GLINT R4®

Reference Guide Volume II-

Hardware Registers

DRAFT ONLY

PROPRIETARY AND CONFIDENTIAL INFORMATION

3D*labs*®

GLINT R4®

Reference Guide Volume II-

Hardware Registers

PROPRIETARY AND CONFIDENTIAL INFORMATION

Issue 3

Proprietary Notice

The material in this document is the intellectual property of **3D***labs*. It is provided solely for information. You may not reproduce this document in whole or in part by any means. While every care has been taken in the preparation of this document, **3D***labs* accepts no liability for any consequences of its use. Our products are under continual improvement and we reserve the right to change their specification without notice. **3D***labs* may not produce printed versions of each issue of this document. The latest version will be available from the **3D***labs* web site.

3D*labs* products and technology are protected by a number of worldwide patents. Unlicensed use of any information contained herein may infringe one or more of these patents and may violate the appropriate patent laws and conventions.

3D*labs* is the worldwide trading name of **3D***labs* Inc. Ltd.

3D*labs*, GLINT and PERMEDIA are registered trademarks of **3D***labs* Inc. Ltd.

Microsoft, Windows and Direct3D are either registered trademarks or trademarks of Microsoft Corp. in the United States and/or other countries. OpenGL is a registered trademark of Silicon Graphics, Inc. All other trademarks are acknowledged and recognized.

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* 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: (408) 530-4700
Fax: (408) 530-4701

Change History

Document	Issue	Date	Change
160.5.2	1	06 October 99	Creation
160.5.2	2	05 December 99	Consistent use of GLINT R4 nomenclature; changes to ChipConfig; include Test register set; add Profiling Registers, other corrections.
160.5.2	3	26 January 2000	Update address extension field in LocalMemControl; add LocalMemCapsLb and other *Lb Mem registers, RDWClk registers, RDStripe control, RDPanelControl; register bitfield updates, palette snoop in CFGCommand, CFGClassCode breakout, video timing register reset values, other updates.

Table of Contents

4
Hardware Registers

This chapter lists GLINT R4 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	Indicates bits that may be used in future members of the Permedia family. To ensure upwards compatibility, any software should not assume a value for these bits when read, and should always write them as zeros.
Not Used/ Unused	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.

For enumeration fields that do not specify the full range of possible values, only the specified values should be used. 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 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 R4. 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.

CFGAGPCommand

Name	Type	Offset	Format
CFGAGPCommand	Config	0x48	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description	
0..2	DataRate	✓	✓	0x00000000	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	✓	✗	00.0000.0000.0000b		

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

CFGAGPRev

Name	Type	Offset	Format
CFGAGPRev	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.

CFGAGPStatus

Name	Type	Offset	Format
CFGAGPStatus	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	
5	4G	✓	✓	0b	
6...8	Reserved	✕	✕	000b	
9	SBA	✓	✕	see Desc.	Configured by AGP Capable Side Band Addressing 0 when AGP Capable = 0 1 when AGP Capable = 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 0xoo 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.

CFGBaseAddr0

Name	Type	Offset	Format
CFGBaseAddr0	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 Kbytes.
17..31	Base Offset	✓	✓	0	Loaded at boot time to set offset of the control register space (region 0)

Notes: Base Address 0 Register contains the GLINT R4 control space offset. The control registers are in memory space. They are not prefetchable and can be located anywhere in 32 bit address space.

CFGBaseAddr1

Name	Type	Offset	Format
CFGBaseAddr1	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	0 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..26	Size Indication	✓	✕	0	0 = Region size of 64Mbytes.
27..31	Base Offset	✓	✓	0	Loaded at boot time to set offset of the memory space for aperture one.

Notes: The Base Address 1 Register contains the GLINT R4 aperture one memory offset. It is prefetchable and can be located anywhere in 32 bit address space

CFGBaseAddr2

Name	Type	Offset	Format
CFGBaseAddr2	Configuration	0x18	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Memory Space Indicator	✓	✕	0x000 0.0000	0 = Region is in PCI memory space.
1..2	Address Type	✓	✕	0	0 = 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..26	Size Indication	✓	✕	0	0 = Region size of 128Mbytes.
27..31	Base Offset	✓	✓	0	Loaded at boot time to set offset of the memory space for aperture two.

Notes:
- The Base Address 2 Register contains the GLINT R4 aperture 2 memory offset. It is prefetchable and can be located anywhere in 32 bit address space
- The Base Address 3 Register contains the base address of the GLINT R4 Indirect IO aperture, and defines the size and type of this region.

CFGBaseAddr3

Name	Type	Offset	Format
CFGBaseAddr3	Configuration	0x1C	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0	Memory Space Indicator	✓	✕	0x000 0.0000	1 = Region is in PCI I/O space.
1	Reserved	✓	✕	0	
2,3	Size Indication	✓	✕	0	0 = Region size of 16 Bytes.
4..31	Base Address	✓	✓	0	Loaded at boot time to set base address of PCI region 3

Notes:

CFGBIST

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 GLINT R4 over the PCI interface

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

CFGCacheLine

Name	Type	Offset	Format
CFGCacheLine	Configuration	0x0C	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0...7	Cache Line Size	✓	✕	0x00	00= Cache line size ***unsupported***

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. GLINT R4 does not use this command.

CFGCapID

Name	Type	Offset	Format
CFGCapID	Configuration	0x040	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..7	Capability ID	✓	✕	see desc.	Configured by AGP Capable 0 when AGP Capable = 0 2 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 commences 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	0 = Not implemented

Notes: This register is optional and not supported in R4.

CFGClassCode[BaseClass]

Name	Type	Offset	Format
CFGClassCode[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 R4. 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

CFGClassCode[SubClass]

Name	Type	Offset	Format
CFGClassCode[SubClass]	Configuration	0x0A	Bitfield
	Control register		

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

Notes: The **ClassCode** register is read-only, and is used to identify the generic function of the R4. The register is best viewed as three byte-sized sub-registers. 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
CFGClassCode[InterfaceClassCode]	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 Class Code register is read-only, and is used to identify the generic function of the R4. The register is best viewed as three byte-sized sub-registers. 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.

If the *BaseClassZero* bit in the ChipConfig register is zero, then the Base Class will be reported as 03h, since the R4 is a PCI display controller. If this bit is one then the Base Class will be reported as 00h (which will allow Windows 95 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 I/O Space Accesses	1 = Enable I/O Space Accesses
					If fixed SVGA addressing is disabled, and indirect I/O region is disabled, this bit will be 0	
1	Memory Space Enable	✓	✓	0	0 = Disable memory Space Accesses	1 = Enable memory Space Accesses
2	Bus Master Enable	✓	✓	0	0 = Disable master access	1 = Enable master access
3	Special Cycle Enable	✓	✕	0	0 = GLINT R4 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 all other SVGA accesses	1 = Enable SVGA Palette snooping
6	Parity Error Response enable	✓	✕	0	0 = GLINT R4 does not support parity error reporting	
7	Address/Data stepping enable	✓	✕	0	0 = GLINT R4 does not perform stepping	
8	SERR driver enable	✓	✕	0	0 = GLINT R4 does not support parity error reporting	
9	Master Fast Back-to-Back Enable	✓	✕	0	0 = GLINT R4 master does not do fast back-to-back accesses 1 = Enable fast back-to-back accesses	
10..15	Reserved	✓	✕	0		
16..31					See CFGStatus	

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 R4 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
CFGDeviceID	Configuration	0x02	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..15					See CFGVendorID
16..31	DeviceID	✓	✕	0x000D or: 0x0011	Device identification number: 0x000D = 3Dlabs R4 device identification number If alternateDeviceID=1 then reset = 0x0011

CFGHeaderType

Name	Type	Offset	Format
CFGHeaderType	Configuration	0x0E	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..15					See CFGLatTimer and CFGCacheLine
16..23	Header Type.	✓	✕	0	PCI Definition: 0 = Single Function Device
24..31					See CFGBist

CFGIndirectAddress

Name	Type	Offset	Format
CFGIndirectAddress	Configuration	0x0F8	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..25	Offset	✓	✓	0	Offset within the region.
26..27	Reserved	✓	✗	0	
29..31	Base Address Select	✓	✓	0	0 = Base Address 0 1 = Base Address 1 2 = Base Address 2 3-6 = Reserved 7 = ROM Region

Notes:
1. 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.
2. 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.

CFGIndirectData

Name	Type	Offset	Format
CFGIndirectData	Configuration	0x0F4	Integer

Control register

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 3 and the ROM region directly 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.

CFGIndirectTrigger

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 GLINT R4 device itself.
8..31					See CFGMinGrant, CFGIntPin and CFGMaxLat

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

CFGIntPin

Name	Type	Offset	Format
CFGIntPin	Configuration	0x3D	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..7	Interrupt Pin	✓	✕	0x01	0x01 = GLINT R4 uses Interrupt pin INTAN

Notes: The Interrupt Pin register specifies the interrupt line that GLINT R4 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	✓	✕	0xC0	00.0000b - always set to zero
6	Maximum Latency[6]	✓	✕	0xC0	This is set to 1
7	Maximum Latency[7]	✓	✕	0xC0	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]	✓	✕	0xC0	00.0000b - always set to zero
6	MinimumGrant [6]	✓	✕	0xC0	This is set to 1
7	MinimumGrant [7]	✓	✕	0xC0	This is set to 1

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

CFGNextPtr

Name	Type	Offset	Format
CFGNextPtr	Configuration	0x041	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0...7	Next Ptr	✓	✕	0	00 = 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.

CFGPMC

Name	Type	Offset	Format
CFGPMC	Configuration	0x4E	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Version	✓	✕	0x221	1=complies with Rev 1.0 of the PCI Power Management Interface Spec.
3	PME Clock	✓	✕	0x0	0=PME# not supported in any state
4	Aux Power source	✓	✕	0	0 = PME# is not supported in D3(cold)
5	DSI	✓	✕	1	1 = GLINT R4 requires special initialization following transition to the D0 uninitialized state
6...8	Reserved	✓	✕	0	
9	D1_Support	✓	✕	0x1	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:

CFGPMCapID

Name	Type	Offset	Format
CFGPMCapID	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	✕	✕	x	

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

CFGPMCS

Name	Type	Offset	Format
CFGPMCS	Configuration	0x50	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..1	PowerState	✓	✓	0x00	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 = GLINT R4 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	This 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.	0x00 = no further capabilities in list if AGP Capable = 0 0x40 = point to AGP Capability if AGP Capable = 1
8...31	Reserved				

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
8..31					See CFGClassCode

Notes:

CFGRomAddr

Name	Type	Offset	Format
CFGRomAddr	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	
11..15	Size Indication	✓	✕	0.0000 b	0 = Indicates that Expansion ROM must be mapped into 64Kbytes.
16..31	Base Offset	✓	✓	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	✓	✕	0	
4	Cap_List	✓	✕	0x1	1 = GLINT R4 can accept additional capabilities beyond PCI2.1. These are power management and AGP (if AGP Capable is set in CFGCapID)
5	66MHz Capable	✓	✕	X	0 = GLINT R4 is 33MHz capable only / 1 = GLINT R4 is 66MHz capable
6	UDF Supported	✓	✕	0	0 = GLINT R4 does not support user-definable configurations
7	Fast back-to-back capable	✓	✕	0x1	1 = GLINT R4 can accept fast back-to-back PCI transactions
8	Data Parity Error Detected	✓	✕	0	0 = Parity checking not implemented on GLINT R4
9...10	DEVSEL Timing	✓	✕	0x1	1 = GLINT R4 asserts DEVSEL# at medium speed
11	Signaled Target Abort	✓	✕	0	0 = GLINT R4 never signals Target-Abort
12	Received Target Abort	✓	✓	0	This bit is set by the GLINT R4 bus master whenever its transaction is terminated with Target-Abort
13	Received Master Abort	✓	✓	0	This bit is set by the GLINT R4 bus master whenever its transaction is terminated with Master-Abort
14	Signalled System Error	✓	✕	0	0 = GLINT R4 never asserts a system error
31	Detected Parity Error	✓	✕	0	0 = Parity checking is not implemented by GLINT R4

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 GLINT R4 device resides. It has two possible reset states: the value may be loaded from the ROM byte addresses 0xFFFE and 0xFFFF, or reset to the Device ID and then written to once before it becomes read only. The option is controlled by a configuration register

CFGSubsystemVendorId

Name	Type	Offset	Format
CFGSubsystemVendorId	Configuration	0x02C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..15	SubsystemVendorID	✗	✓ once	see text	

Notes: This register is used to identify the vendor of the add-in board on which the GLINT R4 device resides. It has two possible reset states: The value may be loaded from the ROM byte addresses 0xFFFC and 0xFFFD, or reset to the vendor ID and then written to once before it becomes read-only. The option is controlled by a configuration register

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

4.2 Region 0 Control Status (0x0000-0x02FF)

The R4 Region Zero is a 128KByte region containing the control registers and ports to and from the graphics processor. The control space is mapped in twice within the 128KByte region. In the second 64K the registers are mapped to be byte swapped for big endian hosts.

A number of Control Status Registers are implemented within the PCI Bus Interface, including registers for interrupt and error handling, 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.

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.

AGPControl

Name	Type	Offset	Format
AGPControl	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					

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 SVGA or the ROM instead of the memory controller

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
12....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..4	Reserved	✓	✗	X	
5	RetryDisable	✓	✓	X	0 = Enable PCI Retry using "Disconnect-Without-Data" 1 = Disable PCI Retry using "Disconnect-Without-Data"
6	Reserved	✓	✗	X	
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	IndirectIO Enable	✓	✓	X	0 = Base Address 3 disabled - Indirect IO accesses cannot be performed 1 = IndirectIO accesses enabled
14	WC Enable	✓	✓	X	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	✓	✓	X	0 = Regions 1 and 2 marked as not prefetchable 1 = Regions 1 and 2 marked as prefetchable
16	Alternate Device IDEnable	✓	✗	X	0=Device ID=0x0D 1=Device ID=0x11

17	AutoCal Enable	✓	✕	X	0=Disable 1=Enable
18..27	Reserved	✓	✕	X	(all bits zero)
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 R4 chip has been configured, and to modify various fields of the configuration if required.

ControlDMAAddress

Name	Type	Offset	Format
ControlDMAAddress	Control Status *Control register*	0x28	Integer

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	✓	✓	0x000 0.0000	This field should only be changed when the ControlDMA controller 0 = Standard. 1 = Byte Swapped is idle.
1	ControlDMA using AGP	✓	✓	0	0 = DMA uses PCI Master 1 = DMA uses AGP Master
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	0x30	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..15	Control DMA Count	✓	✓	0	Number of words to be transferred in the DMA operation. The valid range for this register is 0 to 65535. The register behaviour is undefined if it is written to while non-zero and Mastering is enabled. Mastering is enabled if *ControlDMAUseAGP* = 0 and PCI Bus Master Enabled or *ControlDMAUseAGP* = 1 and AGP Master is enabled. See **DMAControlRegister**.
16..31	Reserved	✓	✕	0x0000	

Notes:
1. 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.
2. 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.

ErrorFlags

Name	Type	Offset	Format
ErrorFlags	Control Status	0x0038	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Input FIFO Error Flag	✓	✓	0x000 0.0000	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	✓	✗	0b	
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
5,6	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	VSA Invalid Interlace Error Flag	✓	✓	0	Reserved
12	VSB Invalid Interlace Error Flag	✓	✓	0	Reserved

13	HostIn DMA Error Flag	✓	✓	0	Flag set if HostIN DMA error occurs 0 = No error 1 = Error Outstanding
14..31	Reserved	✓	✕	0	

Notes: The Error Flags register shows which errors are outstanding in GLINT R4 . 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
FIFODiscon	Control Status	0x68	Bitfield
	Control register		

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..31	Reserved	✓	✕	0	

Notes: The FIFODiscon register enables the input and output FIFO disconnect signals, which drive two physical pins on the GLINT R4. 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				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.

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	✓	✗	0x0000.0008	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.

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	VS External Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
13	Bypass DMA Write Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt
14	HostIn Command Interrupt Enable	✓	✓	0	0 = Disable interrupt. 1 = Enable interrupt.
15	VS DMA Interrupt enable	✓	✓	0	0 = Disable interrupt 1 = Enable interrupt
16..31	Reserved	✓	✗	0	Read Only.

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	VS External	✓	✓	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	VS DMA	✓	✓	0	0 = No interrupt 1 = Interrupt Outstanding
16..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.)

LogicalTexturePage

Name	Type	Offset	Format
LogicalTexturePage	Control Status	0x118	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..15	LogicalTexture Page	3	5	X	
16..31	Reserved	3	5	0	

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

OutFIFOWords

Name	Type	Offset	Format
OutFIFOWords	Control Status	0x0020	Integer
	Control register		

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
PCIAbortAddress	Control Status	0x098	Integer
	Control register		

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.

PCIAbortStatus

Name	Type	Offset	Format
PCIAbortStatus	Control Status	0x090	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..6	ReadSource	✓	✕	0	The read source in the DMA Arbiter that caused the Abort.
7	ReadStatus	✓	✕	0	0 = No read abort 1 = Read abort
8..14	WriteSource	✓	✕	0	The Write source in the DMA Arbiter which caused the Abort.
15	WriteStatus	✓	✕	0	0 = No Write abort 1 = Write abort.
16..31	Reserved	✓	✕	0	

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 **PCIAbortStatus** register can be cleared by writing any value to the register.

PCIFeedbackCount

Name	Type	Offset	Format
PCIFeedbackCount	Control Status	0x088	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	PCI Feedback Count	✓	✕	0	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
PCIPLLStatus	Control Status	0x00F0	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..8	PCIPLLSetup	✓	✓		Provides 9 bits of setup for the deskew PLL.
9..11	PCIPLL PostScale	✓	✓	0x1	Divide by 2
12	PCIPLL Enable	✓	✓	0x1	
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
PclkProfCount0	Control Status	0x260	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
PclkProfCount1	Control Status	0x288	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	0x240	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.

PclkProfInvertA1

Name	Type	Offset	Format
PclkProfInvertA1	Control Status	0x270	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	0x248	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	PclkProfInvertB0	✓	✓	0x000 0.0000	

Notes: Bitwise inverts the "B" set of event flags to profiling counter 0.

PclkProfInvertB1

Name	Type	Offset	Format
PclkProfInvertB1	Control Status	0x270	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	0x250	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	1=AND 2=OR
31	Reserved	✓	✗	X	

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

PclkProfMaskA1

Name	Type	Offset	Format
PclkProfMaskA1	Control Status	0x278	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	1=AND 2=OR
31	Reserved	✓	✗	X	

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

PclkProfMaskB0

Name	Type	Offset	Format
PclkProfMaskA0	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	VSDMA Running	✓	✓	X	
15	VSDMACtrl Full	✓	✓	X	
16	VSDMAData Empty	✓	✓	X	
17	VSDMAData Full	✓	✓	X	
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	✓	✗	X	

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

PclkProfMaskB1

Name	Type	Offset	Format
PclkProfMaskB1	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	VSDMA Running	✓	✓	X	
15	VSDMACtrl Full	✓	✓	X	
16	VSDMAData Empty	✓	✓	X	
17	VSDMAData Full	✓	✓	X	
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	✓	✕	X	

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

ResetStatus

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

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.

TexDMAAddress

Name	Type	Offset	Format
TexDMAAddress	Control Status	0x120	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	TexDMA Address	✓	✗	X	

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

TestInputControl

Name	Type	Offset	Format
TestInputControl	Control Status	0x200	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Input Hostin	✓	✓	0	
1	Test Input Delta0	✓	✓	0	
2	Test Input Delta1	✓	✓	0	
3	Test Input TwoD	✓	✓	0	
4	Test Input Rasterizer	✓	✓	0	
5	Test Input Scissor	✓	✓	0	
6	Test Input Router	✓	✓	0	
7	Test Input LBRead	✓	✓	0	
8	Test Input Stencil	✓	✓	0	
9	Test Input LBWrite	✓	✓	0	
10	Test Input ColorDDA	✓	✓	0	

11	Test Input TX Coord	✓	✓	0	
12	Test Input TX Index	✓	✓	0	
13	Test Input TX Read	✓	✓	0	
14	Test Input TX LUT	✓	✓	0	
15	Test Input TX Filter	✓	✓	0	
16	Test Input TX Comp	✓	✓	0	
17	Test Input TX App	✓	✓	0	
18	Test Input Fog	✓	✓	0	
19	Test Input YUV	✓	✓	0	
20	Test Input AlphaTest	✓	✓	0	
21	Test Input FBRead	✓	✓	0	
22	Test Input ABlend	✓	✓	0	
23	Test Input Dither	✓	✓	0	
24	Test Input Logicop	✓	✓	0	
25	Test Input FBWrite	✓	✓	0	
26	Test Input Hostout	✓	✓	0	
27	Test Input Read Monitor	✓	✓	0	
28	Test Input DeltaSwitch	✓	✓	0	
29	Test Input DeltaMux0	✓	✓	0	
30	Test Input DeltaMux1	✓	✓	0	
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.

TestInputRdy

Name	Type	Offset	Format
TestInputRdy	Control Status	0x208	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Input Hostin Ready	✓	✓	0x03ff.ffff	
1	Test Input Delta0 Ready	✓	✓	0x03ff.ffff	
2	Test Input Delta1 Ready	✓	✓	0x03ff.ffff	
3	Test Input TwoD Ready	✓	✓	0x03ff.ffff	
4	Test Input Rasterizer Ready	✓	✓	0x03ff.ffff	
5	Test Input Scissor Ready	✓	✓	0x03ff.ffff	
6	Test Input Router Ready	✓	✓	0x03ff.ffff	
7	Test Input LBRead Ready	✓	✓	0x03ff.ffff	
8	Test Input Stencil Ready	✓	✓	0x03ff.ffff	
9	Test Input LBWrite Ready	✓	✓	0x03ff.ffff	
10	Test Input ColorDDA Ready	✓	✓	0x03ff.ffff	
11	Test Input TX Coord Ready	✓	✓	0x03ff.ffff	
12	Test Input TX Index Ready	✓	✓	0x03ff.ffff	
13	Test Input TX Read Ready	✓	✓	0x03ff.ffff	
14	Test Input TX LUT Ready	✓	✓	0x03ff.ffff	
15	Test Input TX Filter Ready	✓	✓	0x03ff.ffff	
16	Test Input TX Comp Ready	✓	✓	0x03ff.ffff	

17	Test Input TX App Ready	✓	✓	0x03ff.ffff	
18	Test Input Fog Ready	✓	✓	0x03ff.ffff	
19	Test Input YUV Ready	✓	✓	0x03ff.ffff	
20	Test Input AlphaTest Ready	✓	✓	0x03ff.ffff	
21	Test Input FBRead Ready	✓	✓	0x03ff.ffff	
22	Test Input ABlend Ready	✓	✓	0x03ff.ffff	
23	Test Input Dither Ready	✓	✓	0x03ff.ffff	
24	Test Input Logicop Ready	✓	✓	0x03ff.ffff	
25	Test Input FBWrite Ready	✓	✓	0x03ff.ffff	
26	Test Input Hostout Ready	✓	✓	0x03ff.ffff	
27	Test Input Read Monitor Ready	✓	✓	0x03ff.ffff	
28	Test Input DeltaSwitch Ready	✓	✓	0x03ff.ffff	
29	Test Input DeltaMux0 Ready	✓	✓	0x03ff.ffff	
30	Test Input DeltaMux1 Ready	✓	✓	0x03ff.ffff	
31	Readback Disable Ready	✓	✓	0x03ff.ffff	

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.

TestOutputControl

Name	Type	Offset	Format
TestOutputControl	Control Status	0x210	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Output GPInFifo	✓	✓	0	
1	Test Output Delta Switch0	✓	✓	0	
2	Test Output Delta Switch1	✓	✓	0	
3	Test Output Delta Mux	✓	✓	0	
4	Test Output TwoD	✓	✓	0	
5	Test Output Rasterizer	✓	✓	0	
6	Test Output Scissor	✓	✓	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 Fog	✓	✓	0	
20	Test Output YUV	✓	✓	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 HostIn	✓	✓	0	
29	Test Output Delta0	✓	✓	0	
30	Test Output Delta1	✓	✓	0	
31	Test Output Router	✓	✓	0	

Notes: Sets the source location for FIFO test reads..

TestOutputRdy

Name	Type	Offset	Format
TestOutputRdy	Control Status	0x218	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	Test Output GPInFifo Ready	✓	✓	0	
1	Test Output Delta Switch0 Ready	✓	✓	0	
2	Test Output Delta Switch1 Ready	✓	✓	0	
3	Test Output Delta Mux Ready	✓	✓	0	
4	Test Output TwoD Ready	✓	✓	0	
5	Test Output Rasterizer Ready	✓	✓	0	
6	Test Output Scissor 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 Fog Ready	✓	✓	0	
20	Test Output YUV 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 HostIn Ready	✓	✓	0	
29	Test Output Delta0 Ready	✓	✓	0	
30	Test Output Delta1 Ready	✓	✓	0	
31	Test Output Router Ready	✓	✓	0	

Notes: Sets the source location for FIFO test reads..

TestReadData

Name	Type	Offset	Format
TestReadData	Control Status	0x238	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..4	TestReadData	✓	✓	0x000 0.0000	Data

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

TestReadSelect

Name	Type	Offset	Format
TestWriteSelect	Control Status	0x230	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..4	TestReadSelect	✓	✓	0x0	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 (129:160) 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
TestWriteSelect	Control Status	0x220	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0..4	TestWriteSelect	✓	✓	0x0	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 (129:160)

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	✓	✓	0x000 0.0000	Data

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

TexFIFOSpace

Name	Type	Offset	Format
TexFIFOSpace	Control Status	0x128	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.

TextureDownloadControl

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

Bits	Name	Read	Write	Reset	Description
0	Texture Download Enable	✓	✓	0	
1	Texture Download Busy	✓	✗	0	
2	Texture MemType	✓	✓	0	0 = PCI, 1 = AGP Download
3..7	TextureGranularity	✓	✓	0	
8..12	TextureThreshold	✓	✓	0	
13	SlaveTextureDownload	✓	✓	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	0x110	Integer
	Control register		

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

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

VClkRDacCtl

Name	Type	Offset	Format
VClkRDacCtl	Control Status	0x40	Bitfield
	Control register		

Bits	Name	Read	Write	Reset	Description
0	VidCtl(0) pin	✓	✓	0x000 0.0000	
1	VidCtl(1) pin	✓	✓	0	

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

4.3 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.

This unit does not throttle PCI read requests on room in the fifo to memory, but relies on the speed of the memory to rarely stall the bus. Throttling is not possible because the bypass fifo is shared by all sources, so the amount of room in it is unpredictable. There is control over the size of each DMA request, both reads and writes, to prevent very long bursts that hog the bus. There is also control to align DMA transfers to cache line (64 bytes) boundaries. DMAWrite data is loaded into a storage fifo before being sent to the PCI. There is data throttling on this fifo, as any data returned must be removed from the bypass return data fifo to allow direct reads to complete.

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.

The DirectAccess port from the PCI is required to supply the lower 2 bits of the 32 bit aligned address in the Offset field of the fifo. This is used to steer the data correctly for targets that have a 32 bit interface. The same offset is returned to the PCI for all read operations, including reads from memory. Data is always transmitted along the correct byte lanes, with byte enables set appropriately.

ByAperture1Mode
ByAperture2Mode

Name	Type	Offset	Format
ByAperture1Mode	Bypass Control	0x0300	Bitfield
ByAperture2Mode	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
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	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..24	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
25..31	Reserved	✓	✗	0	

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

ByAperture1UStart
ByAperture2UStart

Name	Type	Offset	Format
ByAperture1UStart	Bypass Control	0x0318	Integer
ByAperture2UStart	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	✓	✗	X	

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
ByAperture1VStart	Bypass Control	0x0320	Integer
ByAperture2VStart	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	✓	✗	X	

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
ByAperture1YStart	Bypass Control	0x0310	Integer
ByAperture2YStart	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	✓	✕	X	

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
ByAperture1Stride	Bypass Control	0x0308	Integer
ByAperture2Stride	Bypass Control	0x0330	Integer

Control register

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

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

ByDMAReadCommandBase

Name	Type	Offset	Format
ByDMAReadCommandBase	Bypass Control	0x0378	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..3	Reserved	✓	✗	X	
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
ByDMAReadCommand 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
ByDMAReadMode	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
5..6	PixelSize	✓	✓	0	0 = 8 bits 1 = 16 bits
7..8	EffectiveStride	✓	✓	0	2 = 4096
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	Active	✓	✓	0	Indicates the status of the DMA. 0 = DMA Idle 1 = DMA Running
23	MemType	✓	✓	0	Type of bus protocol to use for DMA.
					0 = PCI 1 = AGP
24..26	Burst	✓	✓	0	Size of burst defined as log2 of burst size.
27	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 system memory and writing it to local memory.

ByDMAReadStride

Name	Type	Offset	Format
ByDMAReadStride	Bypass Control *Control register*	0x0358	Integer

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

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

ByDMAReadUStart

Name	Type	Offset	Format
ByDMAReadUStart	Bypass Control *Control register*	0x0368	Integer

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

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
ByDMAReadVStart	Bypass Control	0x0370	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	✓	✕	X	

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
ByDMAReadYStart	Bypass Control	0x0360	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	✓	✕	X	

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	✓	✕	X	
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 *Control register*	0x0388	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	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	Active	✓	✓	0	Indicates the status of the DMA. 0 = DMA Idle 1 = DMA Running
23	MemType	✓	✓	0	Type of bus protocol to use for DMA. 0 = PCI 1 = AGP
24..26	Burst	✓	✓	0	Size of burst defined as log2 of burst size.
27	Align	✓	✓	0	Enables alignment of transfers to 64 byte boundaries.
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	✓	✗	X	

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	✓	✗	X	

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	✓	✗	X	

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	✓	✗	X	

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.

4.4 Region 0 Memory Control (0x1000-0x1FFF)

LocalMemCaps

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

Bits	Name	Read	Write	Reset	Description
0..3	Column Address	✓	✓	0	Address bits to use for column address.
4..7	RowAddress	✓	✓	0	Address bits to use for row address.
8..11	BankAddress	✓	✓	0	Address bits to use for bank address.
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	HalfWidth	✓	✓	0x1	0 = off 1 = on
31	NoLookAhead	✓	✓	0x1	0 = off 1 = on

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).

LocalMemCapsLb

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

Bits	Name	Read	Write	Reset	Description
0..3	Column Address	✓	✓	0	Address bits to use for column address.
4..7	RowAddress	✓	✓	0	Address bits to use for row address.
8..11	BankAddress	✓	✓	0	Address bits to use for bank address.
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	Reserved	✗	✗	0	
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	HalfWidth	✓	✓	0x1	0 = off 1 = on
31	NoLookAhead	✓	✓	0x1	0 = off 1 = on

Notes:
1. The Lb registers apply to R4 only
2. The Reset state guarantees access to the lower 512 bytes of memory.
3. 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.
4. 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).

LocalMemControl
LocalMemControlLb

Name	Type	Offset	Format
LocalMemControl	Memory Control	0x1028	Bitfield
LocalMemControlLb	Memory Control Command register	0x1050	Bitfield

Bits	Name	Read	Write	Reset	Description
0..2	CASLatency	✓	✓	0x3	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	Redeploys Memory Controller Bank Select signals as address lines to enable use of larger memory devices: 0: no change 1: BankSelect0=>Address12 BankSelect1=>BankSelect0 BankSelect2=>BankSelect1 BankSelect3=>BankSelect2 2: BankSelect0=>Address12 BankSelect1=>Address13 BankSelect2=>BankSelect0 BankSelect3=>BankSelect1 3: BankSelect0=>Address12 BankSelect1=>Address13 BankSelect2=>Address14 BankSelect3=>BankSelect0 4: BankSelect0=>Address12 BankSelect1=>Address13 BankSelect2=>Address14 BankSelect3=>Address15
7...21	Reserved	✓	✗	0	
22..31	Mode	✓	✓	0x030	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. **LocalMemControlLb** is supported on GLINT R4 only.
2. 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.
3. Bits 22 and 31 of LocalMemControl register correspond respectively to bits 0 and 9 of the mode register in the memory device.
4. *AddressExtension*: The procedure to set up for different configurations of 32MB and 64MB devices is described in the *GLINT R4 Reference Guide* volume IV: Memory System.

LocalMemPowerDown
LocalMemPowerDownLb

Name	Type	Offset	Format
LocalMemPowerDown	Memory Control Command register	0x1038	Bitfield
LocalMemPowerDownLb	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: Timeout between reseting memory to low power mode in 32 clock units.

LocalMemProfileMask0

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

Bits	Name	Read	Write	Reset	Description	
0	Always	✓	✓	0	0 = Exclude	1 = Include
1	Idle	✓	✕	0	@@@@	
17..31	Delay	✓	✓	0	Timeout in 32 clock units	

Notes: Lb signals apply to GLINT R4 only.

LocalMemRefresh
LocalMemRefreshLb

Name	Type	Offset	Format
LocalMemRefresh	Memory Control	0x1030	Bitfield
LocalMemRefreshLb	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.

LocalMemTiming
LocalMemTimingLb

Name	Type	Offset	Format
LocalMemTiming	Memory Control	0x1020	Bitfield
LocalMemTimingLb	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	PrechargeToAc ti vate	✓	✓	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 Pr echarge	✓	✓	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 Precharg e	✓	✓	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	Reserved	✓	✗	0		

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.

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	

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..31	Reserved	✓	✗	0		

Notes:

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

No 0x2000 series registers are listed.

4.6 Region 0 Video Control (0x3000-0x3FFF)

DisplayData

Name	Type	Offset	Format
DisplayData	Video Control *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	
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	0x3078	Bitfield
	Control register		

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: Reset = 0000.0000.0000.0000.0001.0000.0001.0000

HbEnd

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

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	0x3018	Integer
	Control register		

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	0x3030	Integer
	Control register		

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

Notes: 000.000.000.000.000.0XXX.XXX.XXX

HsOffset

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

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 of the locked Hsync is HsStart + HsOffset. See *Multi-rasterizer Setup* in Volume I, chapter 5.
000.000.000.000.000.0XXX.XXX.XXX

HsStart

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

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	0x3010	Integer
	Control register		

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
InterruptLine	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: Reset = 000.000.000.000.000.0XXX.XXX.XXX

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
2..3	Reserved	✓	✕	0	
4..6	StripeSize	✓	✓	0	0 = 1 line 1 = 2 lines 2 = 4 lines 3 = 8 lines 4 = 16 lines
7	ByteDouble	✓	✓	0	

Notes: 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..20	ScreenBase	✓	✓	X	Base address of screen in 128 bit units.
21..31	Reserved	✕	✕	0	

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

ScreenBaseRight

Name	Type	Offset	Format
ScreenBaseRight	Video Control *Control register*	0x3080	Integer

Bits	Name	Read	Write	Reset	Description
0..20	ScreenBase Right	✓	✓	X	Base address of right screen in 128 bit units.
21..31	Reserved	✗	✗	0	

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

StripeStride

Name	Type	Offset	Format
StripeStride	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 *Control register*	0x3040	Integer

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 *Control register*	0x3070	Integer

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
VideoControl	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 = SyncOnFrameBlan k 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	LockToVSB	✓	✓	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 threshold	✓	✓	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
VideoOverlayMode	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 = reserved	1 = reserved 3..7 = Reserved
4	FieldPolarity	✓	✓	0	0 = Normal	1 = Invert
5..6	PixelSize	✓	✓	0	0 = 8 bits	1 = 16 bits
					2 = 32 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	LinearColorExt e nsion	✓	✓	0	0 = Off	1 = On
14..15	Filter	✓	✓	0	0 = Off	1 = Full
					2 = Partial (X with zoom)	3 = Reserved
16..17	DeInterlace	✓	✓	0	0 = Off	1 = Bob
					2..3 = Reserved	
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:

			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	0x3050	Integer
	Control register		

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	0x3048	Integer
	Control register		

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	0x3038	Integer
	Control register		

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

4.7 Region 0 RAMDAC

Direct and Indirect RAMDAC registers are listed separately.

4.7.1 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
RDIndexHigh	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
RDIndexLow	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
RDPaletteData	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
RDPaletteReadAddress	RAMDAC Control	0x4018	Integer
	Control register		

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

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.

RDPaletteWriteAddress

Name	Type	Offset	Format
RDPaletteWriteAddress	RAMDAC Control	0x4000	Integer

Control register

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

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

4.7.2 Indirect RAMDAC Registers (0x200-0xFFF)

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:
- This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.
- You can use this register to tell the RAMDAC to sum the R, G and B values for a scan line. Typically, wait for Vblank, enable checksum before or after LUT, wait for RAMDAC to sum first active scanline (after which enable bits are Reset) then read **RDCheckLUT*** or **RDCheckPixel*** registers for the corresponding RGB component values..

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

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

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

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

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

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

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:
1. This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.
2. 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	0	Reserved	8@0	8@8	8@16	8@24

			Internal Color Channels			
ColorFormat	**RGB**	**Name**	**R**	**G**	**B**	**O**
19..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
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	1	Reserved	8@16	8@8	8@0	8@24
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
19..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: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register

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

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

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.

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	✓	✓	X	Bitmap 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
RDCursortXHigh	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:
1. This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.
2. Value at readback is determined by the ReadbackPosition field in the RDCursorControl register.

RDCursorXLow

Name	Type	Offset	Format
RDCursortXLow	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:
1. This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.
2. 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:
1. This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.
2. 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: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.
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	DACPower Ctl	✓	✓	0	0 = Normal operation. 1 = LowPower
3	SyncOnGreen	✓	✓	0	0 = Disabled. 1 = Enabled
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: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.

RDDClk0FeedbackScale

Name	Type	Offset	Format
RDDClk0FeedbackScale	RAMDAC Control	0x202	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0x7	

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

RDDClk0PostScale

Name	Type	Offset	Format
RDDClk0PostScale	RAMDAC Control	0x203	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Scale	✓	✓	0	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				

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.

RDDClkPostScale
RDDClk1PostScale

Name	Type	Offset	Format
RDDClkPostScale	RAMDAC Control	0x206	Integer
RDDClk1PostScale	RAMDAC Control	0x210	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Scale	✓	✓	X	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: This register is accessed indirectly by first loading the index into the *RDIndexLow* and *RDIndexHigh* registers, and then reading or writing the *RDIndexedData* register.

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

RDDClk0PreScale

Name	Type	Offset	Format
RDDClk0PreScale	RAMDAC Control	0x201	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0x4	

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.

RDDClk1FeedbackScale

Name	Type	Offset	Format
RDDClk1FeedbackScale	RAMDAC Control	0x24F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0x4F	

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.

RDDClk1PreScale

Name	Type	Offset	Format
RDDClk1PreScale	RAMDAC Control	0x28	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0x28	

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.

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	✓	✓	X	

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

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	✓	✓	X	

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

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

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 = NotLocked	1 = Locked
2..3	State	✓	✓	0x2	0 = Drive Low 2 = Run	1 = Drive High 3 = Low Power
4..6	Source	✓	✓	0	0 = PClk 2 = PLL	1 = PClk/2 3..7 = Reserved
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.

RDKClkFeedbackScale

Name	Type	Offset	Format
RDKClkFeedbackScale	RAMDAC Control	0x20F	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0x20	

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

RDKClkPreScale

Name	Type	Offset	Format
RDKClkPreScale	RAMDAC Control	0x20E	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Value	✓	✓	0x10	

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

RDKClkPostScale

Name	Type	Offset	Format
RDKClkPostScale	RAMDAC Control	0x206	Integer

Control register

Bits	Name	Read	Write	Reset	Description	
0..2	Scale	✓	✓	X	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: This register is accessed indirectly by first loading the index into the *RDIndexLow* and *RDIndexHigh* registers, and then reading or writing the *RDIndexedData* register.

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 = Reserved 4 = ExternalMclk 6 = KClk PLL	1 = PClk/2 3 = ExternalMClk/2 5 = KClk PLL/2 7 = Reserved
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.
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.

RDMergeControl

Name	Type	Offset	Format
RDMergeControl	RAMDAC Control Command register	0x012	Bitfield

Bits	Name	Read	Write	Reset	Description
0	ClockIn	✓	✓	Xx00.0000	0 = use internal clk 1 = use external clk
1	StrobeOut	✓	✓		0 = drive external strobe low 1 = drive CClk times 2 out
2	SyncIn	✓	✓		0 = use internal sync 1 = use external sync
3`	SyncOut	✓	✓		0 = drive external sync lines inactive 1 = drive syncs out
4	DataIn	✓	✓		0 = ignore external data 1 = use external data
5	DataOut	✓	✓		0 = drive external video bus to 0 1 = drive data to external video bus
6,7	Reserved	✓	✕		

Notes:

RDMergeSkew

Name	Type	Offset	Format
RDMergeSkew	RAMDAC Control Command register	0x013	Bitfield

Bits	Name	Read	Write	Reset	Description
0...5	Compensation	✓	✓	0000.0000	Compensation for skew between external clock and sync
6,7	Reserved	✓	✕		

RDMiscControl

Name	Type	Offset	Format
RDMiscControl	RAMDAC Control Command register	0x000	Bitfield

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 - return palette access state. 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
					0 = Disabled 1 = Enabled
7	StereoDouble Buffer	✓	✓	0	Controls per-pixel double buffering in 5551 color format. 0 = Disabled. 1 = Enabled.

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.

RDOverlayKey

Name	Type	Offset	Format
RDOverlayKey	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: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register

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 64 bits on line.
7..2	Reserved	✓	✗	X	

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.

RDPanelControl

Name	Type	Offset	Format
RDPanelControl	RAMDAC Control	0x014	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0,1	DataOut	✓	✓	0000.0000	0 = off 1 = Panel24 (24 bit port) 2 = Panel48 (48 bit port) 3 = reserved
2	Clock	✓	✓		0 = fast 1 = slow
3	HsyncCtl	✓	✓		0 = ActiveLow 1 = ActiveHigh
4	HsyncOverride	✓	✓		0 = off 1 = on
5	VsyncCtl	✓	✓		0 = ActiveLow 1 = ActiveHigh
6	VSyncOverride	✓	✓		0 = off 1 = on
7	BlankCtl	✓	✓		0 = ActiveLow 1 = ActiveHigh

Notes: *Clock* is used to adjust the RAMDAC frequency to the pixel bus - see *Multi-rasterizer Setup* in Volume I. This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.

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..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

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	✓	✓	0x2	0 = Drive Low 1 = Drive High 2 = Run 3 = Low Power
4..6	Source	✓	✓	0x0	0 = PClk/2 1 = PClk 2 = Reserved 3 = ExternalSClk/2 4 = ExternalSClk 5 = KClk/2 6 = KClk 7 = Reserved
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: This register is accessed indirectly by first loading the index into the RDIndexLow and RDIndexHigh registers, and then reading or writing the RDIndexedData register.

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

RDStripe

Name	Type	Offset	Format
RDStripe	RAMDAC Control	0x010	Bitfield

Control register

Bits	Name	Read	Write	Reset	Description
0..2	Size	✓	✓	0000.0000	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	✓	✓		Number of rasterizer chips
5...7	Owner	✓	✓		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 DACs

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

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

RDVideoOverlayBlend

Name	Type	Offset	Format
RDVideoOverlayBlend	RAMDAC Control	0x002C	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..5	Reserved	✓	✗	0	
6..7	Factor	✓	✓	X	Proportion to blend main image and overlay, enabled by BlendSrc field of RDVideoOverlay Control Field register. 0 = 0% 0x1 = 25% 0x2 = 59% 0x3 = 75%

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.

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	✓	✓	X	0 = MainKey	1 = OverlayKey
					2 = Always	3 = Blend
3	DirectColor	✓	✓	X	0 = Disabled.	1 = Enabled.
4	BlendSrc	✓	✓	X	0 = Main.	1 = Register.
5	Key	✓	✓	X	0 = Color.	1 = Alpha.
6..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.

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

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

RDVideoOverlayKeyR

Name	Type	Offset	Format
RDVideoOverlayKeyR	RAMDAC Control	0x029	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Red	✓	✓	X	The red component for color key checking is also used to hold the alpha value during alpha test.

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

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

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

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

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

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

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

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

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

RDWClkControl

Name	Type	Offset	Format
RDWClkControl	RAMDAC Control	0x219	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0	Clock	✓	✓	X110.10x0	0=disable 1=enable
1	Lock	✓	✕	X110.10x0	0=not locked 1=locked
2,3	State	✓	✓	X110.10x0	0=drive low 1=drive high 2=Run 3=reserved
4,5	Feedback	✓	✓		
6	Prescale	✓	✓		
7	Reserved	✕	✕		

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

RDWClkDivider

Name	Type	Offset	Format
RDWClkDivider	RAMDAC Control	0x21B	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..6	Value	✓	✓	X000.0001	
7	Reserved	✕	✕		

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

RDWClkMultiplier

Name	Type	Offset	Format
RDWClkMultiplier	RAMDAC Control	0x21A	Integer

Control register

Bits	Name	Read	Write	Reset	Description
0..5	Value	✓	✓	X000. 0001	
6	ClockOut	✓	✓		
7	Reserved	✕	✕		

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

4.8 Region 0 VS GP and VSCtl (0x5000-0x5FFF)

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…5 reserved 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	VActivePolarity B	✓	✓	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	✓	✕	X	
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:

4.9 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).

4.9.1 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 beign 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 BankALowReg, BankAHighReg, BankBLowReg, BankBHighReg instead. This register may be removed from future hardware

4.9.2 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.

4.9.2.1 Sequenced Registers

BankAHighReg

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

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	0x635C index 0x08	Bitfield
	Control register		

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	Bitfield
		index 0x0B	
	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	Bitfield
		index 0x0A	
	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*

GenLockControlReg

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.

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:

4.10 Region 3 Indirect Addressing

Base Address register 3 provides an I/O region which allows any register to be indirectly accessed. The Base Address register must first be enabled by setting the *IndirectIOEnable* bit in **ChipConfig**. The region is defined as 16 bits allocated as shown below.

VGA Indirect accesses to the I/O registers in Region 3 are supported by **VgaIndirectIndex** and **VgaIndirectReg**. The IndirectAccess register must always be selected by a single byte write to **VgaIndirectIndex**, followed by a separate byte read or Write to **VgaIndirectReg** to trigger the internal R4 indirect access.

Both these registers must be enabled before use. The enable bit is bit 1, *IndirectEnable*, in **PCIControlReg** (port 0x3C5 index 0x0).

IndirectAccess

Name	Type	Offset	Format
IndirectAccess	Region 3	0x0C	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	Reserved	✓	✓	0	Accessing any part of these 32 bits triggers an indirect access to the location addressed by **IndirectAddr**. A write here triggers the write of IndirectData into the location. A read here triggers the read of the location into **IndirectData**. The access is further masked by the byte enables specified in **IndirectByteEn**.

Notes:

IndirectAddr

Name	Type	Offset	Format
IndirectAddr	Region 3	0x08	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..28	Offset	✓	✓	0	These bits specify the offset of the location to be accessed.
29..31	Region	✓	✓	0	0 = Base Address 0 1 = Base Address 1 2 = Base Address 2 3-6 = Reserved 7 = ROM Region

Notes:

IndirectByteEnable

Name	Type	Offset	Format
IndirectByteEnable	Region 3	0x00	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..3	Byte Enables	✓	✓	0	These four bits specify the mask to apply to accesses to the location by IndirectAddr. bit 0 set to 1 enables IndirectData byte 0 bit 1 set to 1 enables IndirectData byte 1 bit 2 set to 1 enables IndirectData byte 2 bit 3 set to 1 enables IndirectData byte 3
4..31	Reserved	✓	✕	0x000.0000	

Notes:

IndirectData

Name	Type	Offset	Format
IndirectData	Region 3	0x04	Integer
	Control register		

Bits	Name	Read	Write	Reset	Description
0..31	Data	✓	✓	0	These 32 bits hold the data to be written to, or read from, the location addressed by IndirectAddr. The access is further masked by the byte enables specified in IndirectByteEn.

Notes:

VgaIndirectIndex

Name	Type	Offset	Format
VgaIndirectIndex	Region 3	0xB0 (mono) 0xD0 (color)	Byte

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Data	✓	✓	0	Select **IndirectAccess** register

Notes: Enable using PCIControlReg

VgaIndirectReg

Name	Type	Offset	Format
VgaIndirectReg	Region 3	0xB1 (mono) 0xD1 (color)	Byte

Control register

Bits	Name	Read	Write	Reset	Description
0..7	Data	✓	✓	0	Select **IndirectAccess** register

Notes: Enable using PCIControlReg

INDEX