2

Functional Description

Section 2: Functional Description

2.1 Overview

This section describes the architecture and block diagram for ~~the IMAGINE 128~~. SilverHammer. ~~The IMAGINE 128~~SilverHammer provides a high performance, AGP and PCI compliant interface with no additional external logic required.

Software may interact with ~~IMAGINE 128~~SilverHammer by directly manipulating pixels through the Memory Windows interface or by setting up parameters in the register area to trigger one of the ~~IMAGINE 128~~SilverHammer drawing engine commands.

The **Drawing Engine** commands provide all of the normally required operations including: BIT BLT, 2D Line, 3D Line with setup, Triangle with full setup and vertex sorting, Write Image, and Read Image.

~~The IMAGINE 128~~SilverHammer is implemented using a highly pipelined graphic processor architecture. This architecture allows for high performance 2D and 3D Rendering. After a sequence of commands and parameters are written, ~~IMAGINE 128~~SilverHammer executes the selected command without any further processor intervention.

2.2 Addressing Scheme

~~The IMAGINE 128~~ SilverHammer supports two local buffers, each of which may be partitioned in up to 16 logical buffers. These buffers may be used as displayed below:

LOGICAL FUNCTION	VRAM BUFFER	DRAM BUFFER	SGRAM	WRAM
Display	YES	YES	YES	YES
Virtual	YES	YES	YES	YES
Z Buffer	YES	YES	YES	YES

These buffers may be accessed as linear buffers through the Memory windows interface or through the drawing engine. The Virtual and Display buffers share a common address and data bus which is ~~either 64 or~~ 128 bits~~, depending on configuration~~.

2.2.1 Display Buffer

The Display Buffer may be constructed of EDO VRAM, EDO DRAM, SGRAM or WRAM technology, which supports write-per-bit operations. If non-write per bit memory is used no planar masking is supported. The Display Buffer may be accessed directly in linear address mode, or through the drawing commands.

Writes to the Display Buffer may be shadowed to the virtual buffer, in addition writes to the virtual buffer may be shadowed to the display buffer. The Display Buffer may occupy up to sixteen megabytes of system space.

Although the main purpose of the Display buffer is to store the image data that is displayed on the CRT, any off screen memory may be used for the storage and manipulation of bitmaps, texture maps, Z buffering or fonts.

2.2.2 Virtual Buffer

The Virtual Buffer is an optional buffer and may be constructed with EDO VRAM, EDO DRAM, SGRAM, or WRAM which supports write per bit operation. If non-write per bit memory is used no planar masking will be supported. As with the Display Buffer this buffer may be accessed with linear addressing through the Memory windows interface or through drawing engine commands.

Writes to the display buffer may be shadowed to the Virtual Buffer, or writes to the Virtual Buffer may be shadowed to the display buffer. The Virtual Buffer may occupy up to thirty-two megabytes of system space. The Virtual Buffer's main purpose is to store large bitmaps, texture maps or large Z maps. Portions of these bitmaps may then be transferred to the Display Buffer using one of the drawing engine's bitbit commands.

2.3 Register Map

This section provides a brief overview of all ~~IMAGINE 128~~SilverHammer registers. A full description may be found in **Sections 4** and **5** of this document.

2.3.1 I/O Mapped VGA DAC Registers

The following is a list of the I/O mapped VGA DAC shadowing registers. The address of these registers are absolute I/O address.

~~IMAGINE 128~~SilverHammer can easily be configured to respond or not respond to the following four I/O addresses. There are two modes of operation If VGA DAC register access is enabled in the VGA-CTRL register 9 (*See section)* "snooping" and "owning". These modes are described in subsequent sections.

I/O ADDRESS	REGISTER NAME	DESCRIPTION
0x03C6	PEL_MASK	Pixel Mask
0x03C7	RD_ADR	Read Address
0x03C8	WR_ADR	Write Address
0x03C9	PAL_DAT	Palette Data

2.3.2 I/O Mapped Configuration Registers

The following is a list of the I/O mapped configuration registers. The base I/O address of these registers is set via PCI base register 5 at boot time.

I/O ADDRESS	REGISTER NAME	DESCRIPTION
0x0000	RBASE_G	Base address for the global register
0x0004	RBASE_W	Base address for Mem windows config registers.
0x0008	RBASE_D	Base address for Drawing Engine Register Set
0x000C	Reserved	Reserved
0x0010	RBASE_I	Base address for the Global Interrupt Registers
0x0014	RBASE_E	Base address for the local PROM
0x0018	ID	Chip ID register
0x001C	CONFIG1	Chip Configuration Register 1
0x0020	CONFIG2	Chip Configuration Register 2
0x0024	SGRAM	SGRAM Configuration Register
0x0028	SSWTCH	Soft Switch Register
0x002C	DDC	DDC1 /DDC2B Register
0x0030	VGA_CTRL	VGA Control Register
0x0034 - 0x003C	Reserved	Reserved
0x0040	MW1_CTRL	Memory Window 1 control (shadowed)
0x0044	MW1_AD	Memory Window 1 Address (shadowed)
0x0048	MW1_SZ	Memory Window 1 Size (shadowed)

I/O ADDRESS	REGISTER NAME	DESCRIPTION
0x004C	MW1_PGE	Memory Window 1 Page (shadowed)
0x0050	MW1_ORG	Memory Window 1 Origin (shadowed)
0x0054	MW1_ORG	Memory Window 1 Origin (shadowed)
0x0058	MW1_MSRC	Memory Window 1 Mask source (shadowed)
0x005C	MW1_WKEY	Memory Window 1 Color Key (shadowed)
0x0060	MW1_KYDAT	Memory Window 1 Key Data (shadowed)
0x0064	MW1_MASK	Memory Window 1 Plane Mask (shadowed)
0x0068	BIOS1	General Purpose BIOS Register 1
0x006C	BIOS2	General Purpose BIOS Register 2
0x0070	BIOS3	General Purpose BIOS Register 3
0x0074	BIOS4	General Purpose BIOS Register 4
0x0078	Reserved	Reserved
0x007C	Reserved	Reserved
0x0080	WR_ADDR	Palette/Cursor RAM, Write Address Register
0x0084	PAL_DAT	DAC Palette data Register
0x0088	PEL_MASK	DAC Pixel Mask Register
0x008C	RD_ADDR	Palette/Cursor RAM, Read Address Register
0x0090	WRC_ADDR	Cursor/Overscan Color, Write Address Register
0x0094	CUR_DAT	Cursor/Overscan Color data
0x0098	DAC_CR0	DAC Control Register 1
0x009C	RDC_ADR	Cursor/Overscan Color, Read Address Register
0x00A0	DAC_CR1	DAC Control Register 1
0x00A4	DAC_CR2	DAC Control Register 2
0x00A8	DAC_SR	DAC Status Register (Read Only)
0x00AC	CUR_RDAT	Cursor RAM Array Data
0x00BO	CUR_XL	Cursor X Position Low Byte
0x00B4	CUR_XH	Cursor X Position High Byte
0x00B8	CUR_YL	Cursor Y Position Low Byte
0x00BC	CUR_YH	Cursor Y Position High Byte
0x00D0	DMA_SRC	DMA Address Source Register
0x00D4	DMA_DST	DMA Address Destination Register
0x00D8	DMA_CNT	DMA Count Host Register
0x00DC	DMA_CMD	DMA PCI Command Register
0x00E0	DMA_TRG	DMA Trigger

2.3.3 Memory Mapped Global Registers

The following is a list of the memory mapped global registers. The base address of these registers is offset by the value loaded in the RBASE_G register in Memory space. The DAC registers, 0x0000-0x001C, should only be accessed in non-burst mode.

I/O ADDRESS	REGISTER NAME	DESCRIPTION
0x0000	WR_ADR	DAC Write address
0x0004	PAL_DAT	DAC Palette data
0x0008	PEL_MASK	DAC Pixel Mask reg.
0x000C	RD_ADR	DAC Read address
0x0010	RESERVED	RESERVED
0x0014	RESERVED	RESERVED
0x0018	INDEX_TI	DAC index Reg (Viewpoint)
0x001C	DATA_TI	DAC data Reg (Viewpoint)
0x0020	INT_VCNT	Vertical interrupt counter
0x0024	INT_HCNT	Horizontal interrupt counter
0x0028	DB_ADR	Display start address
0x002C	DB_PTCH	Display pitch
0x0030	CRT_HAC	Horizontal active line width
0x0034	CRT_HBL	Horizontal blank width
0x0038	CRT_HFP	Horizontal front porch width
0x003C	CRT_HS	Horizontal sync width
0x0040	CRT_VAC	Vertical active field width
0x0044	CRT_VBL	Vertical blank width
0x0048	CRT_VFP	Vertical front porch width
0x004C	CRT_VS	Vertical sync width
0x0050	CRT_LCNT	CRT Line Counter
0x0054	CRT_ZOOM	Display zoom factor
0x0058	CRT_1CON	CRT config. register 1.
0x005C	CRT_2CON	CRT config. register 2.
0x0070	WR_ADDR	Palette/Cursor RAM, Write Address Register
0x0074	PAL_DAT	DAC Palette data Register
0x0078	PEL_MASK	DAC Pixel Mask Register
0x007C	RD_ADDR	Palette/Cursor RAM, Read Address Register
0x0080	WRC_ADDR	Cursor/Overscan Color, Write Address Register
0x0084	CUR_DAT	Cursor/Overscan Color Data
0x0088	DAC_CR0	DAC Control Register 1
0x008C	RDC_ADR	Cursor/Overscan Color, Read Address Register
0x0090	DAC_CR1	DAC Control Register 1
0x0094	DAC_CR2	DAC Control Register 2
0x0098	DAC_SR	DAC Status Register (Read Only)
0x009C	CUR_RDAT	Cursor RAM Array Data
0x00A0	CUR_XL	Cursor X Position Low Byte
0x00A4	CUR_XH	Cursor X Position High Byte
0x00A8	CUR_YL	Cursor Y Position Low Byte
0x00AC	CUR_YH	Cursor Y Position High Byte

2.3.4 Memory Mapped Global Control Registers

The following is a list of the memory mapped global interrupt registers. The base address of these registers offset by the value loaded in the RBASE_I register in Memory space.

MEMORY ADDRESS	REGISTER NAME	DESCRIPTION
0x0000	GINTP	Global Interrupt Register.
0x0004	GINTM	Global Interrupt Mask Register.
0x0080	RBASE_G	Base address for the global register
0x0084	RBASE_W	Base address for the Memory windows config registers.
0x0088	RBASE_D	Base address for Drawing Engine register set
0x008C	Reserved	Reserved
0x0090	RBASE_I	Base address for the Global Interrupt registers
0x0094	RBASE_E	Base address for the local PROM
0x0098	ID	Chip ID register
0x009C	CONFIG1	Chip Configuration register 1
0x00A0	CONFIG2	Chip Configuration register 2
0x00A4	SGRAM	SGRAM Configuration Register
0x00A8	SSWTCH	Soft Switch register
0x00AC	DDC	DDC Register
0x00B0	VGA_CTRL	VGA Control Register
0x00B4	BIOS1	General Purpose BIOS Register 1
0x00B8	BIOS2	General Purpose BIOS Register 2
0x00BC	BIOS3	General Purpose BIOS Register 3
0x00C0	BIOS4	General Purpose BIOS Register 4
0x00D0	DMA_SRC	DMA Address Source Register
0x00D4	DMA_DST	DMA Address Destination Register
0x00D8	DMA_CNT	DMA Count Host Register
0x00DC	DMA_CMD	DMA PCI Command Register
0x00E0	DMA_TRG	DMA Trigger

2.3.5 Memory Windows Registers

The following is a list of the memory mapped Memory Windows™ registers. The base address of these registers offset by the value loaded in the RBASE_W register in Memory space.

MEMORY ADDRESS	REGISTER NAME	DESCRIPTION
0x0000	MW0_CTRL	Memory window 0 control
0x0004	MW0_AD	Memory window 0 address
0x0008	MW0_SZ	Memory window 0 size
0x000C	Reserved	Reserved
0x0010 or 0x0014	MW0_ORG	MW0 origin
0x0018	Reserved	Reserved
0x001C	Reserved	Reserved
0x0020	Reserved	Reserved
0x0024	MW0_MASK	MW0 Plane Mask
0x0028	MW1_CTRL	Memory window 1 control
0x002C	MW1_AD	Memory window 1 address
0x0030	MW1_SZ	Memory window 1 size
0x0034	Reserved	Reserved
0x0038 or 0x003C	MW1_ORG	MW1 origin
0x0040	Reserved	Reserved
0x0044	Reserved	Reserved
0x0048	Reserved	Reserved
0x004C	MW1_MASK	MW1 Plane Mask
0x0050	MWC_FCNT	Memory Window Cache Flush counter
0x0054	MWC_FLSH	Manual Cache Flush
0x0058	YUV_LI	YUV LUT index
0x005C	YUV_LA	YUV LUT address
0x0060	MW_CTRL	Memory Window 0 and 1 Control
0x0064 - 0x00FC	RESERVED	RESERVED

2.3.6 Memory Mapped Drawing Registers

The following is a list of the memory mapped registers. All of these registers are pipelined so that they are synchronous with the drawing engine. This register block is placed in system memory space by the setting of RBASE_D in the Memory mapped register block.

REGISTER ADDRESS	REGISTER NAME	DESCRIPTION
0x0000	INTP	Interrupt register
0x0004	INTM	Interrupt Mask register
0x0008	FLOW	Flow status register
0x000C	BUSY	Busy status bit
0x0010	XYW_ADSZ	XY Window Addr. and size.
0x0014	Reserved	Reserved
0x0018	Reserved	Reserved
0x001C	Reserved	Reserved
0x0020	BUF_CNTRL	Buffer enables
0x0024	Reserved	Reserved
0x0028	DE_SORG	Drawing Source origin
0x002C	DE_DORG	Drawing destination origin
0x0030	Reserved	Reserved
0x0034	Reserved	Reserved
0x0038	~~Reserved~~DE_TPTCH	~~Reserved~~Texture Pitch of LOD0
0x003C	DE_ZPTCH	Z buffer pitch
0x0040	DE_SPTCH	Source pitch
0x0044	DE_DPTCH	Destination pitch
0x0048	CMD	Command register
0x004C	Reserved	Reserved
0x0050	CMD_OPC	CMD opcode field
0x0054	CMD_ROP	CMD raster op
0x0058	CMD_STYLE	CMD line style
0x005C	CMD_PATRN	CMD line pattern control
0x0060	CMD_CLP	CMD clip control
0x0064	CMD_PF	CMD Pattern Fetch
0x0068	FORE	Foreground color register
0x006C	BACK	Background color register
0x0070	MASK	Plane Mask
0x0074	DE_KEY	Color Key
0x0078	LPAT	Line pattern register
0x007C	PCTRL	Line pattern control register
0x0080	CLPTL	Clip Rectangle Top Left Corner
0x0084	CLPBR	Clip Rectangle Bottom Right Corner
0x0088	XY0	XY0
0x008C	XY1	XY1
0x0090	XY2	XY2
0x0094	XY3	XY3
0x0098	XY4	XY4
0x009C	Reserved	Reserved
0x00A0	Reserved	Reserved

REGISTER ADDRESS	REGISTER NAME (I2/I3)	DESCRIPTION
0x00A4	Reserved	Reserved

0x00A8	Reserved	Reserved
0x00AC	Reserved	Reserved
0x00B0	Reserved	Reserved
0x00B4	Reserved	Reserved
0x00B8	Reserved	Reserved
0x00BC	Reserved	Reserved
0x00C0	Reserved	Reserved
0x00C4	Reserved	Reserved
0x00C8	Reserved	Reserved
0x00CC	Reserved	Reserved
0x00D0	~~Reserved~~LOD0	~~Reserved~~Level of detail 0 Origin
0x00D4	LOD1~~Reserved~~	Level of detail 1 Origin~~Reserved~~
0x00D8	LOD2~~Reserved~~	Level of detail 2 Origin~~Reserved~~
0x00DC	LOD3~~Reserved~~	Level of detail 3 Origin~~Reserved~~
0x00E0	LOD4~~Reserved~~	Level of detail 4 Origin~~Reserved~~
0x00E4	LOD5~~Reserved~~	Level of detail 5 Origin~~Reserved~~
0x00E8	LOD6~~Reserved~~	Level of detail 6 Origin~~Reserved~~
0x00EC	LOD7~~Reserved~~	Level of detail 7 Origin~~Reserved~~
0x00F0	LOD8~~Reserved~~	Level of detail 8 Origin~~Reserved~~
0x00F4	LOD9~~Reserved~~	Level of detail 9 Origin~~Reserved~~
0x00F8	DL_ADR	Display list address
0x00FC	DL_CNTRL	Display list control
0x0100	DE_ZORG	Z-buffer origin
0x0104	~~LOD0_ORG~~Reserved	Reserved~~Level 1 texture map origin~~
0x0108	Reserved~~LOD1_ORG~~	Reserved~~Level 2 texture map origin~~
0x010C	Reserved~~LOD2_ORG~~	Reserved~~Level 3 texture map origin~~
0x0110	Reserved~~LOD3_ORG~~	Reserved~~Level 4 texture map origin~~
0x0114	Reserved~~LOD4_ORG~~	Reserved~~Level 5 texture map origin~~
0x0118	TPAL_ORG	Texture Palette origin
0x011C	HITH	Hither Register
0x0120	YON	Yon Register
0x0124	FOG_COL	Fog Color (Af,Rf,Gf,Bf)
0x0128	ALPHA	Alpha Register(Atest,Asrc,Adst)
0x012C	~~Reserved~~TEX_BORDER	~~Reserved~~Texture Boarder Color (RGB)
0x0130	~~Reserved~~V0_A_FP	~~Reserved~~Vertex0 Alpha Channel Floating point input
0x0134	~~Reserved~~V0_R_FP	~~Reserved~~Vertex0 Red Channel Floating Point Interface
0x0138	~~Reserved~~V0_G_FP	~~Reserved~~Vertex0 Green Channel Floating Point Interface
0x013C	~~Reserved~~V0_B_FP	~~Reserved~~Vertex0 Blue Channel Floating Point Interface
0x0140	~~Reserved~~V1_A_FP	~~Reserved~~Vertex1 Alpha Channel Floating Point Interface
0x0144	~~Reserved~~V1_R_FP	Vertex1 Red Channel Floating Point Interface~~Reserved~~

~~0x0148~~	~~Reserved~~	~~Reserved~~
~~0x014C~~	~~Reserved~~	~~Reserved~~

0x0150	Reserved	Reserved
0x0154	Reserved	Reserved
0x0158	Reserved	Reserved
0x015C	Reserved	Reserved
0x0160	Reserved	Reserved

REGISTER ADDRESS	REGISTER NAME (I2/I3)	DESCRIPTION
0x0148 0x0164	V1_G_FP Reserved	Vertex1 Green Channel Floating Point Interface Reserved
0x014C 0x0168	V1_B_FP CMD	Vertex1 Blue Channel Floating Point Interface Command Register
0x0150 0x016C	V2_A_FP A_CNTRL	Vertex2 Alpha Channel Floating Point Interface Alpha Control Register
0x0154 0x0170	V2_R_FP 3D_CNTRL	Vertex2 Red Channel Floating Point Interface 3D Control Register
0x0158 0x0174	V2_G_FP TEX_CNTRL	Vertex2 Green Channel Floating Point Interface Texture Control Register
0x015C 0x0178	V2_B_FP CP0	Vertex2 Blue Channel Floating Point Interface Command Parameter 0
0x0160 0x017C	KEY_3D_LOW CP1	3D Key value, low color Command Parameter 1
0x0164	KEY_3D_HI	3D Key value, High color
0x0168	CMD	Command Register
0x016C	A_CNTRL	Alpha Control Register
0x0170	3D_CNTRL	3D Control Register
0x0174	TEX_CNTRL	Texture Control Register
0x0178	CP0	Command Parameter 0
0x017C	CP1	Command Parameter 1
0x0180	CP2	Command Parameter 2
0x0184	CP3	Command Parameter 3
0x0188	CP4	Command Parameter 4
0x018C	CP5	Command Parameter 5
0x0190	CP6	Command Parameter 6
0x0194	CP7	Command Parameter 7
0x0198	CP8	Command Parameter 8
0x019C	CP9	Command Parameter 9
0x01A0	CP10	Command Parameter 10
0x01A4	CP11	Command Parameter 11
0x01A8	CP12	Command Parameter 12
0x01AC	CP13	Command Parameter 13
0x01B0	CP14	Command Parameter 14
0x01B4	CP15	Command Parameter 15
0x01B8	CP16	Command Parameter 16
0x01BC	CP17	Command Parameter 17
0x01C0	CP18	Command Parameter 18
0x01C4	CP19	Command Parameter 19
0x01C8	CP20	Command Parameter 20

0x01CC	CP21	Command Parameter 21
0x01D0	CP22	Command Parameter 22
0x01D4	CP23	Command Parameter 23
0x01D8	CP24	Command Parameter 24
0x01DC	3D_TRIG	Trigger Register for 3D
0x01E0	Reserved	Reserved
0x01E4	~~CP25~~Reserved	Reserved~~Command Parameter 25~~
0x01E8	Reserved~~CP26~~	Reserved~~Command Parameter 26~~
0x01EC	Reserved~~CP27~~	Reserved~~Command Parameter 27~~
0x01F0	Reserved~~CP28~~	Reserved~~Command Parameter 28~~
0x01F4	Reserved~~CP29~~	Reserved~~Command Parameter 29~~
0x01F8	Reserved~~CP30~~	Reserved~~Command Parameter 30~~
0x01FC	Reserved	Reserved

2.4 Diagram

The IMAGINE 128 is partitioned into five functional sections: They are the **Host Bus Interface**, the **Aperture Controller**, the **Drawing Engine**, the **Display/CRT Controller**, and the **Memory Controller**.
The **Host Bus Interface** provides an interface to the system bus. This is a master-slave interface. This means that the host or the IMAGINE DMA processor may initiate access to the IMAGINE 128. The Host Bus Interface operates as a 32-bit multiplexed address/data interface for the PCI local bus.

The **Aperture Controller** provides address decoding, address translation, color space conversion between the host interface and the local memory system. It also provides a mechanism for caching reads and writes from the host bus to the local buffers. In write mode, up to eight 32 bit words may be written to the host bus cache. The cache continuously monitors the address of each word written to determine if they are in the same page. If the words are not in the same page, or if the cache word count reaches eight, the cache will request the required number of memory writes from the Memory Controller. At this time the cache controller swaps access to its second cache and continues to accept host writes. If another page fault is detected during the secondary cache fill, a system stall will occur. This situation can be avoided by testing the cache and by doing cache line fills. During reads latency will be incurred for initial accesses or any page fault conditions. Software should make an effort to maintain scan line coherency during any access to the local buffers for optimal performance.

The **CRT Controller** provides programmable CRT timing signals: horizontal, vertical blanks and syncs. It is also responsible for generating requests to the memory controller for screen refresh cycles. VRAM read transfer cycles and split register cycles are supported. A free running frame counter which generates interrupts to the Host is also provided. This is useful for synchronizing bit map copies.

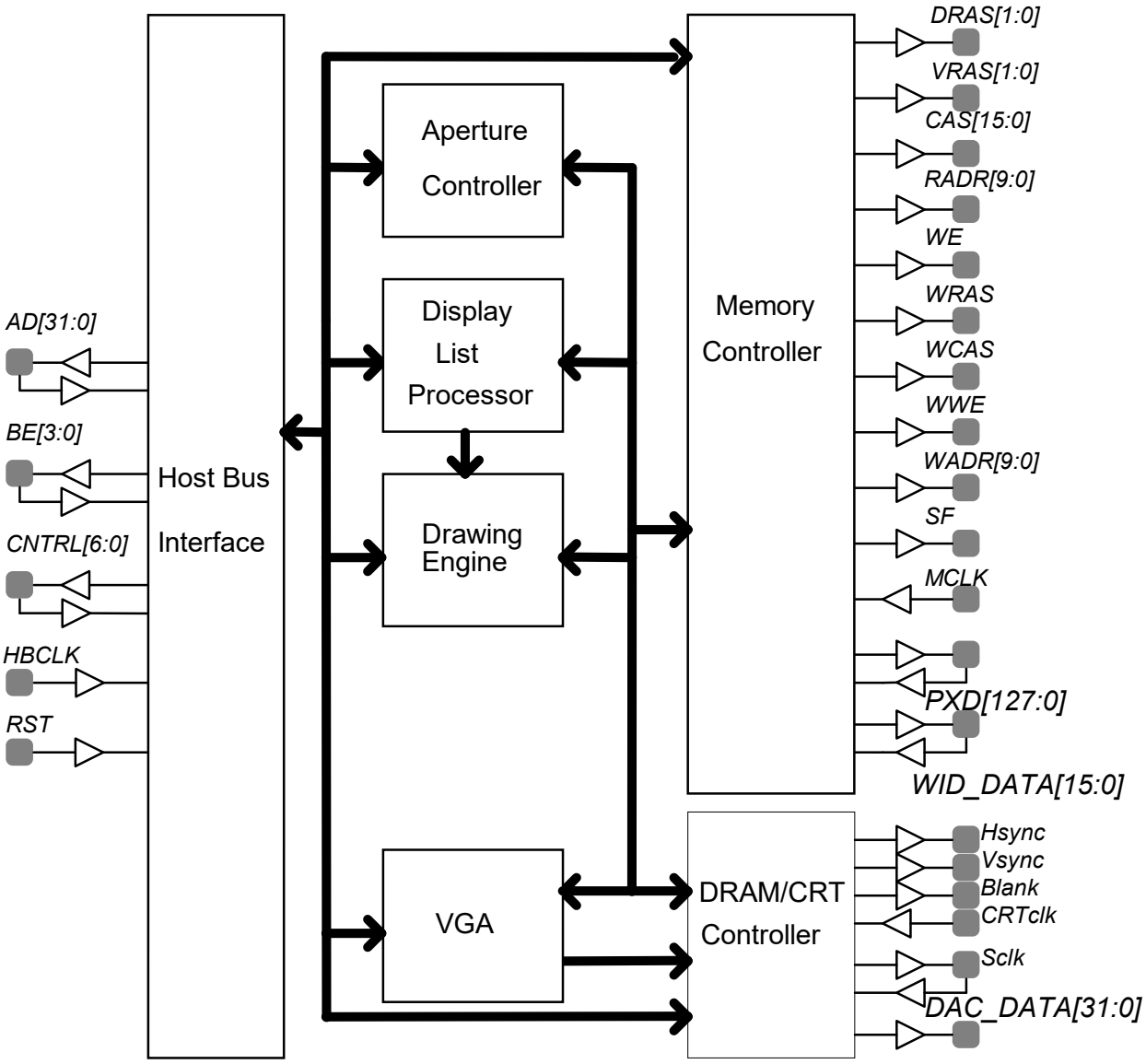
The **Display Controller** provides display refresh data for the external display RAMDAC. The Display Controller reads data from the Virtual or Display Buffers into it's internal display cache and then serializes the data and sends it out over the 32 DAC interface pins. The rate at which this occurs is defined by the setting of the Display Controller registers. The Display Controller allows IMAGINE 128 to operate in a DRAM only system.

The **Drawing Engine** provides all the required logic to implement BITBLT, LINE, LINE_3D, TRIAN_3D, and HOST XFER commands. The Drawing Engine, when triggered, will transfer command and parameter information from the host accessible registers to its own local working registers where it would begin its setup phase. When the Drawing Engine is done with its setup, it will then begin execution of a specific algorithm for the associated command.

For non-rendering commands, after the setup phase, the Drawing Engine begins requesting memory access from the Memory Controller. For 2D and/or 3D rendering commands, the object is piped through the algorithmic rendering engine which begins requesting memory access from the Memory Controller as soon as the first pixel/texel is generated. Up to 2 rendering commands can be piped through the rendering engine at the same time.

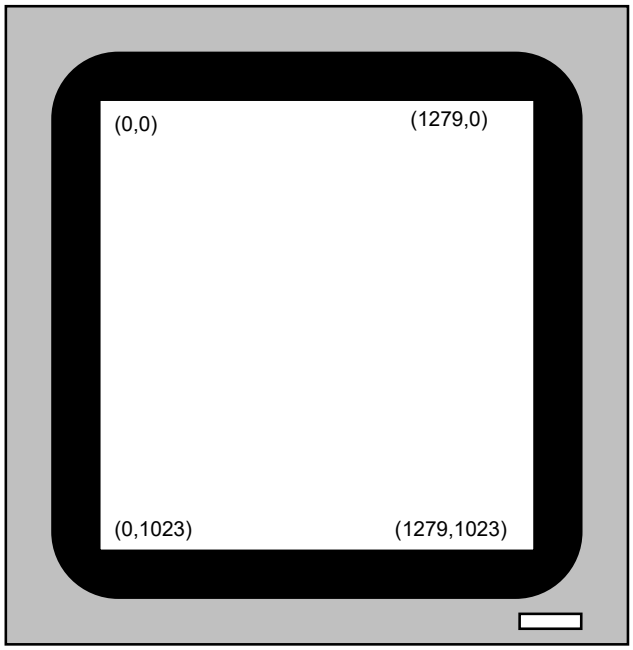
If read data is requested, the memory controller will control the loading of the data into the Drawing Data path and will notify the Drawing Engine that the data is now available. If write data is requested, the data will have been previously setup in the Drawing data path and the Memory controller will control the output of that data to the selected memory buffer.

The **Memory Controller** arbitrates and controls all access to the local buffers by the Host Interface, the CRT controller, the Drawing Engine, and the Display Controller. It also generates control for a FLASH EPROM, a RAMDAC, and an external soft switch register. This unit formats the addresses and generates all of the control signals to support the different local buffers. It also controls the loading and unloading of the host bus cache, the drawing files and the pixel cache.


AD[31:0]
BE[3:0]
CNTRL[6:0]
HBCLK
RST
Host Bus
Interface
Aperture
Controller
Display
List
Processor
Drawing
Engine
VGA
Memory
Controller
DRAM/CRT
Controller
DRAS[1:0]
VRAS[1:0]
CAS[15:0]
RADR[9:0]
WE
WRAS
WCAS
WWE
WADR[9:0]
SF
MCLK
PXD[127:0]
WID_DATA[15:0]
Hsync
Vsync
Blank
CRTclk
Sclk
DAC_DATA[31:0]

2.5 Coordinate System

The screen coordinate system has its origin at the upper left hand corner of the screen, with the X coordinates incrementing left to right and the Y coordinates incrementing top to bottom. The coordinate system for a 1280 by 1024 display is shown below.



Destination X and Y coordinates are 16-bit 2's complement integers. All registers specified in a XY format will be interpreted as 2's complement integers. All internal arithmetic operations are done in 2's complement format; therefore no overflows will be detected or reported. Care must be taken for drawing operations not to step outside of the 16 bit coordinate space.

In the case of the rendering commands, the X and Y coordinates are in IEEE single precision floating point format. The setup for X and Y is done in IEEE single point floating point and is converted to 16-bit 2's complement integers. Again, care must be taken for drawing operations not to step outside of the 16 bit coordinate space.

All three of the local buffers may be accessed in this format by specifying the coordinate, the source and/or destination space, and the buffer pitch. From this organization it can be seen that the pitch for the Display Buffer and Virtual Buffer can be changed on a command by command basis.

Three formats are available for Z coordinate space; they are 16-bit floating point (0-1 input range) full setup, 24-bit floating point (0-1 input range) full setup and 44 bit unsigned fixed point (32.12) which must be setup by the driver. All internal arithmetic operations are done in IEEE single precision floating point; therefore no overflows are detected or reported. Care must be taken for drawing operations not to step outside the Z coordinate space.