GLINT R5®

Reference Guide Volume I -

Overview

PROPRIETARY AND CONFIDENTIAL INFORMATION

3D*labs*®

GLINT R5®

Reference Guide Volume I - Overview

PROPRIETARY AND CONFIDENTIAL INFORMATION

Issue 4

Proprietary Notice

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

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

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

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

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

Change History

Document	Issue	Date	Change
172.1.1	1	15/02/2000	Creation
172.1.2	2	06/04/2000	Rework to clarify pipeline structure and vertex loading
172.1.3	3	25/07/2000	Corrections and updates to pin assignments, thermal, memory and other areas.
172.1.3	4	05/03/2001	Gamma comparison Region 0 map, various clerical edits, remove face normals, correct tickmark font, minor format and font changes

Table of Contents

1 Functional Overview

1.1 Introduction

GLINT R5 integrates the GLINT Gamma G2 geometry processor and an enhanced version of the GLINT R4 rasterizer. It is designed to deliver exceptional graphics performance and image quality for 2D and 3D workstation and embedded applications.

The R5 delivers workstation 3D polygon and textured graphics acceleration without compromising 2D desktop performance. Dual memory controllers make the most effective use of up to 128MB of double data rate local memory devices. The integrated 350MHz RAMDAC and 0.18u silicon process, together with other performance improvements, result in a blend of real world performance and quality unusual even in much more expensive devices.

Performance engineering features include:

- Integrated 350MHz RAMDAC
- 0.18u silicon process
- Dual memory controllers for up to 128MB double data rate devices
- Full geometry acceleration
- Vertex array processing
- Dual on-chip setup units
- Virtual texture caching
- Hierarchical display list DMA
- AGP2X/4X
- Multiple rasterizer capability
- Multiple parallel transform and lighting units (for D3D lighting, stripe filtering and and OpenGL 1.2 features).

...and others described throughout the *Reference Guide* and *Programmer's Guide*.

R5 also offers improved Linux and X Windows hardware support including full color 24-bit RGB 3-D windows on the 8-bit pseudocolor X desktop, X-style cursors and other X-compatibility enhancements.

GLINT R5 Summary	
Performance	
Vetex rate, Mvertices/sec, infinite directional lights, infinite viewer, no attenuation	Chip in character-ization – see Power-Point overview for performance estimates
Vetex rate, Mvertices/sec, multiple local lights, infinite viewer, no attenuation	
Fill rate, Mpixels/sec, depth-buffered, bilinear multi-textured, 16bit pixel/depth/texture	
Fill rate, Mpixels/sec, depth-buffered, trilinear textured, 32bit pixel/depth/texture	
Core clock (MHz)	
Memory clock speed (MHz)	
Transform and lighting clock (MHz)	
Interface and Memory	
Memory bus - dual 64-bit controllers	128bits
Max. memory	128Mbytes
Memory clock independent of core clock	✔
SGRAM Block write support	✔
AGP with sideband and pipelining, 2x/4x protocol	✔
AGP I/O Signal voltage - dual voltage	1.5V / 3.3V
PCI Bus Aperture	128 bits
Multiple rasterizer support and stripe filtering	✔
Native OpenGL and D3D vertex interfaces	✔
Write combined command FIFO	✔
Bounding box and bounding volume tests (hierarchical display lists)	✔
Hierarchical low-latency command DMA	2 levels
Rectangle download and upload DMA	✔
Memory devices supported: DDR SDRAM, DDR SGRAM, SDR SDRAM, SDR SGRAM	✔

GLINT R5 Summary	
Primitives, Parameter Processing, Culling, Lighting	
Points, lines, triangles, strips, fans	✔
Render, feedback, select and polymode	✔
Number of vertex data arrays	16
Diffuse and specular color	✔
Frustrum clipping, user clip planes	✔
Backface cull	✔
Stripe filtering	✔
OpenGL RGB model lighting	✔
OpenGL CI model lighting	X
D3D model lighting	✔
Two sided lighting	✔
Maximum multiple lights	16
Specular and spotlighting	✔
Attenuation	✔
Material - emissive, ambient, diffuse, specular; front and/or back face	✔

GLINT R5 Summary	
Texture and Fog	
Multiple textures	2
Per pixel perspective correct texture	✔
Per pixel true corrected area mipmapping	✔
Per pixel corrected area mipmapping for dual textures	✕
Per polygon mipmapping	✔
Border color	✔
Single-pass bump mapping with surface texture	✔
Hardware texture paging	✔
Per polygon and per pixel mipmapping[1]	✔
Bilinear and trilinear filtering	✔
Anisotropic filtering	emulated
3D textures, RGBA and palletized textures	✔
On-chip texture cache	2Kx2K
Virtual texture management, logical texture addressing	✔
Fog table	✔
Range based fog, Eye Z fog, user-supplied fog, linear, exponential and exponential squared fogs	✔
Object linear, eye linear and sphere map texture generation	✔
Texture wrap (D3D)	✔
3D Features	
Max. Z-buffer depth (bits)	32
All D3D and OpenGL Depth and Stencil Modes	✔
Non-Linear depth format	✔
W-Buffer Emulation with Non-Linear Z-buffer (Direct3D)	✔
Destination Alpha, Alpha and color key tests	✔
All OpenGL and Direct3D blend modes	✔
OpenGL 1.1/1.2 compliant / ready	✔
Anti-aliasing - OpenGL and D3D	✔
YUV - RGB conversion	✔

[1] Per-pixel mipmapping is available on Texture0 only.

GLINT R5 Summary	
2D Features	
Logic Ops (foreground and background)	✔
Pixels processed in parallel (solid fill)	64
Write per bit masks	✔
Comprehensive vertex formatting and data management for flexible, minimum redundancy 2D and 3D primitive setup	✔
Statistic collection	✔
Hardware context save/restore	✔
Stretch and high-speed blts	✔
Font caching	✔
DMA Packing / unpacking on output / input	✔
32x32 stipple pattern	✔
Run-length encode / decode	✔
Anti-aliased line and triangle setup	✔
Bitmaps and rectangular scissor clip	✔
Buffer Formats	
Unified framebuffer/localbuffer	✔
Dual writes through bypass	✔
Flexible GID, stencil and depth formats	✔
Stencil planes	8
GID bits	4
Simultaneous framebuffer accesses	4
Any width framebuffer/localbuffer texture	✔
2d buffer tiling	✔

GLINT R5 Summary	
Video / DVD	
Hardware video overlays	1
15 color on-chip cursor	✔
Windows 2000 blended cursor - transparency levels	8
Display resolutions > 2048x1536x32 @ 70Hz	✔
Hardware scaling and filtering	✔
Digital Video out	✔
MPEG Motion compensation	✔
Scatter/gather download DMA	✔
Flat panel LCD support, parallel bus panel output	✔
LUT accuracy	10 bits
Video Genlock to any video source	✔
Integrated RAMDAC, max clock	350MHz
Video overlay blend	3 bit
Video overlay with stretch and bilinear filtering	bilinear
Power Management	
DPMS for monitors	✔
Memory array power-down	✔
Dynamic clock control	✔

Table 1.1 GLINT R5 Enhancement Summary

1.2 Changes from previous chipsets

GLINT R5 integrates both geometry and rasterizer processors. It therefore shares functional characteristics with earlier members of both the Gamma and GLINT/Permedia families.

Compared with the GLINT R4 and Gamma G2 the R5 incorporates changes and enhancements in the following areas:

1.2.1 Command Processor

A new command processor includes support for hierarchical display lists and vertex arrays. Hierarchical display lists allow DMA commands to be put inside another DMA buffer, improving the efficiency of display list management. This also allows bounding box tests to conditionally execute DMA buffers so that non-visible areas of a large model can be culled very quickly. GLINT R5 allows 2 levels of nesting.

Vertex arrays are common to OpenGL and D3D and allow vertex data to be read from arrays that are separate from the command buffer. They also allow access to vertex data indirectly via index buffer arrays for increased flexibility and vertex efficiency. Indexed arrays let applications maximize vertex sharing between primitives to reduce the number of vertices that are processed more than once. For example, a primitive can be set up with

textures coming from a far store of textures in a contiguous group, with subsequent primitives re-using vertex data while incrementing the texture offset.

GLINT R5 supports 16 vertex arrays and 1 index array, where R4 supports one index array, one vertex array and non-hierarchical display lists.

1.2.2 Transform and Lighting Processor

A transform and lighting processor has been added in front of the rasterizer pipeline. It includes all processing stages needed to transform and light a primitive. The units are derived from GLINT Gamma G2, but with improvements including support for two textures, the calculation of the inverse transpose of the matrix, D3D compliant lighting, stripe filtering, Autodesk normal support, and OpenGL 1.2 features. The GLINT R5 implementation uses a single geometry pipe with multiple lighting units in parallel to improve performance. Light addressing has changed from Gamma2 with the addition of the **LightNumber** register, replacing the number field in individual lighting operations[2].

1.2.3 Small Primitive Rasterization

The suspend-resume read synchronization operation usually limits the rasterization performance of small primitives. GLINT R5 continues the performance improvements started with the R4 Read Monitor unit by adding a cache to further reduce the frequency of resynchronization.

1.2.4 Memory Interface

GLINT R5 supports double data rate (DDR) memory devices, which transfer data on both rising and falling edges of the clock strobe. R5 uses dual 64-bit memory controllers to maintain the 128-bit data path at the DDR rate[3]. Refer to Chapter 9, *Memory Systems,* in Volume IV of the *GLINT R5 Reference Guide* for details of devices supported and typical layout configurations.

---

[2] E.g. **LightAmbientIntensityRed** used to be **Light*n*AmbientIntensityRed**

[3] Test show that dual 64-bit controllers are more efficient than a single 128-bit controller at the higher transfer rates required.

1.3 Functional Blocks

The major functional blocks are shown in Figure 1-1.

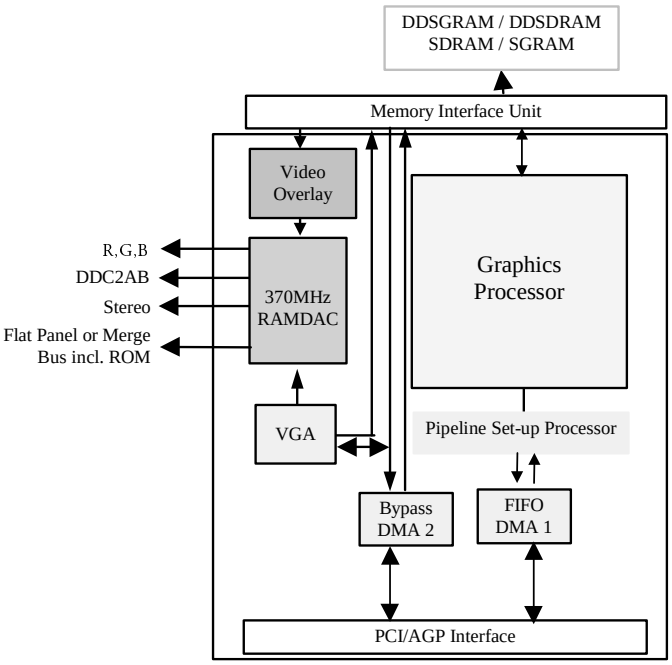

Figure 1-1 Chip Level Block Diagram

1.4 AGP/PCI Interface

The R5 AGP/PCI interface conforms to *PCI Local Bus Specification Rev. 2.2* and *AGP Interface Specification Rev. 2.0*. R5 is a PCI Local Bus Target, a PCI Local Bus Read/Write Master, and an AGP read master with support for pipelined reads and sideband addressing.

The bus interface control unit monitors the AGP/PCI Bus and determines when a valid cycle for the chip occurs. It handles the bus cycle, generating internal select signals to route the access to the target unit. The associated DMA engines are described in section 1.4.6 below.

AGP 4X is Intel's high performance component level interconnect targeted at 3D display applications, based on a 66MHz PCI specification baseline. GLINT R5's AGP implementation includes:

- 266 MHz transfer rate
- DMA and execute mode support
- Sideband addressing

Implementing these features enables the R5 to achieve better than 1 GByte per second bandwidth from the host for instructions, textures and video data (limited by the host system throughput).

The add-in slot defined for AGP uses a connector body which is not compatible with the PCI connector. Boards designed for use in an AGP slot are not mechanically interchangeable with PCI boards.

GLINT R5 also remains AGP2X compliant, including support for both 3.3VDC and 1.5VDC logic.

1.4.1 Unified 2D/3D/Video Integrated Graphics Processor

For further information on the functionality of the graphics processor (GP), refer to chapter 5, Graphics Registers in volume III of the *GLINT R5 Reference Guide*, and to the *GLINT R5 Programmer's Guide*.

1.4.2 Memory Interface

The memory clock speed can exceed the core clock speed although both are provisionally clocked at 150MHz. The two memory controllers each support 64Mbytes so the maximum memory that can be fitted is 128MBytes. Single data rate memories are still supported, as is SGRAM. DDR SGRAM is generally of little use because block fills do not have column masks, but there are some suitable parts available. The supported memories have been limited to:

Type	Density (Mbits)	Width (bits)	Board total (Mbytes)
DDR SDRAM	64	8	128
DDR SDRAM	128	16	128
DDR SDRAM	64	16	64
DDR SDRAM	128	32	64
DDR SDRAM	64	32	32
DDR SGRAM	32	32	16
SDR SDRAM	64	8	128
SDR SDRAM	64	16	64
SDR SDRAM	64	32	32
SDR SGRAM	32	32	16

Table 1.2 Supported R5 Memory Devices.

For more information about Memory System layout and timings, see Memory System in the *GLINT R5 Reference Guide*, volume IV.

1.4.3 SVGA

The on-chip SVGA unit is register level compatible with standard VGA devices and requires no software emulation. It natively supports all standard VGA modes and certain VESA VBE extended modes.

The following standard VESA VBE extended video modes are supported - those not directly supported by the SVGA unit can be supported using the Graphics Processor.

Mode (hex)	Pixels	Colors	Window-ed	Lin-ear	Support-able in SVGA	Support-able in GP
0x100	640x400	256	✓	✓	✓	✓
0x101	640x480	256	✓	✓	✓	✓
0x103	800x600	256	✓	✓	×	✓
0x105	1024x768	256	✓	✓	×	✓
0x107	1280x1024	256	✓	✓	×	✓
0x109	320x200	32K (5:5:5:1)	✓	✓	×	✓
0x10D	320x200	64K (5:6:5)	✓	✓	×	✓
0x10F	320x200	16.8M (8:8:8)	✓	✓	×	✓
0x110	640x480	32K (5:5:5:1)	✓	✓	×	✓
0x111	640x480	64K (5:6:5)	✓	✓	×	✓
0x112	640x480	16.8M (8:8:8)	✓	✓	×	✓
0x113	800x600	32K (5:5:5:1)	✓	✓	×	✓
0x114	800x600	64K (5:6:5)	✓	✓	×	✓
0x115	800x600	16.8M (8:8:8)	✓	✓	×	✓
0x116	1024x768	32K (5:5:5:1)	✓	✓	×	✓
0x117	1024x768	64K (5:6:5)	✓	✓	×	✓
0x118	1024x768	16.8M (8:8:8)	✓	✓	×	✓
0x119	1280x1024	32K (5:5:5:1)	✓	✓	×	✓
0x11A	1280x1024	64K (5:6:5)	✓	✓	×	✓
0x11B	1280x1024	16.8M (8:8:8)	✓	✓	×	✓

Table 1- 3 VESA VBE Graphics Modes

The following VESA VBE text modes are supportable in the SVGA:

Mode (hex)	Characters (col/row)
0x108	80x60
0x109	132x25
0x10A	132x43
0x10B	132x50
0x10C	132x60

Table 1- 4 VESA VBE Text Modes

R5 allows VESA bankswitching to be done through the bypass to enable additional VESA mode support. ModeX is also supported.

1.4.4 Video Overlay

The video overlay is used to display incoming video data on screen. The overlay selection is based on a transparent color, the overlay key, which can be any RGB color or alpha value. Optionally, the overlay can be blended with the main image by using a 3-bit blend factor. A filter process supports zooming and shrinking at any rate. It combines four pixels into one by using bilinear filtering to achieve best results. The filtered output can be converted from YUV to RGB color space format.

The video overlay already supports blending of the overlay surface with the desktop, so a small change has been made to allow its use as a cursor. This facility is particularly useful for Windows 2000 which uses a blended cursor by default. Note that this supports 8 levels of transparency only[4].

1.4.5 DMA

Two DMA controllers - the FIFO DMA controller and the Bypass DMA controller - can be used to trigger either the internal PCI bus master or AGP bus master to read from devices on the PCI Bus to either the Graphics processor input FIFO or the bypass FIFO. These are zero wait-state with programmable block size.

1.4.5.1 Bypass DMA Controller

The bypass unit is used to access 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**). These can run concurrently. 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.

The command mechanism allows for full gather-scatter DMA. This gives the ability to access non-contiguous system memory, important e.g. for virtual texture management.

1.4.5.2 FIFO DMA Controller

The FIFO DMA controller allows GLINT R5 to read data directly into the Graphics Core input FIFO or directly from the output FIFO. The input FIFO is 256 words deep, the output FIFO is 8 words deep. As with the Bypass Controller, read/write accesses can run concurrently and scatter/gather is supported. The FIFO Controller supports:

- Fast texture/image uploads and downloads
- Fast software MPEG2 download, fast frame capture

The FIFO DMA controller can also be used to write to devices on the PCI Bus using data from the Graphics processor output FIFO.

---

[4] Cursors for Windows 2000 with 16 levels of transparency are still available if the blended cursor is implemented in software instead of hardware, but performance may be affected.

1.4.5.3 Interrupt Controller

- End-of-DMA - allows DMA chaining
- VSYNC - efficient double buffering
- Scanline - special effects
- Texture invalid
- Bypass DMA interrupt
- I2C start condition - alert host to start of I2C transfer
- Sync - indicates graphics core is idle
- Error - e.g. writing to a full FIFO

1.4.6 Video Streaming

R5 supports digital video output. The 24-bit streamed output is designed to work with common PAL/NTSC encoders or flat panel controllers via an I2C bus.

1.5 ROM support

R5 supports a Flash ROM. This ROM may store code needed for device-specific initialization and the SVGA BIOS. This is configured via the **VideoMergeBus** pins and, optionally, a QuickSwitch as described in the *R5 Reference Guide Volume IV*, "Reset".

2

Address Maps and Regions

2.1 PCI Configuration Region

The PCI Configuration Region provides information that satisfies the needs of current and anticipated system configuration mechanisms. Configuration parameters are read and modified via the Configuration registers (see volume II) and on the trailing edge of a reset by configuration pins (see volume IV, Chapter 10 - Reset).

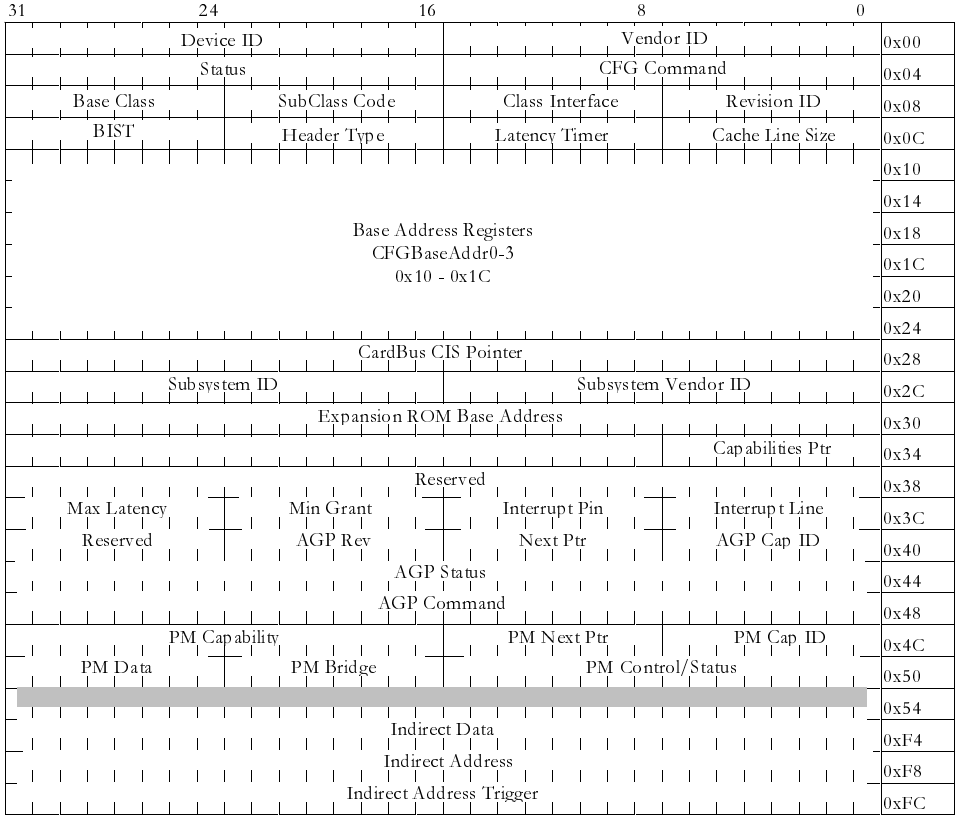

Figure 2.1 PCI Configuration Region

2.2 Region Zero Address Map

The R5 region zero address map is shown in Table 2-1: To clarify changes since Gamma I, the G1 map is shown on the right.

Note: The G1 Control Status registers have largely been superceded. Only those in the range 0000.0000 - 0000.0088 are unaffected.

So **DMAControl** (.0060). **FIFODiscon** (.0068) and **ResetStatus** (at .0000) are unaffected, **DMAAddress** and **DMACount** (renamed **ControlDMAAddress** and **ControlDMACount**) at .0028 and .0030 are still supported, but **CommandMode** (.00C40) is not. The DMACommand unit registers (**CommandErrorFlags** (.0C58), **CommandIntEnable** and **CommandIntFlags**) used to clear outstanding flags are now supported via **IntEnable** (.0008), **IntFlags** (.0010) and **ErrorFlags** (.0038).

Address Range	Region Select (R5)	Byte Swap/ Write Com-bined	Gamma1
0000.0000 -> 0000.01FF	Control Status	No	Control Status
0000.0200 -> 0000.02FF	Test Control	No	
0000.0300 -> 0000.03FF	Bypass Control	No	
0000.0400 -> 0000.07F8	Repeats Control, Test & Bypass Decodes	No	Target registers - passed through
0000.07FF -> 0000.0FFF			*Control Status*
0000.1000 -> 0000.1BFF	Memory Control	No	Target Control
0000.1C00 -> 0000.1FFF	FB Sync Data FIFO	No	
0000.2000 -> 0000.2FFF	GP FIFO Access	No	GP FIFO Access
0000.3000 -> 0000.33FF	Video & Overlay Control	No	Target Control
0000.3400 -> 0000.37FF	DMA Arbiter	No	
0000.3800 -> 0000.3FFF	Reserved	No	
0000.4000 -> 0000.4FFF	RAMDAC	No	
0000.5000 -> 0000.57FF	Reserved	No	
0000.5800 -> 0000.5FFF	VSCtl	No	
0000.6000 -> 0000.6FFF	VGA Control	No	
0000.7000 -> 0000.7FFF	TextureData FIFO	No	
0000.8000 -> 0000.FFFF	GP Registers	No	GP Registers
0001.0000 -> 0001.01FF	Control Status	Yes	
0001.0200 -> 0001.02FF	Test Control	Yes	
0001.0300 -> 0001.03FF	Bypass Control	Yes	
0001.0400 -> 0001.0FFF	Repeats Control, Terst & Bypass Decodes	Yes	
0001.1000 -> 0001.1BFF	Memory Control	Yes	
0001.1C00 -> 0001.1FFF	FB Sync Data FIFO	Yes	
0001.2000 -> 0001.2FFF	GP FIFO Access	Yes	

Address Range	Region Select (R5)	Byte Swap/ Write Com-bined	Gamma1
0001.3000 -> 0001.33FF	Video & Overlay Control	Yes	
0001.3400 -> 0001.37FF	DMA Arbiter	Yes	
0001.3800 -> 0001.3FFF	Reserved	Yes	
0001.4000 -> 0001.4FFF	RAMDAC	Yes	
0001.5000 -> 0001.57FF	Reserved	Yes	
0001.5800 -> 0001.5FFF	VSCtl	Yes	
0001.6000 -> 0001.6FFF	VGA Control	Yes	
0001.7000 -> 0001.7FFF	TextureData FIFO	Yes	
0001.8000 -> 0001.FFFF	GP Registers	Yes	

Table 2.1 Region Zero Address Map

2.3 PCI Address Regions

GLINT R5 has six base address regions, as shown in Table 2-2:

Region	Address Space	Bytes	Description	Comments
Config	Configuration	256	PCI Configuration	PCI special
Zero	Memory	128K	Control Registers	Relocatable
One	Memory	128M	Memory Aperture One	Relocatable
Two	Memory	128M	Memory Aperture Two	Relocatable
ROM	Memory	64 K	Expansion ROM	Relocatable
VGA	Memory & I/O	—	VGA Address	Optional and Fixed

Table 2.2 GLINT R5 PCI Address Regions

2.4 Pipeline Architecture

GLINT R5 uses an integrated pipeline with a variety of additional units to provide enhancements and services such as switchable 2D/3D Routing, hardware Context Dump/Restore, Statistics and Bypass modes. The units are primarily virtual, i.e. collections of related functionality, rather than physical. Units operate asynchronously but are in practice synchronized to a great extent by their common clock domains.

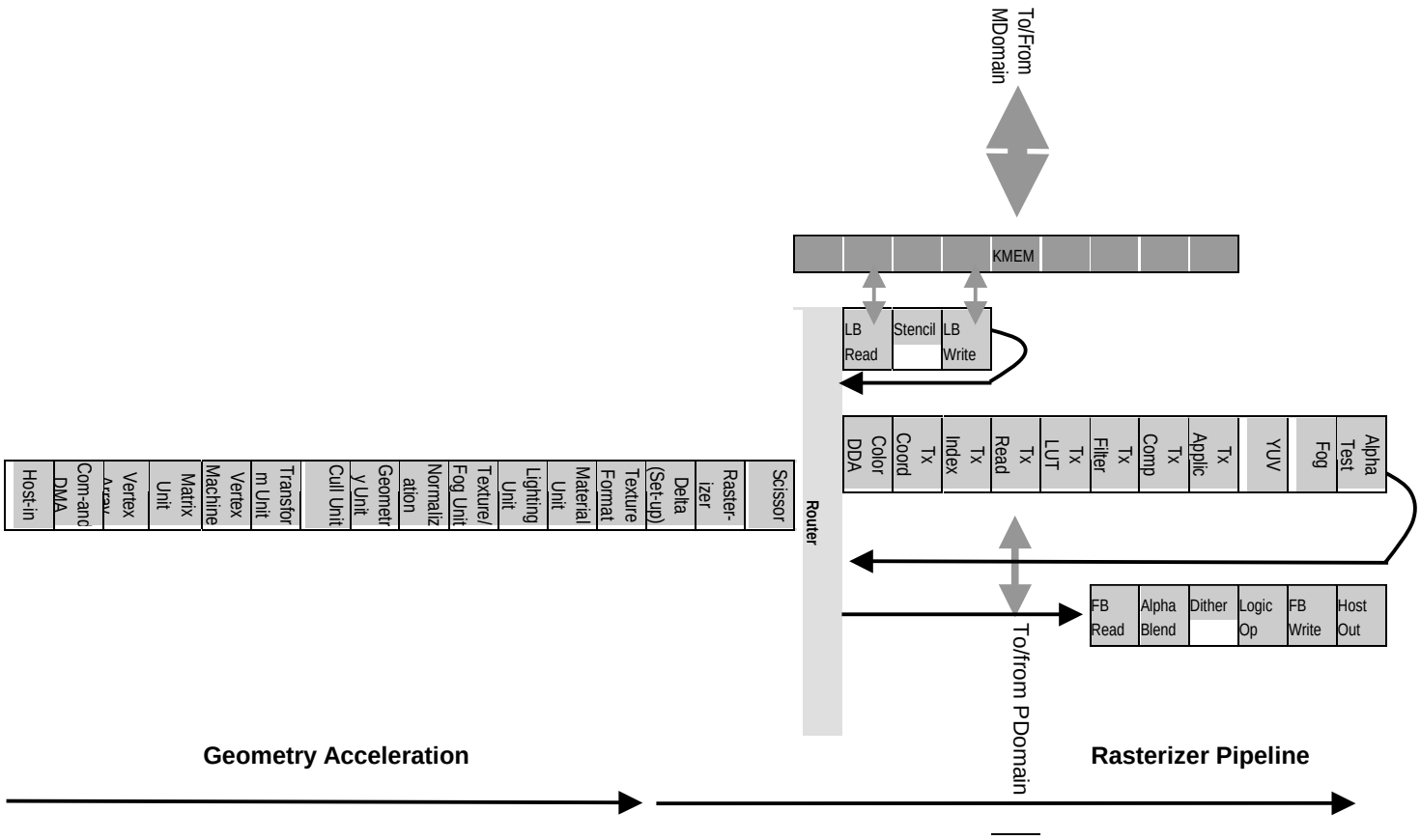

Figure 2.2 Pipeline Architecture

3

Video Unit and RAMDAC

GLINT R5 incorporates a high performance 350MHz RAMDAC which accepts data from the video unit, processes it to integrate cursors, overlays, genlock etc. and passes it on to the DACs to output as analogue RGB and horizontal and vertical syncs. The video unit and RAMDAC should be configured to display the framebuffer data with the format, resolution, and refresh frequency required.

3.1 RAMDAC Characteristics

R5 incorporates a 350MHz 128-bit RAMDAC. It supports packed pixel formats with color depths of 8, 16, and 32 bits per pixel and typical screen resolutions up to 4096 horizontal.

The RAMDAC contains three phase locked loops (PLLs). One is dedicated to generating the Dot Clock (DClk) and the derived Video Clock (VClk). The others generate the core clock (KClk) and Transform & Lighting Clock (TClk).

The RAMDAC contains a 64x64x2 bit cursor array to support a 2, 4, or 16 color hardware cursor with cursor shapes cache. Supported cursor types include Windows, X, 3 color and 15 color. Optionally, when the main overlay is not in use[5] the video overlay can be configured as a cursor. This allows unlimited size, a full range of colors, and the option of using the overlay blend hardware for transparent cursors.

- High resolution 350 MHz, 128-bit RAMDAC
- Supports screen resolutions up to 4096 horizontal
- Supports packed pixel formats
- Color depths of 8, 16 and 32 bits/pixel
- Dot clock phase-locked loop (PLL)
- Triple pseudo-10bit D/A converters
- 64x64x2-bit cursor array to support a 2, 4 or 15 color hardware cursor with cursor shapes cache
- Analog and digital striping

3.1.1 Display Resolutions

R5 supports all the standard screen resolutions at ergonomic refresh rates. For each resolution in the table below, the frequency figure represents the refresh rate supported using the VESA generalized Timing formula with 50% of the memory bandwidth used for screen refresh and 50% for drawing assuming a pixel clock of 350MHz.

[5] If the main overlay is in use the stacking order will be incorrect - the video overlay lies beneath the cursor and main overlays..

Resolution	Refresh Rate (Vertical Retrace in Hz)
320x200	1238
640x480	554
800x600	404
1024x768	269
1152x864	220
1280x1024	172
1600x1200	122
1920x1280	96
2048x1536	77
4096x1024 (Hypothetical)	60

Table 3-1 Display Resolutions

Resolutions are driver- and memory- limited. A 32MB framebuffer for example can support 2048x1200 @ 32bit colour, 32bit Z; or 2048x1536 @ 32bit colour, 16bit Z.

3.1.2 Display Data Channels (DDC)

Two control lines are dedicated on R5 to support DDC1 and DDC2AB+ monitor configuration utilities. The DDC2 serial bus is independent of the serial bus in the video stream interface.

3.2 Display Timing Values

Parameter	Hex	Decimal
Htotal	0x065	101
HsStart	0x003	3
HsEnd	0x00B	11
HbEnd	0x016	22
HgEnd	0x016	22
VTotal	0x1F5	501
VsStart	0x000	0
VsEnd	0x003	3
VbEnd	0x016	22
ScreenStride	0x050	80
ScreenBase	0x000	0
VideoControl	0x029	41

Table 3-2 Timing Values for 640x480 16 BPP 75Hz

Parameter	Hex	Decimal
HTotal	0x103	259
HsStart	0x00A	10
HsEnd	0x01E	30
HbEnd	0x03C	60
HgEnd	0x03C	60
VTotal	0x272	626
VsStart	0x000	0
VsEnd	0x003	3
VbEnd	0x01B	27
ScreenStride	0x0C8	200
ScreenBase	0x000	0
VideoControl	0x029	41

Table 3-3 Timing Values for 800x600 32 BPP 75Hz

3.3 Multi-rasterizer and Genlock implementation

3.3.1 Clocks

R5 involves a number of clock domains: the Pixel Bus (Dot Clock or DClk), the Memory interface (MClk), the Geometry Pipeline (TClk), Rasterization setup (SClk), the Core Graphics Clock (KClk) and the PCI Interface (PClk). PLLs support KClk, TClk and DClk (and the VClk derived from it).

The KClk and TClk PLLs reset to approximately 50MHz. All figures assume a 14.31818MHz external reference. An external signal, can put the PLLs into bypass mode in which TClk, KClk and DClk are taken from pins, and the PLLs have no effect.

Two other clocks, MClk (for the memory) and SClk (for the geometry setup unit) do not have PLLs but are generated from other clock sources. All clocks have optional power management which reduces their frequency to ½ when the chip is idle. The modes supported are:

3.3.1.1 Dot Clock PLL

The Dot Clock PLL has four sets of control registers. The set to use can be register-selected without waiting for reset. The hard reset states for the DClk PLL control registers are:

Register set	Frequency
0	25.057MHz
1	28.278MHz
2	Undefined
3	Undefined

Table 3-4 DClk PLL Reset Values

The RAMDAC PLLs generally are reset on power-up only - for a more comprehensive description of the affected registers see the *Reset* chapter in the Volume IV.

The DClk PLL can be configured to take its input from the 14MHz oscillator or a separate external clock. An external clock is required when genlocking to another video source. There is also a programmable delay to allow compensation for skewing relative to the input (see DClk Programming, below).

The KClk and TClk PLLs reset to approximately 50MHz. All figures assume a 14.31818MHz external reference. An external signal can put the PLLs into bypass mode in which TClk, KClk and DClk are taken from pins, and the PLLs have no effect.

Two other clocks, MClk (for the memory) and SClk (for the geometry setup unit) do not have PLLs but are generated from other clock sources. All clocks have optional power management which reduces their frequency to ½ when the chip is idle. The modes supported are:

Clock	Power Management Mode
PClk	PCI clock
HalfPClk	Half PCI clock
KPLL	Output of the K PLL (no power management support)
HalfKPLL	Half the frequency of K PLL (with power mangement support, reduces to 1/4 frequency in low power)
ExternalClk	Take clock from pin (one pin for each clock)
HalfExternalClk	Half the frequency of the clock on the pin (one pin for each clock)
TPLL	Output of the T PLL (no power management available)
HalfTPLL	Half the frquency of the T PLL (with power managment support, reduces to 1/4 frequency in low power)

Table 3-5 Power Management Clock Modes

3.3.2 Striping

Digital striping passes video data between rasterizers on a dedicated video merge bus. If there are two rasterizers: a direct point to point bus is used, one rasterizer ('back end') sends data and syncs to the other ('front end') which combines the input data with its own data to produce a complete display. Clocking is controlled by the Front End device.

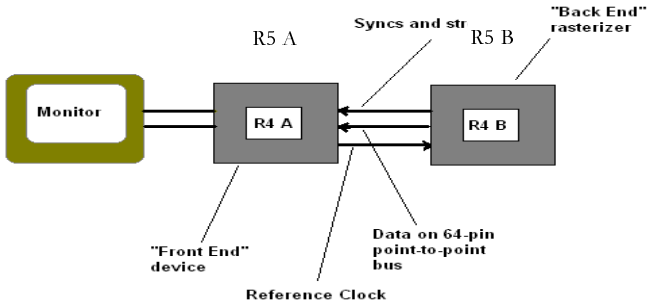

Figure 3-1 Dual-rasterizer configuration

3.3.2.1 Broadcast Mode

If there are more than two rasterizers they use a common broadcast bus and both clock and syncs are provided by the front end chip.

Note: This requires detailed knowledge of board latency characteristics. We recommend that developers contact 3Dlabs Technical Support for more information.

In ***Broadcast Mode*** the software is responsible for ensuring that the chips are properly synchronized before enabling striping to avoid bus contention.

Note: Although the data is shared there are three strobe inputs (in the figure below) which are OR'd within the receiving chip to avoid reflections on the strobe tracks. The strobe output is independent of the inputs.

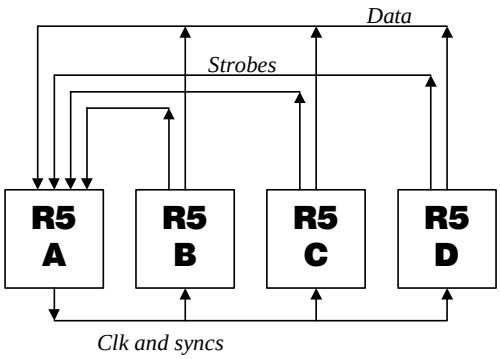

Figure 3-2 Multi-rasterizer configuration

3.3.2.2 Buffered Pipeline

A third mode uses pipeline registers to rebuffer the data between chips making it easier to support several chips and connections between separate boards. The strobe from each chip is used to clock the data into a register. In this mode the clocks run all the time to keep the registers valid, so the clock into the master chip must be gated by the blank signal that indicates when data is valid. This blank signal also controls the tristate of the registers.

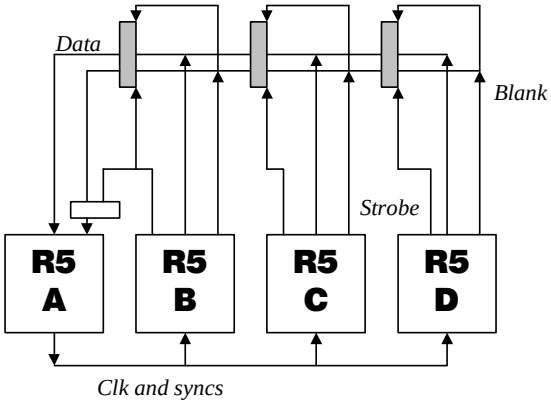

Figure 3-3 Multi-rasterizer buffered pipeline configuration

3.3.2.3 Timing

Pixel sizes 8, 16, and 32 bits can have any number of VClks for any of the horizontal timing parameters (i.e. front porch, sync, back porch, active period).

Note: It is normal for all timing parameters in any pixel size to be a multiple of 8.

3.3.3 Rasterizer and PLL/Clock Setup

The rasterizer which drives the monitor (the "Front End" device) provides sync and clock in broadcast bus configurations as shown in Figures 3-2 and 3-3 above. However in dual chip configuration the syncs and strobe, together with the video merge bus data, is provided by the Secondary or 'back end' chip as shown in Figure 3-1.

The recommended procedure to configure the front- and back end devices in a dual chip configuration is:

1. Set up clocks
2. Configure striping
3. Configure RAMDACs
4. Set up Genlocking

3.3.3.1 Set up Dot Clock PLLs and Clock Selects (dual rasterizer configuration)

Clock setup is required to ensure that board-level latencies are controlled. It scales the video bus pixel clock (whether generated by internal or external oscillator) to a frequency compatible with use across the board, and sets DClk on the front- and back end chips to adjust external clocks to a reasonable internal frequency using the PLL multipliers.

This is controlled by the **RDMergeBusControl** and **RDDClkControl** registers as shown below:

1. *Isolate the chips from the bus while setting up*

   To avoid contention problems during setup, reset **RDMergeBusControl** *DataEnable* and *SyncEnable* before making any other changes**.**

2. *Set Front and Back End Devices to External Clock*

   In **RDDClkControl** set *Clock*=Enable and *Source*=External. This directs both devices to use their **VideoMergeClockIn** pins where an external video source is involved. (If no external video is being configured, leave the front end chip clock *Source* = Internal.)

3. *Adjust the Merge Bus Clock Rate*

   The external reference frequency needs to be scaled by the front end device before being applied to the merge bus. Typically, the external reference is scaled to suit the display frequency requirement, depending on pixel depth (and board layout). The 64-bit bus bandwidth means that we reduce the dot clock proportionally to keep the pixel rate constant for different pixel depths:

   - 32 bpp - divide DClk by 2
   - 16 bpp - divide DClk by 4
   - 8 bpp - divide DClk by 8

   This is controlled by the **RDMergeBusControl** *ClockOut* parameter.

4. *Adjust the Front End Device Dot Clock PLL*

   The external reference frequency will probably also need to be adjusted for internal use by the front end device. This uses the same DClk PLL equation and registers in the front end device as are used in the next step to reprogram the back end DClk PLL.

5. *Reprogram the Back End Device Dot Clock PLL*

The output merge bus clock rate from the front end device needs to be brought back up to a consistent rate for internal use by the back end chip(s). This is handled by the Dot Clock PLL. The DClk frequency may be locked to the PLL's or the external reference clock - the source is selected by the **RDDClkControl** indirect RAMDAC register.

There are four sets of DClk registers, of which any one can be selected as required without reprogramming the DClk PLL. The selection is controlled by the **VClkRDACCtl** register (bits 1 and 0).

Each set of DClk PLL control registers comprise 3 registers, **RDDClk*n*PreScale, RDDClk*n*FeedbackScale** and **RDDClk*n*PostScale,** used to control the output frequency according to the following formula:

*Output Frequency = (Frequency of reference clock * ClkFeedbackScale) / (ClkPreScale * (1 << ClkPostScale))*

The first two sets of DClk control registers are configured at reset to generate 25.057MHz (Set 0), and 28.278 MHz (Set 1)[6] The second pair of DClk control registers are un-initialised at reset and need to be populated with appropriate values if required.

So for example, if the external reference clock is 14.31818MHz, to return it to a display rate of 28.278MHz set **VClkRDACCtl** *VClkCtl*[0,1] =01 (i.e. select **RDDClk1Prescale** etc.)

6. *Enable Video Bus Clock and Data*

   Set the **RDMergeBusControl** *DataEnable* and *SyncEnable* =1 (enabled) after making your configuration changes.

3.3.3.2 Configure Striping

This sets up the number and identity of rasterizers and stripe characteristics.

Striping is enabled in the Video Unit **MiscControl** register:

StripeMode	= 1 (enable)
Count	= 1 (i.e., 2 rasterizers)
StripeID	= rasterizer owning current stripe; stripe 0 = master, stripe 1 = slave.
StripeOffset	= vertical pan value, the number of lines panned modulo 256
Size	= stripe height in lines
CompactFB	= compress FB where space is allocated for unused rasterizers (since the stripes map to different portions of each R5's framebuffer).

*Note: The **MiscControl** register controls digital striping. It should not be confused with the **RDStripe** register which controls analogue striping.*

[6] These frequencies assume that the external reference clock is 14.31818MHz.

3.3.4 Configure the RAMDAC

This sets up the individual rasterizers for digital striping. In the **RDMergeControl** register, set the following parameters:

Field	Front End Device	Back End Device
ClockIn	=0 (use internal, unless genlock to external source is required)	=1 (use external)
ClockOut	=divide ratio (above)	=0
SyncIn	=1	=0
SyncOut	=0	=1 (drive external sync lines inactive)
DataIn	=1	=0
DataOut	=0	=1
DataOr	=0	=0
Broadcast	=0	=0

Depending on the board layout, it may be necessary to set a value for **RDMergeSkew** to compensate for latency in long traces etc.

3.3.4.1 Set up Genlocking

1. In the **HsOffset** register set the offset value to 2. This is used to compensate for clocking in genlock. (The start of the locked hsync is **HsStart** plus an **HsOffset,** usually = 2. **ScreenStride** can usually be ignored.)
2. Enable syncs and data by setting **RDMergeBusControl** *DataEnable* and **RDMergeBusControl** *SyncEnable* = 1.

3.3.5 Programming The Clocks

Except for the PCI Clock (PClk), R5 clock domains can be configured to use either an external reference clock or one of the internal Phase-Locked Loops (PLL's). All clock configuration (including PLL setup) is through registers in the RAMDAC. The D,M, S, and K Clocks are controlled by the indirect RAMDAC registers defined in Volume II of the *Reference Guide*.

3.3.5.1 Dot Clock Programming

DClk ("Dot Clock") is used to control the frequency of the video output from the RAMDAC, and must be set to a frequency suitable for the display resolution and refresh rate.

The DClk frequency may be locked to the PLLs or an external reference clock. The source is selected by the **RDDClkControl** indirect RAMDAC register.

If DClk is driven by the PLL (**RDDClkControl** *Source* field = 0) then the frequency is controlled by one of four sets of PLL registers: 0, 1, 2 and 3. Only one of them can be selected at a time. The selection is controlled by the **VClkRDacCtl** register (bits 1 and 0) located at offset 0000.0040h of Memory Region Zero. This allows several alternate

frequencies to be configured and the required frequency selected without reconfiguring the PLL. The set selection based on VidCtl signals from PCI/VGA is as shown below:

```
if (VidCtl == 0)
{
      DClkFeedbackScale = RDDClk0FeedbackScale;
      DClkPreScale = RDDClk0PreScale;
      DClkPostScale = RDDClk0PostScale;
}
else if (VidCtl == 1)
{
      DClkFeedbackScale = RDDClk1FeedbackScale;
      DClkPreScale = RDDClk1PreScale;
      DClkPostScale = RDDClk1PostScale;
}
else if (VidCtl == 2)
{
      DClkFeedbackScale = RDDClk2FeedbackScale;
      DClkPreScale = RDDClk2PreScale;
      DClkPostScale = RDDClk2PostScale;
}
else if (VidCtl == 3)
{
      DClkFeedbackScale = RDDClk3FeedbackScale;
      DClkPreScale = RDDClk3PreScale;
      DClkPostScale = RDDClk3PostScale;
}
```

These are used to control the output frequency according to the following formula :-

Output Frequency = (Frequency of reference clock * ClkFeedbackScale) / (ClkPreScale * (1 << ClkPostScale))

The first two sets of DClk control registers are configured at (hard) reset to generate 25.057MHz (Set 0), and 28.278 MHz (Set 1). These frequencies assume that the external reference clock is 14.31818MHz. The second pair of DClk control registers are un-initialised at reset.

INDEX