name	**Type**	**Offset**	**Format**
RDCursorControl	RAMDAC Control	0x006	Bitfield

Control register

Bits	**Name**	**Read**	**Write**	**Reset**	**Description**
0	DoubleX	✓	✓	0	0 = Disabled. 1 = Enabled.
1	DoubleY	✓	✓	0	0 = Disabled. 1 = Enabled.
2	Readback Position	✓	✓	0	0 = Disabled - readback last value written. 1 = Enabled - readback position in use.
3..7	Reserved	✓	✗	0	

Notes: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register

RDCursorHotSpotX

Name	Type	Offset	Format
RDCursorHotSpotX	RAMDAC Control	0x00B	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..5	X	✓	✓	X	X position of hot spot in cursor.
6..7	Reserved	✓	✗	0	

Notes:

RDCursorHotSpotY

Name	Type	Offset	Format
RDCursorHotSpotY	RAMDAC Control	0x00C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..5	Y	✓	✓	X	Y position of hot spot in cursor.
6..7	Reserved	✓	✗	0	

Notes:

RDCursorMode

Name	Type	Offset	Format
RDCursorMode	RAMDAC Control	0x005	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	CursorEnable	✓	✓	0	0 = Disabled. 1 = Enabled.
1..3	Format	✓	✓	0	0 = 64x64 (2 bits per entry, partitions 0, 1, 2, and 3). 1 = 32x32 (2 bits per entry, partition 0). 2 = 32x32 (2 bits per entry, partition 1). 3 = 32x32 (2 bits per entry, partition 2). 4 = 32x32 (2 bits per entry, partition 3). 5 = 32x32 (4 bits per entry, partitions 0 and 1). 6 = 32x32 (4 bits per entry, partitions 2 and 3).
4..5	Type	✓	✓	0	0 = Microsoft Windows. 1 = X Windows 2 = 3 Color 3 = 15 color
6	ReversePixel Order	✓	✓	0	0 = Disabled (incrementing pixel index goes left to right on screen). 1 = Enabled (incrementing pixel index goes right to left on screen).
7	Reserved	✓	✕	0	

Notes: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the *RDIndexedData* register

RDCursorPalette[0…44]

Name	Type	Offset	Format
RDCursorPalette[0…44]	RAMDAC Control	0x303 to 0x32F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Color	✓	✓	xxxx.xxxx	RGB for 15 cursor colors

Notes: Stores the red, green, and blue color components for 15 cursor colors. These index from 1 to 15.

RDCursorPattern[0…1023]

Name	Type	Offset	Format
RDCursorPattern[0…1023]	RAMDAC Control	0x400 to 0x7FF	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Pattern	✓	✓	xxxx.xxxx	Bitmap (pattern) for the cursor

Notes: These registers are accessed indirectly by first loading the indexes into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register

RDCursorXHigh

Name	Type	Offset	Format
RDCursorXHigh	RAMDAC Control	0x008	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	XHigh	✓	✓	X	The high order bits of the cursor X position.
4..7	Reserved	✓	✗	0	

Notes: Value at readback is determined by the *ReadbackPosition* field in the **RDCursorControl** register.

RDCursorXLow

Name	Type	Offset	Format
RDCursorXLow	RAMDAC Control	0x007	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	XLow	✓	✓	X	The low order bits of the cursor X position.

Notes: Value at readback is determined by the *ReadbackPosition* field in the **RDCursorControl** register

RDCursorYHigh

Name	Type	Offset	Format
RDCursorYHigh	RAMDAC Control	0x00A	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	YHigh	✓	✓	X	The high order bits of the cursor Y position.
4..7	Reserved	✓	✗	0	

Notes: Value at readback is determined by the *ReadbackPosition* field in the **RDCursorControl** register.

RDCursorYLow

Name	Type	Offset	Format
RDCursorYLow	RAMDAC Control	0x009	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	YLow	✓	✓	X	The low order bits of the cursor Y position.

Notes: Value at readback is determined by the *ReadbackPosition* field in the **RDCursorControl** register.

RDDACControl

Name	Type	Offset	Format
RDDACControl	RAMDAC Control	0x002	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	DACPowerCtl	✓	✓	0	This functionality may not be available in a specific implementation. 0 = Normal operation. 1 = LowPower
3	Reserved	✓	✓	0	[SyncOnGreen]
4	BlankRedDAC	✓	✓	0	0 = Disabled. 1 = Enabled.
5	BlankGreen DAC	✓	✓	0	0 = Disabled. 1 = Enabled.
6	BlankBlueDAC	✓	✓	0	0 = Disabled. 1 = Enabled.
7	BlankPedestal	✓	✓	0	0 = Disabled. 1 = Enabled.

Notes:

RDDClk0FeedbackScale

Name	Type	Offset	Format
RDDClk0FeedbackScale	RAMDAC Control	0x202	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	1000.1100	

Notes: The 0 and 1 feedback scales are used with **RDDClk0Prescale** to determine the multiplier applied to an external reference frequency to bring it back to a useful internal value. See Volume 1, Video Unit, in the *GLINT R5 Reference Guide*

RDDClk0PostScale

Name	Type	Offset	Format
RDDClk0PostScale	RAMDAC Control	0x203	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Scale	✓	✓	0000.0100	0 = Divide by 1. 1 = Divide by 2. 2 = Divide by 4. 3 = Divide by 8. 4 = Divide by 16 5..7 = Reserved
3..7	Reserved	✓	✕	0	

Notes:

RDDClk0PostScale
RDDClk1PostScale

Name	Type	Offset	Format
RDDClk0PostScale	RAMDAC Control	0x203	Integer
RDDClk1PostScale	RAMDAC Control	0x206	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Scale	✓	✓	0000.0100	0 = Divide by 1. 1 = Divide by 2 2 = Divide by 4. 3 = Divide by 8. 4 = Divide by 16. 5..7 = Reserved
3..7	Reserved	✓	✕	0	

Notes:

RDDClk2PostScale
RDDClk3PostScale

Name	Type	Offset	Format
RDDClk2PostScale	RAMDAC Control	0x209	Integer
RDDClk3PostScale	RAMDAC Control	0x20C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Scale	✓	✓	xxxx.xxxx	0 = Divide by 1 1 = Divide by 2 2 = Divide by 4 3 = Divide by 8 4 = Divide by 16. 5..7 = Reserved
3..7	Reserved	✓	✗	0	

Notes:

RDDClk0PreScale

Name	Type	Offset	Format
RDDClk0PreScale	RAMDAC Control	0x201	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0000.1010	

Notes: **RDDClk0Prescale** together with **RDDClk0FeedbackScale** are used to determine the multiplier applied to an external reference frequency to bring it back to a useful internal value. See Volume 1, Video Unit, in the *GLINT R5 Reference Guide*.

RDDClk1FeedbackScale

Name	Type	Offset	Format
RDDClk1FeedbackScale	RAMDAC Control	0x24F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	1001.1110	

Notes: The 0 and 1 feedback scales are used with **RDDClk0Prescale** to determine the multiplier applied to an external reference frequency to bring it back to a useful internal value. See Volume 1, Video Unit, in the *GLINT R5 Reference Guide*

RDDClk1PreScale

Name	Type	Offset	Format
RDDClk1PreScale	RAMDAC Control	0x28	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0000.1010	

Notes:

RDDClk2FeedbackScale
RDDClk3FeedbackScale

Name	Type	Offset	Format
RDDClk2FeedbackScale	RAMDAC Control	0x208	Integer
RDDClk3FeedbackScale	RAMDAC Control	0x20B	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	xxxx.xxxx	

Notes:

RDDClk2PreScale
RDDClk3PreScale

Name	Type	Offset	Format
RDDClk2PreScale	RAMDAC Control	0x207	Integer
RDDClk3PreScale	RAMDAC Control	0x20A	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	xxxx.xxxx	

Notes:

RDDClkControl

Name	Type	Offset	Format
RDDClkControl	RAMDAC Control	0x200	bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0	Clock	✓	✓	1	0 = Disable	1 = Enable
1	Lock	✓	✗	X	0 = Not locked.	1 = Locked.
2..3	State	✓	✓	0x2	0 = Drive Low 2 = Run	1 = Drive High 3 = Reserved
4..5	Source	✓	✓	0	0 = PLL 2 = reserved	1 = reserved 3 = External
6..7	Reserved	✓	✗	0		

Notes: Sets the reference frequency source for the chip, either internal PLL or an external sync from the **VideoMergeClockIn** pin. If *Source*=0 then the frequency is controlled by one of 4 sets of PLL registers selected by **VclkRDacCtl**.

RDDClkSetup1

Name	Type	Offset	Format
RDDClkSetup1	RAMDAC Control	0x1F0	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Setup	✓	✓	0010.0111	

Notes:

RDDClkSetup2

Name	Type	Offset	Format
RDDClkSetup2	RAMDAC Control	0x1F1	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Setup	✓	✓	1	
1...4	Delay	✓	✓	0	
5...7	Reserved	✓	✗	0	

Notes: The Delay field sets a value for skew between internal and external clocks which can be configured for board design variations.

RDKClkControl

Name	Type	Offset	Format
RDKClkControl	RAMDAC Control	0x20D	bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0	Clock	✓	✓	1	0 = Disable	1 = Enable
1	Lock	✓	✗	0	0 = Not locked.	1 = Locked.
2..3	State	✓	✓	10b	0 = Drive Low	1 = Drive High
					2 = Run	3 = Low Power
4..6	Source	✓	✓	000b	0 = PClk	1 = HalfPClk
					2 = KPLL	3 = HalfKPLL
					4 = External KClk	
					5 = HalfExternalKPLL	6, 7 = reserved
7	Reserved	✓	✗	0		

Notes:

RDKClkSetup1

Name	Type	Offset	Format
RDDClkSetup2	RAMDAC Control	0x1F2	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Setup	✓	✓	0010.0111	

Notes:

RDDClkSetup2

Name	Type	Offset	Format
RDDClkSetup2	RAMDAC Control	0x1F3	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Setup	✓	✓	1	
1..7	Reserved	✓	✗	0	

Notes:

RDTClkSetup1

Name	Type	Offset	Format
RDTClkSetup1	RAMDAC Control	0x1F4	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Setup	✓	✓	0010.0111	

Notes:

RDTClkSetup2

Name	Type	Offset	Format
RDTClkSetup2	RAMDAC Control	0x1F5	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Setup	✓	✓	1	
1..7	Reserved	✓	✗	0	

Notes:

RDKClkControl

Name	Type	Offset	Format
RDKClkControl	RAMDAC Control	0x20D	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0	Clock	✓	✓	1	0 = Disable	1 = Enable
1	Reserved	✓	✗	0		
2..3	State	✓	✓	10b	0 = Drive Low	1 = Drive High
					2 = Run	3 = Low Power
4..6	Source	✓	✓	0	0 = PClk	1 = PClk/2
					2 = KPLL	3 = KPLL/2
					4 = External Memclock	
					5 = External Memclock/2	
					6 = TPLL	TPLL/2
7	Reserved	✓	✗	0		

Notes:

RDKClkFeedbackScale

Name	Type	Offset	Format
RDKClkFeedbackScale	RAMDAC Control	0x20F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	1000.1100	

Notes:

RDKClkPreScale

Name	Type	Offset	Format
RDKClkPreScale	RAMDAC Control	0x20E	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0000.1010	

Notes:

RDKClkPostScale

Name	Type	Offset	Format
RDKClkPostScale	RAMDAC Control	0x210	Integer

Control register

Bits	Name	Read	Write	Reset	Description	
0..2	Scale	✓	✓	010b	0 = Divide by 1. 2 = Divide by 4. 4 = Divide by 16.	1 = Divide by 2 3 = Divide by 8. 5..7 = Reserved
3..7	Reserved	✓	✗	0		

Notes:

RDMClkControl

Name	Type	Offset	Format
RDMClkControl	RAMDAC Control	0x211	Bitfield

Command register

Bits	Name	Read	Write	Reset	Description	
0	Clock	✓	✓	1	0 = Disable	1 = Enable
1	Reserved	✓	✗	0		
2..3	State	✓	✓	0x2	0 = Drive Low	1 = Drive High
					2 = Run	3 = Low Power
4..6	Source	✓	✓	0x2	0 = PClk 2 = KPLL 4 = ExternalMclk 6 = TPLL	1 = PClk/2 3 = HalfKPLL 5 = Half ExternalMClk 7 = Half TPLL
7	Reserved	✓	✗	0		

Notes: When sourcing from KClk (Source=5 or Source=6) note that the KClk value is always set to the PLL, not to the value determined by the **KclkControl** register.

RDMergeBusControl

Name	Type	Offset	Format
RDMergeBusControl	RAMDAC Control	0x015	Bitfield

Command register

Bits	Name	Read	Write	Reset	Description	
0	DataEnable	✓	✓	0	0 = Off	1 = On
1	SyncEnable	✓	✓	0	0 = Off	1 = On
2	DACStripe	✓	✓	0	0 Off	1 = On
3	PanelStripe	✓	✓	0	0 Off	1 = On
4...6	ClockOut	✓	✓	0	0 = Clock Off 2 = Divide by 2 4 = Divide by 8 6 = Divide bt 32	1 = Divide by 1 3 = Divide by 4 5 = divide by 16 7 = Divide by 64
7	Pipeline	✓	✓	0	0 = Off	1 = On

Notes: Controls video merge bus setup including the external video bus dot clock rate.

RDMergeControl

Name	Type	Offset	Format
RDMerge Control	RAMDAC Control	0x012	Bitfield

Command register

Bits	Name	Read	Write	Reset	Description
0	ClockIn	✓	✓	0	0 = use internal clk 1 = use external clk
1	StrobeOut	✓	✓	0	0 = drive external strobe low 1 = drive VClk times 2 out
2	SyncIn	✓	✓	0	0 = use internal sync 1 = use external sync
3`	SyncOut	✓	✓	0	0 = drive external sync lines inactive 1 = drive syncs out
4	DataIn	✓	✓	0	0 = ignore external data 1 = use external data
5	DataOut	✓	✓	0	0 = drive external video bus to 0 1 = drive data to external video bus
6	DataOr	✓	✓	0	0 = Disabled 1 = Enabled (logically OR merge data + local data)
7	Broadcast	✓	✓	0	0 = Disabled 1 = Enabled

Notes:

RDMergeSkew

Name	Type	Offset	Format
RDMerge Skew	RAMDAC Control	0x013	Bitfield

Command register

Bits	Name	Read	Write	Reset	Description
0...5	Compensation	✓	✓	0.0000 b	Compensation for skew between external clock and sync
6,7	Reserved	✓	✗	0	

Notes: Used to improve accuracy of genlock between chips.

RDMiscControl

Name	Type	Offset	Format
RDMiscControl	RAMDAC Control	0x000	Bitfield

Command register

Bits	Name	Read	Write	Reset	Description
0	HighColor Resolution	✓	✓	0	Controls the width of the palette data. 0 = Disabled - use 6 bits per entry. 1 = Enabled - use 8 bits per entry.
1	PixelDouble	✓	✓	0	0 = Disabled. 1 = Enabled.
2	LastRead Address	✓	✓	0	Controls data returned by read from **RDPaletteReadAddress** register. 0 = Disabled - returns mode of last access to palette (0=write, 3=read). 1 = Enabled - return last palette read address.
3	DirectColor	✓	✓	0	0 = Disabled. 1 = Enabled.
4	Overlay	✓	✓	0	0 = Disabled. 1 = Enabled.
5	PixelDouble Buffer	✓	✓	0	0 = Disabled. 1 = Enabled.
6	BlankToZero	✓	✓	0	Forces data to zero during blank: 0 = Disabled 1 = Enabled
7	StereoDouble Buffer	✓	✓	0	Controls per-pixel double buffering in 5551 color format. 0 = Disabled. 1 = Enabled.

Notes:

RDOverlayKey

Name	Type	Offset	Format
RDOverlyKey	RAMDAC Control	0x00D	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Key	✓	✓	X	Indicates the overlay bit pattern that should be treated as transparent.

Notes:

RDPan

Name	Type	Offset	Format
RDPan	RAMDAC Control	0x00E	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Enable	✓	✓	X	Delay data by 32 bits.
1	Gate	✓	✓	X	Discard first 32 bits on line.
2...7	Reserved	✓	✗	0	

Notes:

RDPanelControl

Name	Type	Offset	Format
RDPanelControl	RAMDAC Control	0x014	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0,1	DataOut	✓	✓	00b	0 = off 1 = PanelSingle (1 pixel port) 2 = PanelDouble (2 pixel port) 3 = reserved
2	Clock	✓	✓	0	0 = fast 1 = slow
3	HsyncCtl	✓	✓	0	0 = ActiveLow 1 = ActiveHigh
4	HsyncOverride	✓	✓	0	0 = off 1 = on
5	VsyncCtl	✓	✓	0	0 = ActiveLow 1 = ActiveHigh
6	VSyncOverride	✓	✓	0	0 = off 1 = on
7	BlankCtl	✓	✓	0	0 = ActiveLow 1 = ActiveHigh

Notes: *Clock* is used to adjust the RAMDAC frequency to the pixel bus - see *Multi-rasterizer Setup* in Volume I.

RDPixelSize

Name	Type	Offset	Format
RDPixelSize	RAMDAC Control	0x003	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Pixel Size	✓	✓	X	0 = 8 bits. 1 = 16 bits. 2 = 32 bits. 3 = Reserved 4 = 24 bits. 5..7 = Reserved
3	ExtPalette	✓	✓	X	0 = disabled (read/write 8-bit palette) 1 = enabled (use extended palette)
4, 5	ExtPalette Mode	✓	✓	X	0 = zero (force all but upper 8 bits of palette entries to zero) 1 = Extend (linearly extend palette entries) 2 = Load (use data already programmed) 3 = reserved
6	ExtDirectColor Mode	✓	✓	0	0 = force all but upper 8 bits of color to 0 1 = Linearly extend palette entries
7...31	Reserved	✓	✗	0	

Notes:

RDSClkControl

Name	Type	Offset	Format
RDSClkControl	RAMDAC Control	0x215	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Clock	✓	✓	1	0 = Disable 1 = Enable
1	Reserved	✓	✗	0	
2..3	State	✓	✓	10	0 = Drive Low 1 = Drive High 2 = Run 3 = Low Power
4..6	Source	✓	✓	0x0	0 = PClk 1 = PClk/2 2 = KPLL 3 = KPLL/2 4 = ExternalSClk 5 = ExternalSClk/2 6 = TPLL 7 = TPLL/2
7	Reserved	✓	✗	0	

Notes: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register

RDScratch

Name	Type	Offset	Format
RDScratch	RAMDAC Control	0x001F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Scratch	✓	✓	X	User definable register for storing state.

Notes:

RDSense

Name	Type	Offset	Format
RDSense	RAMDAC Control	0x00F	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Red	✓	✕	X	
1	Green	✓	✕	X	
2	Blue	✓	✕	X	
3..7	Reserved	✓	✕	0	

Notes:

RDStripe

Name	Type	Offset	Format
RDStripe	RAMDAC Control	0x010	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Size	✓	✓	000b	Size of stripe in scanlines, as power of 2: 0=single line, striping disabled 1=2 lines 2=4 lines 3=8 lines 4=16 lines ...etc.
3,4	Count	✓	✓	00b	Number of rasterizer chips
5...7	Owner	✓	✓	000b	Stripe number owned by this chip

Notes: This register is used to control analog video striping. Analog striping combines the video sognals by current summing analog signals - the RAMDAC is blanked for stripes it doesn't own. This is simple but may produce visible artefacts due to changes in the characteristics of the DACs being used.

RDStripeOffset

Name	Type	Offset	Format
RDStripeOffset	RAMDAC Control	0x011	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Offset	✓	✓	0000.0000	Number of lines to offset stripe calculation, used when panning in Y

Notes: This register is used with **RDStripe** to control analog video striping

RDSyncControl

Name	Type	Offset	Format
RDSyncControl	RAMDAC Control	0x001	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0..2	HSyncCtl	✓	✓	0	0 = Active low at pin. 2 = Tri-state at pin. 4 = Force inactive	1 = Active high at pin. 3 = Force active 5..7 = Reserved
3..5	VSyncCtl	✓	✓	0	0 = Active low at pin. 2 = Tri-state at pin. 4 = Force inactive.	1 = Active high at pin. 3 = Force active. 5..7 = Reserved
6	HSyncOverride	✓	✓	0	0 = As set by HsyncCtl	1 = Force high
7	VSyncOverride	✓	✓	0	0 = As set by VsyncCtl	1 = Force high

Notes:

Decimal values for MSBs used
0 = 0%
64 = 25%
128 = 50%
192 = 75%

RDTClkControl

Name	Type	Offset	Format
RDSClkControl	RAMDAC Control	0x215	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0	Clock	✓	✓	1	0 = Disable	1 = Enable
1	Lock	✓	✓	0	0 = Not locked	1 = Locked
2..3	State	✓	✓	10b	0 = Drive Low 2 = Run	1 = Drive High 3 = Low Power
4..6	Source	✓	✓	000b	0 = PClk 2 = reserved 4 = ExternalTClk 6 = TPLL	1 = PClk/2 3 = reserved/2 5 = ExternalTClk/2 7 = TPLL/2
7	Reserved	✓	✗	0		

Notes:

RDTClkFeedbackScale

Name	Type	Offset	Format
RDTClkFeedbackScale	RAMDAC Control	0x21F	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Value	✓	✓	0000.1100	

Notes:

RDTClkPostScale

Name	Type	Offset	Format
RDTClkI ostScale	RAMDAC Control	0x220	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...2	Scale	✓	✓	010b	0 = divide by 1 1 = divide by 2 2 = divide by 4 3 = divide by 8 4 = divide by 16
3…7	Reserved	✓	✕	0	

Notes:

RDTClkPreScale

Name	Type	Offset	Format
RDTClkI reScale	RAMDAC Control	0x21E	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Value	✓	✓	0000.1010	

Notes:

RDVideoOverlayBlend

Name	Type	Offset	Format
RDVideoOverlayBlend	RAMDAC Control	0x002C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Factor	✓	✓	X	Proportion to blend main image and overlay, enabled by *BlendSrc* field of **RDVideoOverlayControl** mode register. 0 = 0% 0x1 = 25% 0x2 = 59% 0x3 = 75%

Notes:

RDVideoOverlayControl

Name	Type	Offset	Format
RDVideoOverlayControl	RAMDAC Control	0x020	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0	Enable	✓	✓	0	0 = Disabled.	1 = Enabled.
1..2	Mode	✓	✓	00b	0 = MainKey 2 = Always	1 = OverlayKey 3 = Blend
3	DirectColor	✓	✓	0	0 = Disabled.	1 = Enabled.
4	BlendSrc	✓	✓	0	0 = Main.	1 = Register.
5	Key	✓	✓	0	0 = Color.	1 = Alpha.
6	Cursor	✓	✓	0	0 = Disabled.	1 = Enabled.
7	Swap	✓	✓	0	0 = Disabled.	1 = Enabled.

Notes:

RDVideoOverlayKeyB

Name	Type	Offset	Format
RDVideoOverlayKeyB	RAMDAC Control	0x02B	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Blue	✓	✓	X	The blue component for color key checking

Notes:

RDVideoOverlayKeyG

Name	Type	Offset	Format
RDVideoOverlayKeyG	RAMDAC Control	0x02A	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Green	✓	✓	X	The green component for color key checking

Notes:

RDVideoOverlayKeyR

Name	Type	Offset	Format
RDVideoOverlayKeyR	RAMDAC Control	0x029	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Red	✓	✓	xxxx.xxxx	Red component for color key checking

Notes: The red component for color key checking is also used to hold the alpha value during alpha test.

RDVideoOverlayXEndHigh

Name	Type	Offset	Format
RDVideoOverlayXEndHigh	RAMDAC Control	0x026	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	XEndHigh	✓	✓	X	High order bits of right hand edge of video overlay.
4..7	Reserved	✓	✗	0	

Notes:

RDVideoOverlayXEndLow

Name	Type	Offset	Format
RDVideoOverlayXEndLow	RAMDAC Control	0x025	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	XEndLow	✓	✓	X	Low order bits of right hand edge of video overlay.

Notes:

RDVideoOverlayXStart High

Name	Type	Offset	Format
RDVideoOverlayXStart High	RAMDAC Control	0x022	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	XStartHigh	✓	✓	X	High order bits of left hand edge of video overlay.
4..7	Reserved	✓	✗	0	

Notes:

RDVideoOverlayXStartLow

Name	Type	Offset	Format
RDVideoOverlayXStartLow	RAMDAC Control	0x021	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	XStartLow	✓	✓	X	Low order bits of left hand edge of video overlay.

Notes:

RDVideoOverlayYEndHigh

Name	Type	Offset	Format
RDVideoOverlayYEndHigh	RAMDAC Control	0x028	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	YEndHigh	✓	✓	X	High order bits of last line of video overlay.
4..7	Reserved	✓	✗	0	

Notes:

RDVideoOverlayYEndLow

Name	Type	Offset	Format
RDVideoOverlayYEndLow	RAMDAC Control	0x027	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	YEndLow	✓	✓	X	Low order bits of last line of video overlay.

Notes:

RDVideoOverlayYStartHigh

Name	Type	Offset	Format
RDVideoOverlayYStartHigh	RAMDAC Control	0x024	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..3	YStartHigh	✓	✓	X	High order bits of first line of video overlay.
4..7	Reserved	✓	✗	0	

Notes:

RDVideoOverlayYStartLow

Name	Type	Offset	Format
RDVideoOverlayYStartLow	RAMDAC Control	0x023	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	YStartLow	✓	✓	X	Low order bits of first line of video overlay.

Notes: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register

4.11 Region 0 VSCtl (0x5800-0x5FFF)

VSConfiguration

Name	Type	Offset	Format
VSConfiguration	Video stream Control *Control register*	0x5800	Bitfield

Bits	Name	Read	Write	Reset	Description
0..2	Unit mode	✓	✓	0	0 = ROM Access 1 = MPEG data to decoder via GP bus, decoded video into input port. 2 = Wide output 16 bit. 3 = Simultaneous input and output, program decoder and encoder through I2C. 4 = Wide input 16 bit. 5 = VSA/VSB reset removed, use to probe for external chips. 6 = Drive flat panels 7 = Default to mode 0.
3	GPModeA	✓	✓	0	0 = Operate GP bus in Mode B 1 = Operate GP bus in Mode A
4	VActiveVideoA	✓	✓	1	0 = Ignore VActive for Video data 1 = Gate Video data with VActive
5	VActiveVideoB	✓	✓	1	0 = Ignore VActive for Video data 1 = Gate Video data with VActive
6	GPStopPolarity	✓	✓	0	0 = Active low at pin 1 = Active high at pin
7..8	Reserved	✓	✗	0x7	
9	HRefPolarityA	✓	✓	0	0 = Active low 1 = Active high
10	VRefPolarityA	✓	✓	0	0 = Active low 1 = Active high
11	VActivePolarity A	✓	✓	0	0 = Active low 1 = Active high
12	UseFieldA	✓	✓	0	0 = Disabled 1 = Enabled
13	FieldPolarityA	✓	✓	0	0 = Active low 1 = Active high
14	FieldEdgeA	✓	✓	0	0 = Inactive edge 1 = Active edge
15	VActiveVBIA	✓	✓	0	0 = Ignore VActive for VBI data 1 = Gate VBI data with VActive
16	InterlaceA	✓	✓	0	0 = Video is not interlaced 1 = Video is interlaced
17	ReverseDataA	✓	✓	0	0 = Disabled 1 = Enabled

18	HRefPolarityB	✓	✓	0	0 = Active low	1 = Active high
19	VRefPolarityB	✓	✓	0	0 = Active low	1 = Active high
20	VActivePolarityB	✓	✓	0	0 = Active low	1 = Active high
21	UseFieldB	✓	✓	0	0 = Disabled	1 = Enabled
22	FieldPolarityB	✓	✓	0	0 = Active low	1 = Active high
23	FieldEdgeB	✓	✓	0	0 = Inactive edge	1 = Active edge
24	VActiveVBIB	✓	✓	0	0 = Ignore VActive for VBI data 1 = Gate VBI data with VActive	
25	InterlaceB	✓	✓	0	0 = Video is not interlaced 1 = Video is interlaced	
26	ColorSpaceB	✓	✓	0	0 = YUV	1 = RGB
27	ReverseDataB	✓	✓	0	0 = Disabled	1 = Enabled
28	DoubleEdgeB	✓	✓	0	0 = Disabled	1 = Enabled
29	CCIR656A	✓	✓	0	0 = Disabled	1 = Enabled
30	InvertDoubleEdgeB	✓	✓	0	0 = Disabled	1 = Enabled
31	Reserved	✓	✗	0		

VSDMACommandBase

Name	Type	Offset	Format
VSDMACommandBase	Video stream Control *Control register*	0x5AC8	Integer

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	0	
4..31	Address	✓	✓	0	

Notes:

VSDMACommandCount

Name	Type	Offset	Format
VSDMACommandCount	Video stream Control *Control register*	0x5AD0	Integer

Bits	Name	Read	Write	Reset	Description
0..31	Count	✓	✓	0	

Notes:

VSDMAMode

Name	Type	Offset	Format
VSDMAMode	Video stream Control *Control register*	0x5AC0	Bitfield

Bits	Name	Read	Write	Reset	Description	
0..21	Reserved	✓	✗	0		
22	Active	✓	✓	0	0 = DMA complete	1 = DMA running
23	MemType	✓	✓	0	0 = PCI	1 = AGP
24..25	Burst	✓	✓	0	Log2 of burst length	
26	Reserved	✓	✗	0		
27	Align	✓	✓	0	0 = Disable	1 = Enable
28..31	Reserved	✓	✗	0		

Notes:

VSSerialBusControl

Name	Type	Offset	Format
VSSerialBusControl	Video stream Control	0x5810	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	DataIn	✓	✗	X	0 = Data line is low 1 = Data line is high
1	ClkIn	✓	✗	X	0 = Clock line is low 1 = Clock line is high
2	DataOut	✓	✓	1	0 = Drive data line low 1 = Tri-state data line
3	ClkOut	✓	✓	1	0 = Drive Clock line low 1 = Tri-state clock line
4	LatchedData	✓	✗	0	0 = Data latched at 0 1 = Data latched at 1
5	DataValid	✓	✓	0	0 = DataIn not valid 1 = DataIn valid
6	Start	✓	✓	0	0 = Has not passed through start state 1 = Has passed through start state
7	Stop	✓	✓	0	0 = Has not passed through stop state 1 = Has passed through stop state
8	Wait	✓	✓	0	0 = Do not insert wait states 1 = Insert wait states
9..31	Reserved	✓	✗	0	

Notes: Some bits in this register are set during operation and cleared by writing to the register with those bits set. The bits are DataValid, Start and Stop.

VSStatus

Name	Type	Offset	Format
VSStatus	Video stream Control *Control register*	0x5808	Bitfield

Bits	Name	Read	Write	Reset	Description
0	GPBusTimeOut	✓	✓	0	cleared by writing 1
1..7	Reserved	✓	✗	0	
8	FifoOverflowA	✓	✓	0	cleared by writing 1
9	FieldOne0A	✓	✗	0	
10	FieldOne1A	✓	✗	0	
11	FieldOne2A	✓	✗	0	
12	InvalidInterlaceA	✓	✗	0	
13	BufferFieldA0	✓	✗	0	
14	BufferFieldA1	✓	✗	0	
15	BufferFieldA2	✓	✗	0	
16	FifoUnderflowB	✓	✓	0	cleared by writing 1
17	FieldOne0B	✓	✗	0	
18	FieldOne1B	✓	✗	0	
19	FieldOne2B	✓	✗	0	
20	InvalidInterlaceB	✓	✗	0	
21	BufferFieldB0	✓	✗	0	
22	BufferFieldB1	✓	✗	0	
23	BufferFieldB2	✓	✗	0	
24..31	Reserved	✓	✗	0	

Notes:

4.12 Region 0 VGA Control (0x6000-0x6FFF)

The VGA registers generally follow industry VGA conventions. The registers described below are chip-specific variants accessible both via VGA I/O and addressable memory (described here), togather with the index registers which support them (*GraphicsIndexReg* and *SequencerIndexReg*.). To read or write an indexed register first write the index value to the indexing register, then read/write the memory-mapped address (or VGA I/O Port).

4.12.1 Graphics Index Register

GraphicsIndexReg

Name	Type	Offset	Format
GraphicsIndexReg	VGA *Control register*	0x63CE	Bitfield

Bits	Name	Read	Write	Reset	Description
3:0	Index	✔	✔	X	This index points to one of the Graphics registers which will get read or written on the next I/O access to the GraphicsPort (0x3cf). The registers and their corresponding indices are: 0x0 SetResetReg 0x1 SetResetEnableReg 0x2 ColorCompareReg 0x3 DataRotateReg 0x4 ReadMapSelectReg 0x5 GraphicsModeReg 0x6 GraphicsMiscReg 0x7 ColorDontCareReg 0x8 BitMaskReg 0x9 Mode640Reg 0xa None : : 0xf None
7:4	Reserved	✔	✕	0	Reserved

Notes: Writes to a register denoted 'None' have no effect as the write is simply discarded. Reading from a register denoted 'None' just returns zero.

Mode640Reg

Name	Type	Offset	Format
Mode640Reg	VGA *Control register*	0x63CF	Bitfield

Bits	Name	Read	Write	Reset	Description
2:0	BankA[2:0]	✔	✔	00	This field provides the additonal address bits needed when the horizontal screen resolution is 640 pixels and a host address is being made to the 64K region starting at address 0xa0000.
5:3	BankB[2:0]	✔	✔	00	This field provides the additonal address bits needed when the horizontal screen resolution is 640 pixels and a host address is beign made to the 64K region starting at address 0xb0000.
6	StartAddress16	✔	✔	00	The most significant bit of the *StartAddress* when mode 640 is enabled.
7	Enable	✔	✔	00	0 No action. 1 The VGA core operates in 640 resolution mode.

Notes: This register supports the 640 horizontal resolution modes used in SVGA. The BankA and BankB parts of this register are now obsolete. Programmers should use the sequencer registers instead. This register may be removed from future hardware

4.12.2 Sequencer Registers

SequencerIndexReg

Name	Type	Offset	Format
SequencerIndexReg	VGA *Control Register*	0x63C4	Bitfield

Bits	Name	Read	Write	Reset	Description
5:0	Index	✔	✔	X	This index points to one of the sequencer registers which will get read or written on the next I/O access to the SequencerPort (0x3c5). The registers and their corresponding indices are: 0x00 ResetReg 0x01 ClockModeReg 0x02 MapMaskReg 0x03 CharacterMapSelectReg 0x04 MemoryModeReg 0x05 VGAControlReg 0x06 LockExtended1Reg 0x07 LockExtended2Reg 0x08 BankALowReg 0x09 BankAHighReg 0x0a BankBLowReg 0x0b BankBHighReg 0x0c PCIControlReg 0x0d HLockShiftReg 0x0e VLockShiftReg 0x0f GenLockControlReg 0x10 .. 0x1f ScratchRegs 0x20 .. 0x23 IndirectBaseRegs 0x27 .. 0x3f None
7:6	Reserved	✔	✘	0	Reserved

Notes:
- This register indexes data for the memory mapped *VGAControlReg* register and others shown below. To write to VGAControlReg first write a 0x05 to this regiater, then write data to VGAControlReg
- Writes to a register denoted 'None' have no effect as the write is simply discarded. Reading from a register denoted 'None' just returns zero.

4.12.2.1 Sequenced Registers

BankAHighReg

Name	Type	Offset	Format
BankAHighReg	VGA *Control register*	0x635C index 0x09	Bitfield

Bits	Name	Read	Write	Reset	Description
0,1	BankA9_8	✔	✔		This field holds the 2 high order bits of the 10-bit BankA base address. The 8 low order bits can be found in the BankALowReg. The BankA base address is used for bank switching the 0xa0000 region through the bypass (if enabled). The BankA bits provide the HBankA signals to the PCI interface.
2..7	Reserved	✔	✕	0	

Notes: To read/write this register, first write 0x0F to *SequencerIndexReg*. Not to be confused with Mode640Reg.BankA, which will become obsolete

BankALowReg

Name	Type	Offset	Format
BankALowReg	VGA *Control register*	0x635C index 0x08	Bitfield

Bits	Name	Read	Write	Reset	Description
0…7	BankA7_0	✔	✔		This field holds the 8 low order bits of the 10-bit BankA base address. The 2 high order bits can be found in the BankAHighReg. The BankA base address is used for bank switching the 0xa0000 region through the bypass (if enabled). The BankA bits provide the HBankA signals to the PCI interface.

Notes: To read/write this register, first write 0x08 to *SequencerIndexReg*. Not to be confused with Mode640Reg.BankA, which will become obsolete.

BankBHighReg

Name	Type	Offset	Format
BankBHighReg	VGA	0x635C index 0x0B	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0,1	BankB9_8	✔	✔		This field holds the 2 high order bits of the 10-bit BankB base address. The 8 low order bits can be found in the BankBLowReg. The BankB base address is used for bank switching the 0xb0000 region through the bypass (if enabled). The BankB bits provide the HBankB signals to the PCI interface.
2…7	Reserved	✔	✘	0	

Notes: To read/write this register, first write 0x0B to *SequencerIndexReg*

BankBLowReg

Name	Type	Offset	Format
VGAControlReg	VGA	0x635C index 0x0A	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0…7	BankB7_0	✔	✔		This field holds the 8 low order bits of the 10-bit BankB base address. The 2 high order bits can be found in the BankBHighReg. The BankB base address is used for bank switching the 0xb0000 region through the bypass (if enabled). The BankB bits provide the HBankB signals to the PCI interface.

Notes: Not to be confused with **Mode640Reg.BankB**, which will become obsolete. To read/write this register, first write 0x0A to *SequencerIndexReg*

VGAControlReg

Name	Type	Offset	Format
VGAControlReg	VGA	0x635C index 0x0F	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Enable	✔	✔		If set, syncs the VTG to to an external video source.
1…7	Reserved	✓	✕	0	

Notes: Allows the VTG to be synchronized to an external video source. This causes the horizontal & vertical sync starts & blank ends to be delayed. Sync starts are delayed until the arrival of the ExtHSync & ExtVSync signals. Blank ends are delayed by the numbers specified in the **HLockShiftReg** & **VLockShiftReg** registers. If **VideoControl** *SyncMode*=1 then the sync is tied to the **VideoMergeHSyncIn** pin (B15)

HLockShiftReg

Name	Type	Offset	Format
HLockShiftReg	VGA	0x635C index 0x0D	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0…7		✔	✔		If genlocking is enabled, this field specifies the number of characters by which the horizontal blank end is delayed.

Notes:

IndirectBaseReg[0x0…0x3]

Name	Type	Offset	Format
IndirectBaseReg[0x0…0x3	VGA	0x635C index 0x20 – 0x23	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0…7		✔	✕	x	These 4 registers follow the state of the HIndirectBase signals from the PCI interface. IndirectBaseReg[0] returns bits 7..0, IndirectBaseReg[1] returns bits 15..8, IndirectBaseReg[2] returns bits 23..16, and IndirectBaseReg[3] returns bits 31..24.

Notes: To read from this register, first write the index value (0x20 to 0x23) to *SequencerIndexReg*, then read the required index entries.

LockExtended1Reg

Name	Type	Offset	Format
LockExtended1Reg	VGA	0x63C5 index 0x06	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0…7	Lock	✕	✔		These 2 registers act as a lock for the extended registers. On reset extended registers are locked – they cannot be written and read back as 0, and the sequencer index behaves as a 3-bit index. Writing the value 0x3d to *LockExtended1Reg* followed by 0xdb to *LockExtended2Reg* unlocks the extended registers. Writing any other values locks them.
8…31	Reserved	✓	✕	0	

Notes: To read/write this register, first write 0x06 to *SequencerIndexReg*.

LockExtended2Reg

Name	Type	Offset	Format
LockExtended2Reg	VGA	0x63C5 index 0x07	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0…7	Lock	✕	✔		Acts as a lock for the extended registers. On reset extended registers are locked - they cannot be written and read back as 0, and the sequencer index behaves as a 3-bit index. Writing the value 0x3d to LockExtended1Reg followed by 0xdb to LockExtended2Reg unlocks the extended registers. Writing any other values locks them.

Notes: To read/write this register, first write 0x07 to *SequencerIndexReg*.

PCIControlReg

Name	Type	Offset	Format
PCIControlReg	VGA	0x63C5 index 0x0C	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	BankEnable	✔	✔		If set, enables bank switching of the 0xa0000/0xb0000 regions through the bypass, using the 10-bit BankA/BankB base addresses. This bit provides the HBankEnable signal to the PCI interface.
1	IndirectEnable	✔	✔		If set, enables access to chip registers via I/O ports 0x3b0/0x3b1/0x3d0/0x3d1. This bit provides the HIndirectEnable signal to the PCI interface.
2…7	Reserved	✔	✕	0	Reserved.

Notes: To read/write this register, first write 0x0C to *SequencerIndexReg*

ScratchReg[0x0…0xf]

Name	Type	Offset	Format
ScratchReg[0x0…0xF]	VGA	0x635C index 0x10 to 0x1F	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0…7		✔	✔		These registers are available for use as an information store and do not affect the VGA operation.

Notes: To read/write this register first write the index value (0x10 to 0xF) to *SequencerIndexReg*, then read the required index entries.

VGAControlReg

Name	Type	Offset	Format
VGAControlReg	VGA	0x63C5 index 0x05	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	EnableHost MemoryAccess	✔	✔		Controls access to the display memory by the host. 0 No access to the display memory is made in response to host VGA memory accesses. Writes are ignored and reads always return zero. All the host bus cycles are completed as normal. 1 Normal access to the display memory occurs. This bit is further qualified by the VGAEnable signal which acts as a global disable.
1	EnableHost DacAccess	✔	✔		Controls access to the RAMDAC by the host. 0 No access to the RAMDAC is made in response to host Dac accesses. Writes are ignored and reads always return zero. All the host bus cycles are completed as normal. 1 Normal access to the RAMDAC occurs. This bit is further qualified by the VGAEnable signal which acts as a global disable.

2	Enable Interrupts	✔	✔		0 Prevents any interrupts from being generated by the VGA core. 1 Enables interrupt generation from the VGA core providing the VerticalSyncEndReg.DisableVerticalInterrupt field is set to zero. This bit is further qualified by the VGAEnable signal which acts as a global disable. This additional enable bit is provided so the VGA core can be disabled from one place.
3	EnableVGA Display	✔	✔		Controls access to the display memory by the Memory Reader for the purpose of keeping the display refreshed. It also tells (on the VGAVidSelect signal) the video select logic external to the VGA core that the display should be driven from the VGA core. 0 No accesses to display memory are to be made and the video source should not be the VGA core. The Memory Reader, Attribute Controller and Video Timing Generator are held in their reset state. 1 Accesses to the display memory are made and the video to be displayed comes from the VGA core. This bit is further qualified by the VGAEnable signal which acts as a global disable.
4	DacAddr2	✔	✔		This bit extends the RAMDAC address range.
5	DacAddr3	✔	✔		This bit extends the RAMDAC address range.
6	EnableVTG	✔	✔	x	0 Stops the VTG running and producing sync pulses. 1 Enables the VTG to run and produce sync pulses. This bit only has an effect when the VGA display has been disabled by EnableVGADisplay. When the display has been disabled by VGAEnable this bit is ignored. When the VGA dispaly is active then this bit is ignored.
7	InvertVBlank	✔	✔	0	0 No Invert VBlank. 1 Invert VBlank

Notes:
- On reset EnableHostMemoryAccess, EnableHostDacAccess and EnableVGADisplay are enabled, EnableInterrupts is disabled and DacAddr2 and DacAddr3 bits are set to 0, InvertVBlank is set to 0.
- This is a non standard VGA register.
- To read/write this register, first write 0x05 to *SequencerIndexReg*

VLockShiftReg

Name	Type	Offset	Format
VLockShiftReg	VGA	0x635C index 0x0E	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0…7		✔	✔	0	If genlocking is enabled, this field specifies the number of scanlines by which the vertical blank end is delayed.

Notes:

4.13 Region 0 Texture Data FIFO (0x7000 - 0x7FFF)

4.14 Regions 1 and 2 Memory Apertures

Two memory apertures are provided, each being a PCI region with a fixed size of 128 MBytes.

The two memory apertures can also be programmed to allow reading and writing of the Expansion ROM instead of the memory. This ensures that the “ROM” is visible beyond system boot time, making it possible to program a FLASH or EEPROM device in the field.

4.15 Expansion ROM

The base address register for the Expansion ROM is **CFGBaseAddrROM**, in PCI Configuration Space. The Expansion ROM is mapped into a 64K byte region, so the bypass address is formed from bits 15 down to 2 of the incoming bus address, to give a 14-bit address aligned to a 32-bit boundary. Any higher bypass address bits are set to zero. The address is accompanied by four byte enables.

The two memory apertures can also be programmed to allow reading and writing of the Expansion ROM instead of the memory. In this situation, the bypass address for the ROM is formed from bits 22 down to 2 of the incoming bus address, to give a 21-bit address aligned to a 32-bit boundary. The address is accompanied by four bytes enable, and 32 bits of write data if required. This allows the ROM to be up to 8 MByte in size, but only the bottom 64K is visible as a PCI Expansion ROM.

No byte-swapping or data formatting is performed for ROM accesses.

4.16 VGA Registers

4.16.1 Fixed address decoding

The bus interface can be configured to respond to the standard VGA-compatible Memory and I/O Space addresses using the *VGAFixed* and *VGAEnable* bits in the **ChipConfig** register (or the corresponding configuration pins - see *Reset* in volume IV). The interface will then respond to Memory addresses A0000h through BFFFFh, and I/O addresses 3B0h to 3BBh and 3C0h to 3DFh. These are not affected by the base address registers in PCI configuration space.

The exact memory and I/O addresses which the bus interface should respond to within these ranges is a function of the configuration and mode of the VGA Core Unit. Enable signals from the VGA Unit are used to select which of the possible address sub-ranges are currently active. For example, in monochrome and colour modes a different set of I/O ports must be enabled.

If the *VGANoAlias* config bit is unset then all aliases of these I/O address will also be decoded (only the bottom 10 bits of an I/O address is decoded). If the *VGANoAlias* is set then all 32 bits on the address bus are decoded (no aliasing is supported). The particular addresses which will be responded to depend upon enable signals supplied by the VGA Unit.

4.16.2 Relocatable addressing

When VGA fixed address decoding is disabled it is still possible to access the VGA controller and memory though the relocatable Control Registers in PCI Region Zero, and the VGA memory through either of the two relocatable memory apertures.

Accesses using these relocatable addresses are not affected by enables from the VGA Unit, but will always be forwarded to the VGA Unit (provided that the VGA Unit itself is enabled).

Select signals are provided as part of the PCI/VGA interface to indicate whether a memory access or a control access has been requested. See the Top Level specification for details.

The two additional byte registers provided to support VGA Indirect Accesses are:

VgaIndirectIndex - I/O address 0xB0 (in mono mode) 0xD0 (in color mode)

VgaIndirectReg - I/O address 0xB1 (in mono mode) 0xD1 (in color mode)

These are used to access the I/O registers in Region 3 (namely **IndirectByteEn**, **IndirectData**, **IndirectAddr**, and **IndirectAccess**) in the normal VGA manner. The **VgaIndirectIndex** register is set with the byte offset within Region 3 of the byte to read or write, and then a byte access is made to the **VgaIndirectReg** to register to actually read or write the data.

As an optimisation, it is permitted to write all but the IndirectAccess register using 16-bit I/O writes to 0xB0 or 0xD0. Both the VgaIndirectIndex and VgaIndirectReg are updated together, with the byte data being written using the new value of the index register.

The **IndirectAccess** register must always be selected by a single byte write to **VgaIndirectIndex**, and then a separate byte read or write to **VgaIndirectReg** to trigger the internal R5 indirect access.

These registers (**VgaIndirectIndex** and **VgaIndirectReg**) must be enabled before they can be accessed. This is achieved by setting a bit in one of the VGA extended registers. Index 0xC of I/O register 0x3C5, bit 1 is the **IndirectEnable**. If this bit is not set, (the default), R5 does not respond to accesses to either **VgaIndirectIndex** or **VgaIndirectReg**

4.16.3 Region Zero Addresses

The VGA Control Registers are also mapped into a 4K Byte space within PCI Region Zero. The address for the VGA Unit is formed from bits 9 down to 2 of the incoming bus address, as for Fixed I/O Addresses above. (Bits 11 and 10 of the incoming bus address are ignored, to ensure that the same width of address is generated for both Fixed and Region Zero accesses.)

4.16.4 Memory Aperture Accesses

When the *VGAAccess* bit in either of the **ApertureOne** or **ApertureTwo** registers is set, then all accesses to the relevant aperture will be forwarded to the VGA Unit rather than directly to the memory controller.

INDEX