Overview

2. OVERVIEW

The CL-GD546X family of high-performance VisualMedia™ accelerators, based on the latest Rambus® technology, offers VRAM performance at DRAM cost. The first member, the CL-GD5462, includes a 64-bit GUI accelerator engine with multiple window video acceleration, an integrated palette DAC, clock synthesizers, and a glueless PCI/VESA® interface. Also included is the Enhanced V-Port™ that allows a GUI accelerator based on the CL-GD546X to be easily expanded to include multimedia functions. The second member, the CL-GD5464, includes a 3D texture mapping and rendering engine, display list instruction processing, and related features to bring realtime animation of realistic images to PC-class machines.

2.1 Features

Table 2-1 summarizes the major features of each member of the CL-GD546X family of VisualMedia accelerators.

Table 2-1. CL-GD546X Features List

Features	CL-GD5462	CL-GD5464
Rambus® display memory (Mbytes)	1 to 8	1 to 8
Rambus® channels	1	1
Maximum resolution (8 bpp, non-interlaced)	1600 × 1200	1600 × 1200
Maximum true-color resolution (24 bpp, non-interlaced)	1024 × 768	1024 × 768
GUI acceleration width (in bits)	64	64
Three-operand BitBLT	✔	✔
Color expansion for 8-, 16-, 24-, 32-bpp modes	✔	✔
Linedraw acceleration	✔	✔
Integrated triple 8-bit DAC	✔	✔
Hardware cursor	✔	✔
Programmable dual-clock synthesizer	✔	✔
General-purpose I/O port	✔	✔
Video playback acceleration	✔	✔
X,Y interpolated scaling	✔	✔
Simultaneous occluded video screens	3	3
Mixed graphic/video formats in frame buffer (9-bit RDRAMs)	✔	✔
Color key support	✔	✔
YCrCb and AccuPak™ support	✔	✔
24-bit pixel bus (video playback width)	✔	✔
Maximum pixel clock	170 MHz	230 MHz
Maximum memory clock	264 MHz	264 MHz
‘Green PC’ power-saving features	✔	✔

Table 2-1. CL-GD546X Features List *(cont.)*

Features	CL-GD5462	CL-GD5464
Direct PCI bus interface (v2.1) with zero-wait-state burst	✔	✔
Low-power CMOS, 208-pin PQFP package	✔	✔
Perspective, filtered, area-based texture mapping		✔
Internal texture cache and support for 4-, 8-, and 16-bpp texture maps		✔
4- and 8-bit indexed texture source to 16- and 24-bpp display modes		✔
Support for M × N power of two source maps up to 512 × 512 texels		✔
Texture map source from system or RDRAM memory		✔
Lighted and shaded textures with Gouraud ramp and transparent texture support		✔
Z-buffering compare depth >, ≥, <, ≤, = to source depth, source-to-mask, and set status event on depth compare		✔
Z storage and retrieval from either system or RDRAM memory		✔
PCI Bus Master mode for 2D/3D display list instruction fetch (Processor mode) and data fetch and store to system memory		✔
Command and data prefetching from system memory		✔
Display list (Processor mode) instruction fetch uses batch mode execution (runs in parallel with system CPU subsystem)		✔
Hardware paging scheme for operating systems which allocate fragmented memory spaces for Instruction, texture, depth, and color data		✔
Display list screen event synchronization at line resolution		✔
Color compare events for display list instruction context switching		✔
Double and multiple buffering support through instruction synchronization		✔
Polygon engine for XYZ DDA interpolation of randomly orientated triangles		✔
Support for Gouraud shading in 8-, 16-, and 24-bpp display modes		✔
Point and line draw support via polygon engine DDA’s		✔
Hardware dithering for 8-bpp monochrome and 3:3:2, and 16-bpp 5:6:5 modes		✔
Floor ± 1 correction for XOR alignment of triangle edges		✔
Color saturate and mask to high/low bounds		✔
YUV-to-RGB conversion in stretch engine path, supports MPEG textures		✔
Color keying for substitution of video on graphics		✔
Chroma keying for substitution of graphics on video		✔

2.2 Chip Architecture

2.2.1 Introduction

The CL-GD546X family integrates the logic necessary for a flexible multimedia display system. This includes a 64-bit 2D/3D graphics engine, integrated palette DAC, Rambus® channel interface, and support logic described in the following sections. Figure 2-1 shows the glueless interfaces to the PCI bus, the Rambus display memory, the Enhanced V-Port™, and the monitor.

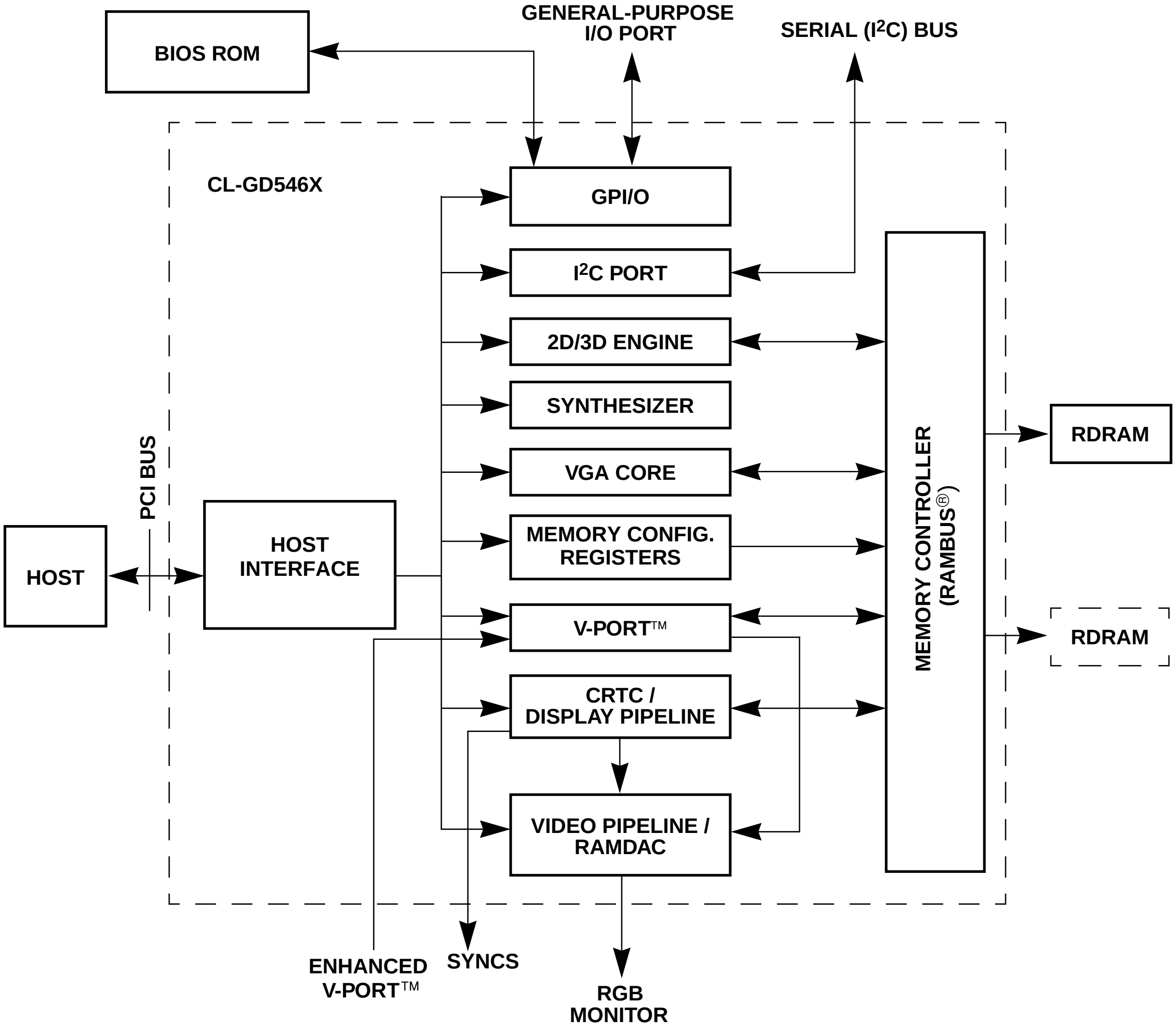

Figure 2-1. CL-GD546X Block Diagram

The following sections describe the functional blocks that are integrated into the CL-GD546X. There are conceptual block diagrams, not necessarily indicative of the actual implementation, with most of the descriptions.

2.2.2 Host Interface

The CL-GD546X has a glueless interface that connects directly to any PCI host bus. Since all 32 address bits are available, no external address decoders are required and no address aliasing occurs. Since all 32-data bits are available, no external buffers are required.

The CL-GD5464 host interface is PCI v2.1-compliant and supports zero-wait-state bursts at up to 33 MHz as a bus slave and bus master. The host interface meets the PCI single electrical load specification by buffering the address and data for the BIOS ROM. The CL-GD546X can provide the interface for an additional peripheral attached to the general-purpose I/O port while still meeting the single-load specification.

Figure 2-2 on page 2-6 presents the functional units of the host interface. The command queue, read queue, bi-endian swap logic (for writes and reads), address decoder, and PCI control (target) units provide PCI-compliant target transfers of data to or from the CL-GD5464. The Prefetch unit and the PCI control (master) unit use PCI bus master cycles to fetch display list instructions and parameters from host system memory. Similarly, the HostXY and PCI control (master) units allow reads of texture maps stored in host memory and rendering (reads and writes) of a color buffer and Z buffer to host memory.

The 8 × 43 command queue allows the CL-GD5464 as a target to release the host as soon as the transaction parameters have been recorded. This enables the host and media accelerator to operate with a high degree of parallelism.

The host address bus (or the address phase of the multiplexed AD bus) enters the address decoder where the CL-GD546X determines if it is the target of the transaction about to occur. If it is the target of the transaction, the appropriate acknowledge lines are activated by the bus control block and the address is placed in the command queue along with a tag value that indicates the transaction type.

Entries are removed from the command queue and passed on to the appropriate internal block for execution. If the command queue is full, the bus control unit inserts wait cycles until one or more free entries are available.

Read transactions must be executed by CL-GD546X before the host can be released (since the data must be made available to the host). Generally, this requires a number of wait states. For BIOS reads, up to 4 bytes are assembled into the read queue before the data is placed on the data bus and the host is released.

The prefetch unit is responsible for the fetch and pre-decode of display list instructions from host memory. Rendering instructions are forwarded to the 2D/3D engine by the command queue, whereas control instructions are executed immediately by the prefetch unit. To achieve maximum animation performance, internal interrupt or wait events, such as a display buffer switch or a VSYNC are handled by this unit.

The HostXY unit translates texture or pixel transactions from the 2D/3D engine into PCI bus master transactions to the host memory, and checks that the addresses are in a valid range. Pixel or Z data writes are queued in the read queue for optimal performance.

Both the prefetch and HostXY units contain virtual-to-physical address translation logic that reads a translation table from the host memory.

Bi-endian swaps (dword or word) can occur (as needed) on data passing in either direction through the host interface.

The general-purpose I/O port is closely integrated with the host interface. If CL-GD546X is configured for general-purpose I/O port, accesses to a specific range of memory-mapped I/O offsets are converted into accesses to the local peripheral.

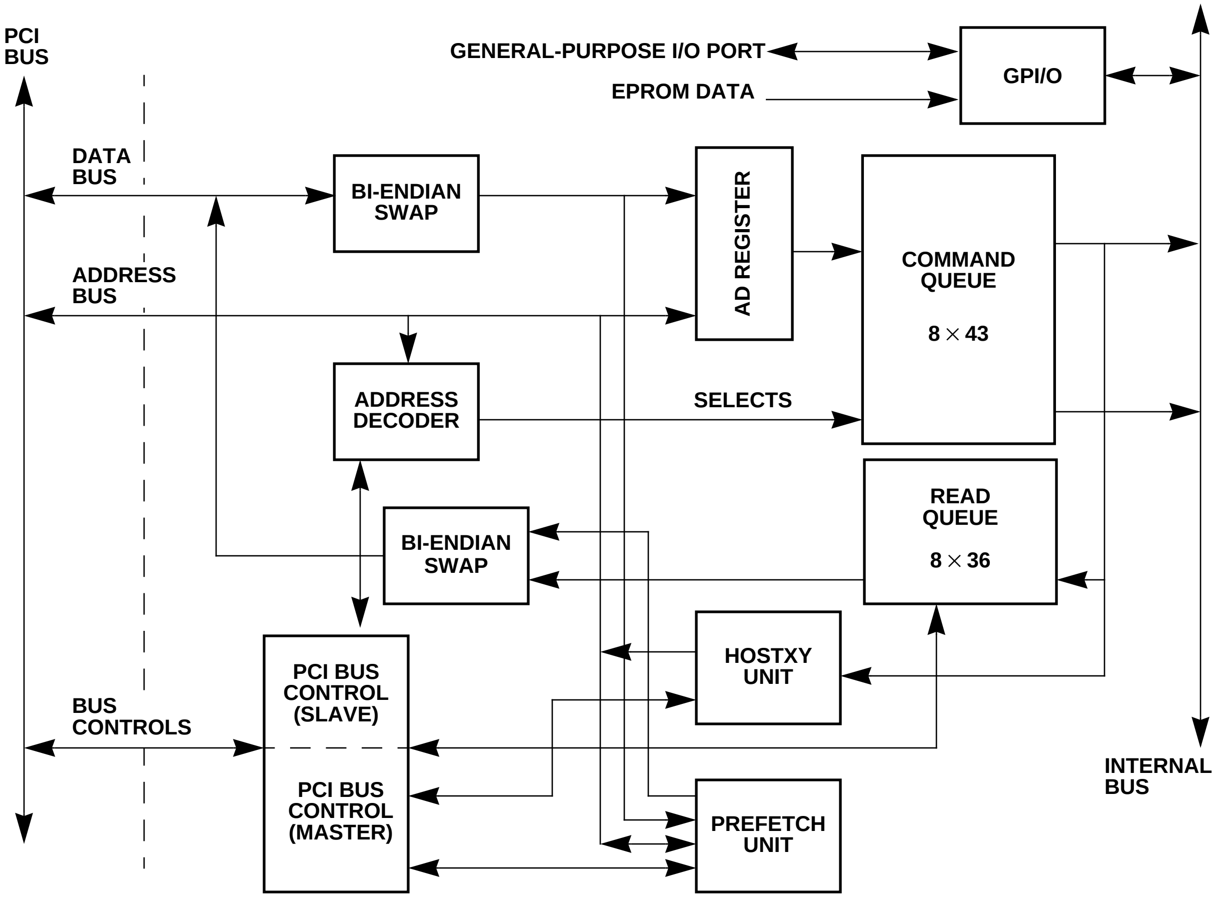

Figure 2-2. Host Interface Block Diagram

2.2.3 2D/3D Graphics Engine

The CL-GD546X 2D/3D graphics engine includes an advanced 64-bit three-operand BitBLT engine with stretch and shrink logic. The engine requires that parameters be loaded into various registers to define the operation. The operation is actually started by loading the appropriate Extent register. There is a 24-entry command buffer FIFO used by the interface to minimize register access time. In addition, all registers are double buffered.

Figure 2-3 on page 2-8 presents a functional block diagram of the CL-GD546X 2D/3D graphics engine.

The 2D engine is built around three 64-bit-wide scratch pad memories labeled SRAM2, SRAM1, and SRAM0. Each SRAM cell has separate read and write ports. Data can be written into each SRAM from display memory and from the output of the pixel path (the result data). The SRAMs allow the pixel path logic to pipeline operations by acting as data reservoirs.

Data from the SRAMs can be manipulated in a number of ways in the data flow according to the details of the operation. It can be converted from monochrome to color, aligned, and combined in any of the 256 possible ROPs. If a stretch BitBLT is occurring, pixels are generated by replication or interpolation. If a shrink BitBLT is occurring, pixels are discarded or averaged.

The outputs from the pixel path are buffered in the output FIFO, and can be directed to the frame buffer, host, or back to the top of data flow as necessary.

The 3D graphics engine incorporated into the CL-GD5464 can draw randomly oriented triangles with Gouraud shading, texture mapping, alpha blending, and Z-buffering. The CL-GD5464 has support for copy, decal, blended 2D/3D, and modulated textures as well as bilinear, bilinear mip mapped, and trilinear textures.

The 3D graphics engine typically operates in Display List mode, fetching instructions and parameters from system memory that it executes autonomously. When the 3D engine reaches the end of the display list, it generates an interrupt to the host, which can then start the engine on another display list.

The display lists consist of the drawing instructions (DRAW_POINT, DRAW_LINE, and DRAW_POLYGON) and their parameters, as well as control instructions such as BRANCH, C_BRANCH, TEST, WAIT, and INTerrupt.

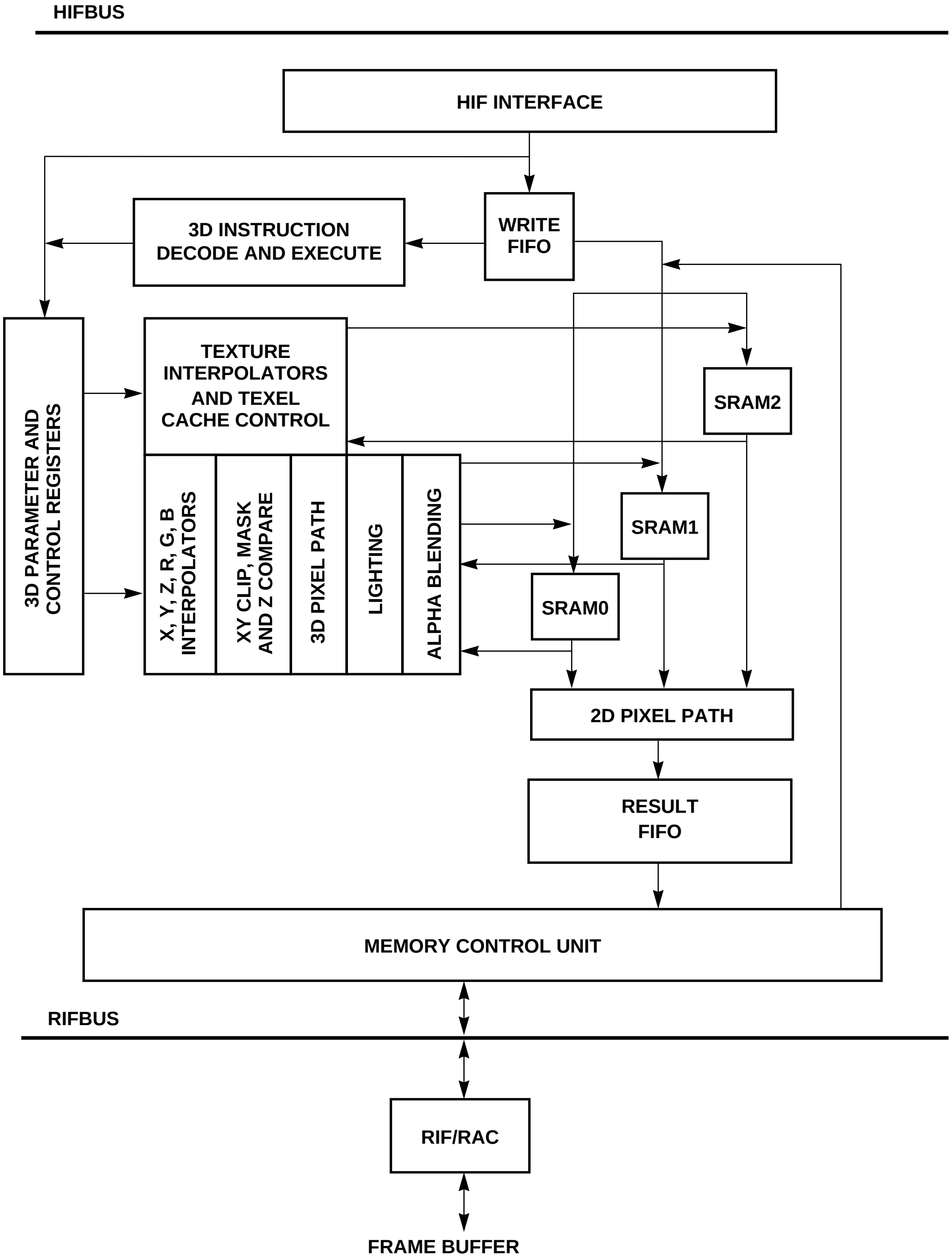
HIFBUS
HIF INTERFACE
WRITE FIFO
3D INSTRUCTION DECODE AND EXECUTE
3D PARAMETER AND CONTROL REGISTERS
TEXTURE INTERPOLATORS AND TEXEL CACHE CONTROL
X, Y, Z, R, G, B INTERPOLATORS
XY CLIP, MASK AND Z COMPARE
3D PIXEL PATH
LIGHTING
ALPHA BLENDING
SRAM2
SRAM1
SRAM0
2D PIXEL PATH
RESULT FIFO
MEMORY CONTROL UNIT
RIFBUS
RIF/RAC
FRAME BUFFER

Figure 2-3. 2D/3D Graphics Engine Block Diagram

2.2.4 Rambus® Interface

The CL-GD546X connects directly to one or two Rambus channels (the CL-GD5462 and CL-GD5464 connect to a single channel), each capable of supplying burst data at up to 528 Mbytes per second. The block diagram in Figure 2-4 shows a single Rambus interface.

Requests to access the Rambus interface come from each of the six modules (including memory refresh requests). These go into an arbitration unit that determines which requestor executes the next transaction.

Once the request is granted, the Rambus state machine tests whether the row being accessed is currently in the row cache. In this case, the transaction occurs more quickly than otherwise. In either case, the request is executed, transferring up to 256 bytes. Then the arbitration unit determines which requestor is next.

The Rambus access cell actually connects to the Rambus control and data lines. The I/O cells in this module use the special Rambus low-swing logic levels.

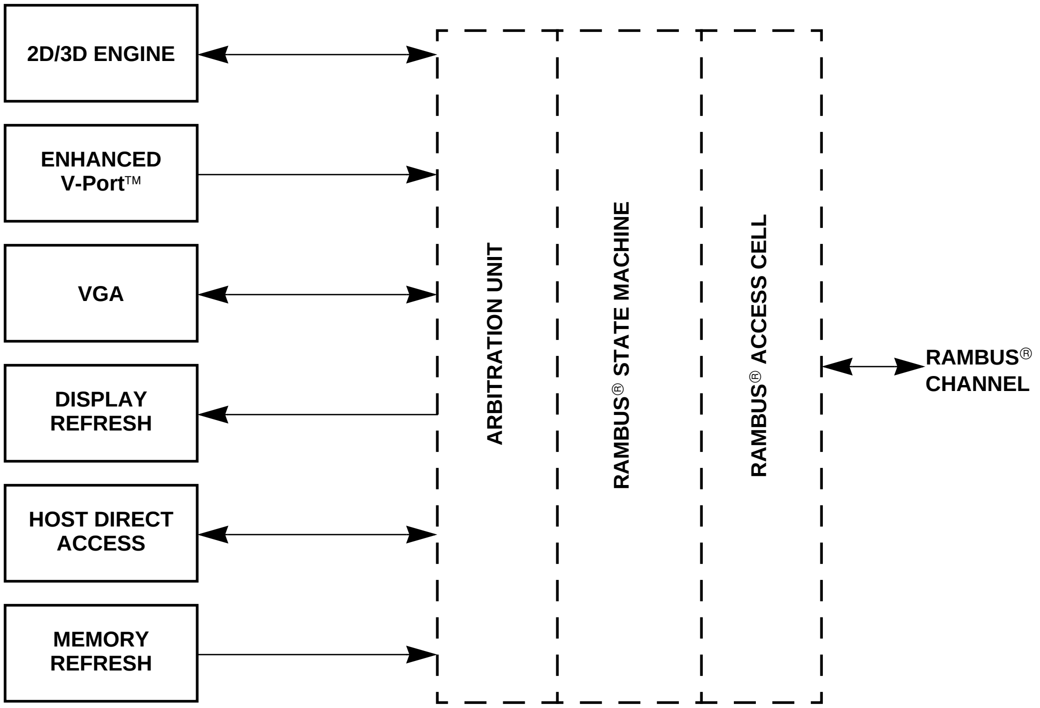

Figure 2-4. Rambus® Interface

2.2.5 VGA Core and Extended I/O

The CL-GD546X includes a VGA core and extended I/O logic that is compatible with the CL-GD542X VGA and CL-GD543X/'4X VGA families. The VGA core is also compatible with the industry-standard IBM VGA adapter.

Figure 2-5 presents a functional block diagram of the VGA core and extended I/O logic. The VGA core contains the VGA-compatible access path to the frame buffer. This includes the VGA Sequencer and VGA Graphics Controller registers. Additional information on these blocks can be found in the *CL-GD542X True Color VGA Family Technical Reference Manual*. The descriptions of these registers can be found in Chapter 3, "VGA Core Registers", of this manual.

The Cirrus Logic Extended registers are contained in the extended I/O. These registers are described in Chapter 4, "Extended I/O Registers".

The extended I/O block allows the CPU to write or read anywhere in the frame buffer, independently of the BitBLT engine.

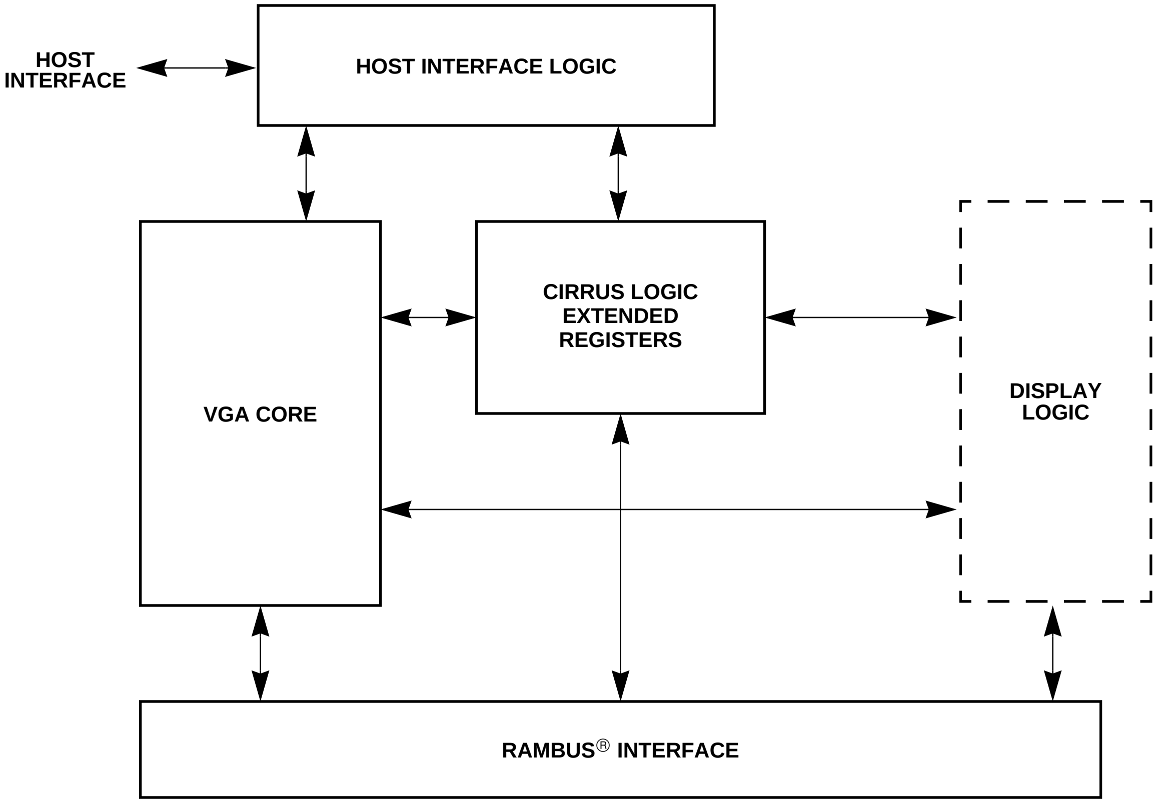

Figure 2-5. VGA and Direct Frame Buffer Access

2.2.6 Display Pipeline and Display FIFO

The CL-GD546X contains a VGA display pipeline that corresponds approximately to the logic driven by the VGA Attribute Controller registers. This subset of the CL-GD546X display path is compatible with the IBM VGA standard and is used for the standard video modes 0–13h.

The graphics pipeline is used in Extended modes, and supports color space interpolation and conversion, color expansion, multiple hardware windows, and pixel replication and interpolation.

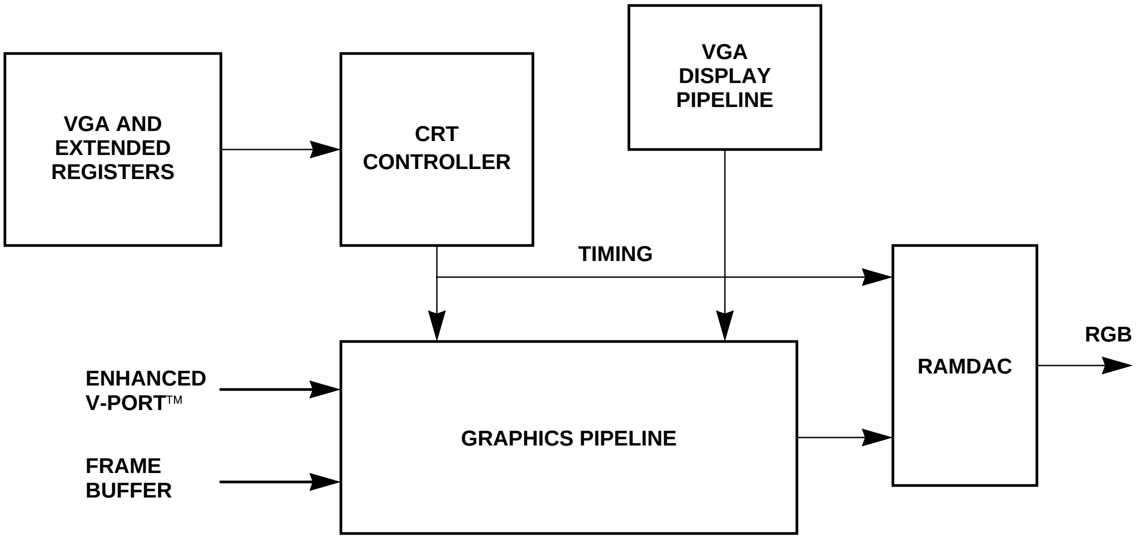

Figure 2-6. Display Path and Display FIFO

2.2.7 Enhanced V-Port™

The enhanced V-Port provides for capture and display of realtime digitized video. In addition to supporting the industry-standard VGA feature connector, the enhanced V-Port also supports capture and display of video from a TV decoder or an MPEG decoder.

The enhanced V-Port captures video in off-screen bufffers and then moves it on-screen, with scaling, synchronous with CRT refresh. Once initiated, this occurs, optionally with double buffering, with a minimum of host intervention. The I^{2}C serial interface provides a method of controlling a video decoder from the host.

2.2.8 Palette DAC

The CL-GD546X palette DAC consists of a very fast SRAM and three 8-bit DACs. The SRAM consists of three individually addressable arrays, each with 256 plus 16 eight-bit entries. Each DAC is made up of 256 current sources; the input to each DAC determines how many current sources are turned on for each pixel.

The SRAM is used for 8-bpp palettized modes (for example, 1024 × 768 × 256 colors) and for other modes when gamma correction is enabled. In 8-bpp palettized modes, the pixel value is used to access one of the 256-color triplets in the SRAM. Each value in the color triplet is directed to one of the three DACs: Red, Green, and Blue. Since the values were stored as 6-bit values, they are left-shifted two bit positions to allow full-scale outputs from the 8-bit DACs.

In modes with gamma correction, each of the three color values is used to access one of the 256 entries in each of three sections of the SRAM. The values in these three entries are directed to the three DACs.

For all modes other than 8-bpp palettized and modes with gamma correction, the pixel value itself is directed to the three DACs. The number of bits directed to each DAC depends on the mode, and can vary from five (16-bpp RGB) to eight (24-bpp RGB). In each of these modes, the bits are scaled so that they drive the most-significant bit positions of the respective DACs.

The current reference for the DACs is integrated and requires only an external resistor.

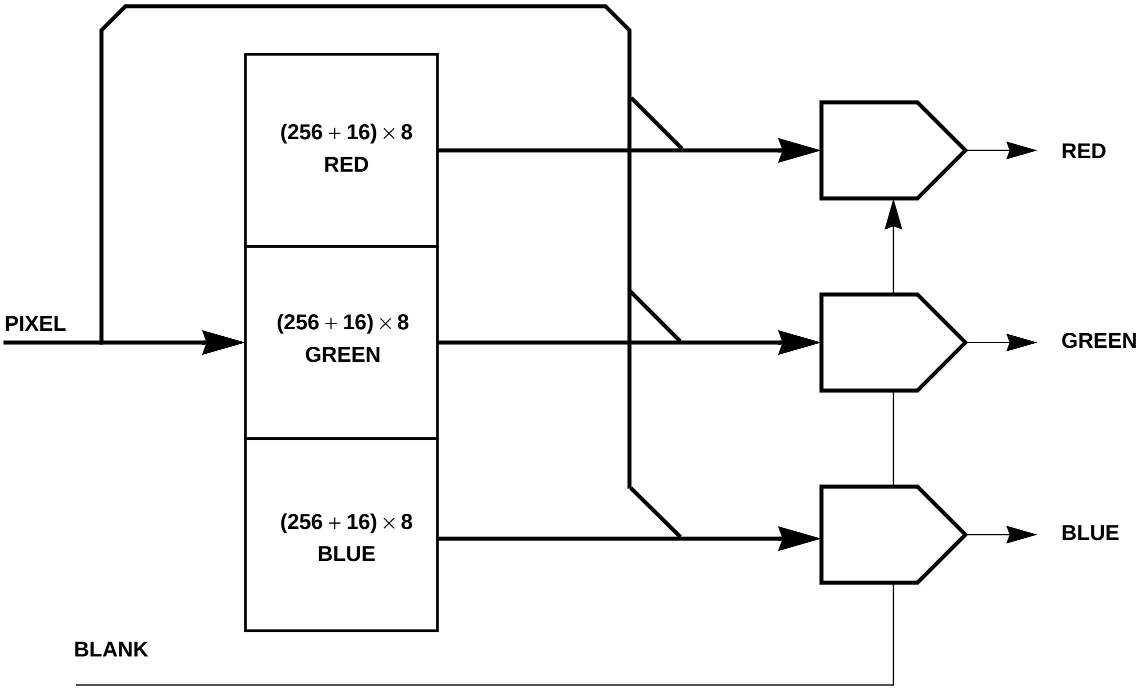

Figure 2-7. Palette DAC

2.2.9 Programmable Synthesizers

There are two frequency synthesizers on the CL-GD546X; one for the pixel (video) clock and the CRTC timing, and one for Rambus. Each synthesizer has a PLL to multiply the 14.31818-MHz reference to the appropriate frequency.

The pixel clock is programmed to a frequency appropriate for the current video mode; this can be as low as 12.5 MHz or as high as 230 MHz. For very high pixel rates, the synthesizer is programmed to one half of the pixel rate and a final frequency doubling occurs in the palette DAC.

The Rambus clock synthesizer is typically programmed to 258 MHz.

2.2.10 CRT Controller

The CRTC is compatible with the IBM VGA CRTC (and has additional most-significant bits for some of the values). A functional block diagram of the CRTC is shown in Figure 2-8.

The horizontal timing is generated by counting a character clock that is typically one-eighth of the pixel clock. That is, the horizontal timing generator typically counts once for every eight pixels. The horizontal timing generator makes horizontal sync and horizontal blank timing.

The vertical timing is generated by counting scanlines in the form of horizontal sync pulses. The vertical timing generator makes vertical sync and vertical blank timing. Vertical blank timing is logically OR'ed with horizontal blank timing to produce a composite blank.

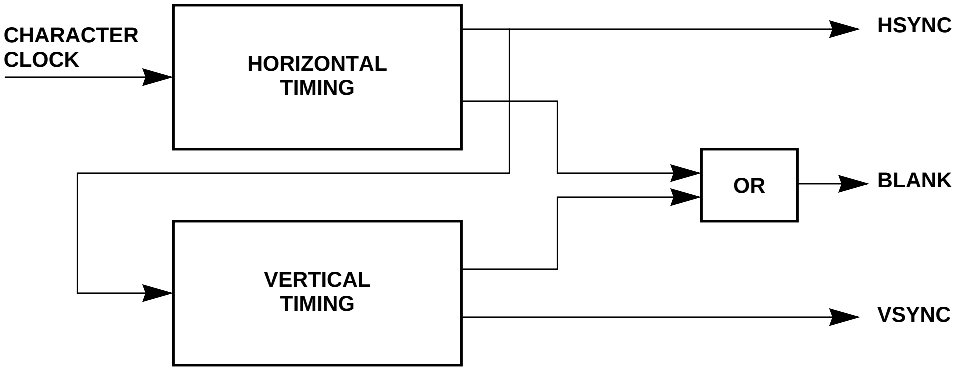

Figure 2-8. CRT Controller

2.2.11 I²C (Serial) Port

The I^2C (serial) port consists of two software-controlled, open-collector outputs that can be controlled with register bits and whose levels can be sensed. These pins, together with the controlling software, implement an I^2C interface for DDC 2B. This interface can also be used to control devices such as a television tuner.

2.2.12 General-Purpose I/O Port (PCI Configuration)

The pins that are used to access the BIOS ROM can be reused as a general-purpose I/O port after the BIOS ROM has been read and stored in system memory.

Three configuration pins determine the GPI/O port operating mode: none, 8-bit Intel® configuration, 16-bit Intel configuration, ATT AV4400A video codec, or C-CUBE CL480 MPEG decoder.

The hardware interface is shown in Figure 2-9. The details of the pins depend on the mode that is configured. There can be up to 6-address bits, 6-control bits, and 16-bidirectional data bits. There is always a single-handshake line that the local peripheral can use to stretch the access cycle.

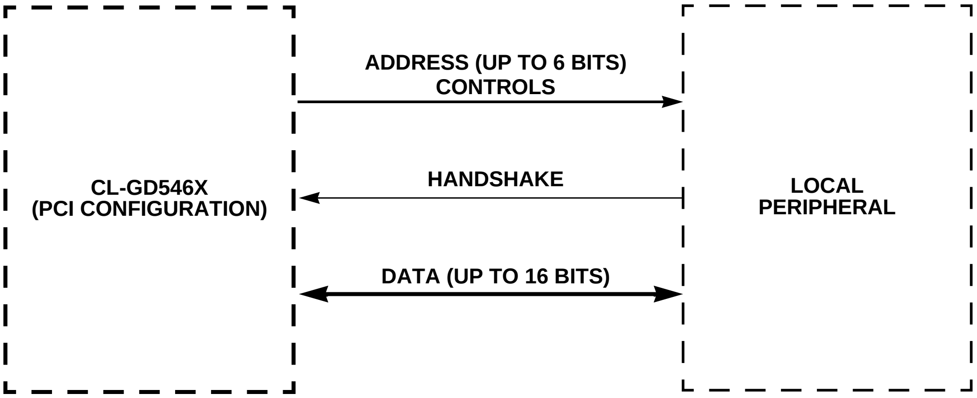

Figure 2-9. General-Purpose I/O Port

2.3 Hardware/Software Compatibility

Compatibility is a key item in the architecture of the CL-GD546X. On the hardware side, the CL-GD546X provides truly glueless interfaces to PCI, one or two Rambus channels, industry-standard monitors, and a VESA standard feature connector interface.

On the software side, the CL-GD546X is software compatible with the IBM VGA standard, and register compatible with industry-standard VGA. Cirrus Logic provides a VGA-compatible BIOS with VESA extensions, as well as drivers for the many industry-standard GUI interfaces (such as, Microsoft® Windows® v3.x, Windows® 95, Microsoft NT, Microsoft DirectDraw™, and IBM® OS/2®).

2.4 Graphics Subsystem Architecture

Figure 2-10 shows the main components required to implement a functional graphics subsystem using the CL-GD5464. The CL-GD5464 is combined with RDRAM and connected to a PCI bus and a monitor to provide a small, cost-effective video subsystem. When an adapter card is built around the CL-GD546X, a BIOS ROM is required. When the video capture capability of the CL-GD546X is used, the V-Port bus connections must be made.

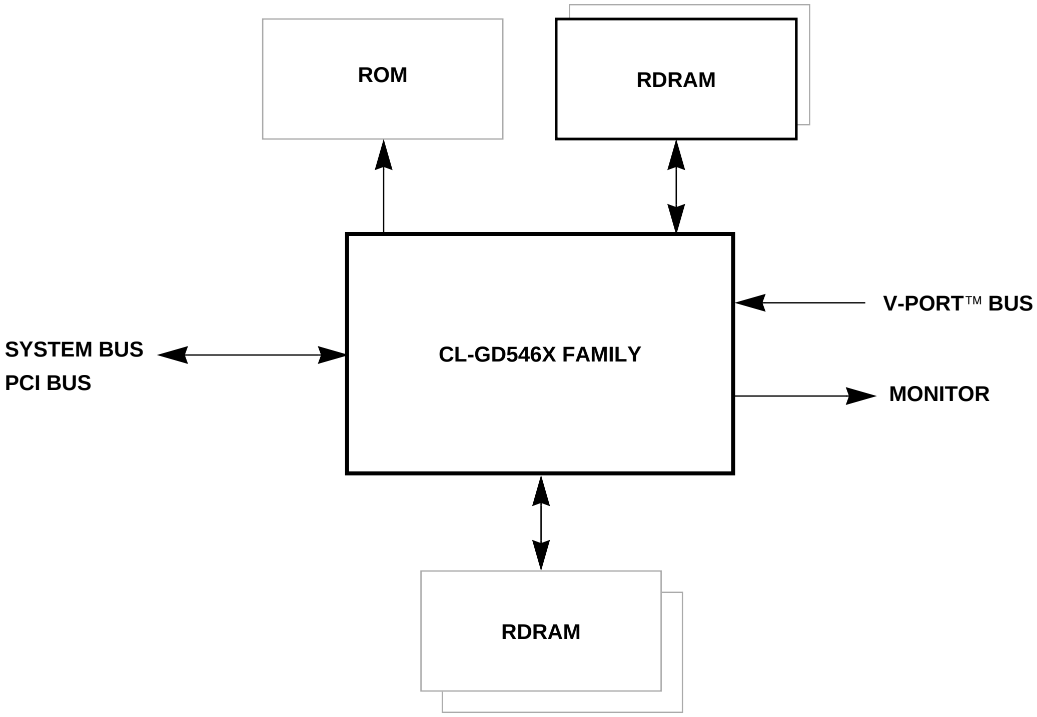

Figure 2-10. Video Subsystem Architecture