2

Functional Description

Section 2: Functional Description

2.1 Overview

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

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

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

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, SilverHammer executes the selected command without any further processor intervention.

2.2 Memory Buffer

SilverHammer supports one local memory buffer up to 32Mbytes with a data bus width of 128 bits of either SGRAM or WRAM memory.

These buffers may be accessed as linear buffers through the Memory windows interface or through the drawing engine.

The Local Buffer may be used as a display buffer, as well as off-screen memory to be used for the storage and manipulation of bitmaps, texture maps, Z buffering or fonts.

2.3 Register Map

This section provides a brief overview of all 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 addresses of these registers are absolute I/O addresses.

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 (*See Appendix D.2.4),* it can be in one of two modes: snooping and owning *(See Section 5.2).*

I/O ADDRESS	REGISTER NAME	DESCRIPTION
0x03C6	PEL_MASK (DAC_02)	Pixel Mask
0x03C7	RD_ADR (DAC_03)	Read Address
0x03C8	WR_ADR (DAC_00)	Write Address
0x03C9	PAL_DAT (DAC_01)	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 4 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	SGR_CONFIG	SGRAM Configuration Register
0x0028	SSWTCH	Soft Switch Register
0x002C	DDC	DDC 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	Reserved	Reserved
0x0050	MW1_ORG	Memory Window 1 Origin (shadowed)
0x0054	MW1_ORG	Memory Window 1 Origin (shadowed)
0x0058	Reserved	Reserved
0x005C	Reserved	Reserved
0x0060	Reserved	Reserved
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	DAC_00	DAC Register 00
0x0084	DAC_01	DAC Register 01
0x0088	DAC_02	DAC Register 02
0x008C	DAC_03	DAC Register 03
0x0090	DAC_04	DAC Register 04
0x0094	DAC_05	DAC Register 05
0x0098	DAC_06	DAC Register 06
0x009C	DAC_07	DAC Register 07
0x00A0	DAC_08	DAC Register 08
0x00A4	DAC_09	DAC Register 09
0x00A8	DAC_0A	DAC Register 0A
0x00AC	DAC_0B	DAC Register 0B
0x00BO	DAC_0C	DAC Register 0C
0x00B4	DAC_0D	DAC Register 0D
0x00B8	DAC_0E	DAC Register 0E
0x00BC	DAC_0F	DAC Register 0F
0x00D0	DMA_SRC	DMA Address Source Register
0x00D4	DMA_DST	DMA Address Destination Register
0x00D8	DMA_CMD	DMA PCI Command Register
0x00DC	Reserved	Reserved
0x00E0	PCI_BMWA	PCI Bus Master Write Address Register
0x00E4	PCI_BMTM	PCI Bus Master Trigger Mask Register

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.

MEMORY ADDRESS	REGISTER NAME	DESCRIPTION
0x0000	DAC_00	DAC Register 00
0x0004	DAC_01	DAC Register 01
0x0008	DAC_02	DAC Register 02
0x000C	DAC_03	DAC Register 03
0x0010	DAC_04	DAC Register 04
0x0014	DAC_05	DAC Register 05
0x0018	DAC_06	DAC Register 06
0x001C	DAC_07	DAC Register 07
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.
0x0060	DB_ADR2	CRT Display Start Address 2
0x0070	DAC_00	DAC Register 00
0x0074	DAC_01	DAC Register 01
0x0078	DAC_02	DAC Register 02
0x007C	DAC_03	DAC Register 03
0x0080	DAC_04	DAC Register 04
0x0084	DAC_05	DAC Register 05
0x0088	DAC_06	DAC Register 06
0x008C	DAC_07	DAC Register 07
0x0090	DAC_08	DAC Register 08
0x0094	DAC_09	DAC Register 09
0x0098	DAC_0A	DAC Register 0A
0x009C	DAC_0B	DAC Register 0B
0x00A0	DAC_0C	DAC Register 0C
0x00A4	DAC_0D	DAC Register 0D
0x00A8	DAC_0E	DAC Register 0E
0x00AC	DAC_0F	DAC Register 0F

2.3.4 Memory Mapped Global Interrupt/Control Registers

The following is a list of the memory mapped global interrupt/control registers. The base address of these registers is 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_CMD	DMA PCI Command Register
0x00DC	Reserved	Reserved
0x00E0	PCI_BMWA	PCI Bus Master Write Address Register
0x00E4	PCI_BMTM	PCI Bus Master Trigger Mask Register

2.3.5 Memory Windows Registers

The following is a list of the memory mapped Memory Windows™ registers. The base address of these registers is 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	DE_TPTCH	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
0x00A4	Reserved	Reserved
0x00A8	Reserved	Reserved

REGISTER ADDRESS	**REGISTER NAME**	**DESCRIPTION**

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	LOD0	Level of detail 0 Origin
0x00D4	LOD1	Level of detail 1 Origin
0x00D8	LOD2	Level of detail 2 Origin
0x00DC	LOD3	Level of detail 3 Origin
0x00E0	LOD4	Level of detail 4 Origin
0x00E4	LOD5	Level of detail 5 Origin
0x00E8	LOD6	Level of detail 6 Origin
0x00EC	LOD7	Level of detail 7 Origin
0x00F0	LOD8	Level of detail 8 Origin
0x00F4	LOD9	Level of detail 9 Origin
0x00F8	DL_ADR	Display list address
0x00FC	DL_CNTRL	Display list control
0x0100	DE_ZORG	Z-buffer origin
0x0104	Reserved	Reserved
0x0108	Reserved	Reserved
0x010C	Reserved	Reserved
0x0110	Reserved	Reserved
0x0114	Reserved	Reserved
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	TEX_BORDER	Texture Border Color (RGB)
0x0130	V0_A_FP	Vertex0 Alpha Channel Floating point Input
0x0134	V0_R_FP	Vertex0 Red Channel Floating Point Input
0x0138	V0_G_FP	Vertex0 Green Channel Floating Point Input
0x013C	V0_B_FP	Vertex0 Blue Channel Floating Point Input
0x0140	V1_A_FP	Vertex1 Alpha Channel Floating Point Input
0x0144	V1_R_FP	Vertex1 Red Channel Floating Point Input
0x0148	V1_G_FP	Vertex1 Green Channel Floating Point Input
0x014C	V1_B_FP	Vertex1 Blue Channel Floating Point Input

REGISTER ADDRESS	REGISTER NAME	DESCRIPTION
0x0150	V2_A_FP	Vertex2 Alpha Channel Floating Point Input
0x0154	V2_R_FP	Vertex2 Red Channel Floating Point Input
0x0158	V2_G_FP	Vertex2 Green Channel Floating Point Input
0x015C	V2_B_FP	Vertex2 Blue Channel Floating Point Input
0x0160	KEY_3D_LOW	3D Key value, low color
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	GLBLENDC	OpenGL Blend Color
0x01E4	Reserved	Reserved
0x01E8	Reserved	Reserved
0x01EC	Reserved	Reserved
0x01F0	Reserved	Reserved
0x01F4	Reserved	Reserved
0x01F8	Reserved	Reserved
0x01FC	Reserved	Reserved

2.4 Block Diagram

The IMAGINE 128⊠ is partitioned into five functional sections: They are the **Host Bus Interface**, the **Aperture Controller**, the **Drawing Engine**, the **CRT Controller**, and the **Memory Controller**.

The **Host Bus Interface** provides an interface to the system bus (PCI or PCI/AGP). It implements a full PCI slave interface, responding to reads and writes of configuration, memory, and I/O cycles. It also implements a PCI master interface for specific memory writes (See Section 5.6). Lastly, it implements an AGP master interface for DMA reads (See Section 5.5, 5.9). It also generates peripheral bus control for a FLASH EPROM, a external RAMDAC, and an external soft switch register.

The **Linear Windows 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 **Drawing Engine** provides all the required logic to implement BITBLT, LINE, LINE_3D, TRIAN_3D, and HOST XFER commands. The Drawing Engine, when triggered, transfers command and parameter information from the host accessible registers to its own local working registers where it begins its setup phase. When the Drawing Engine is done with its setup, it begins the 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.

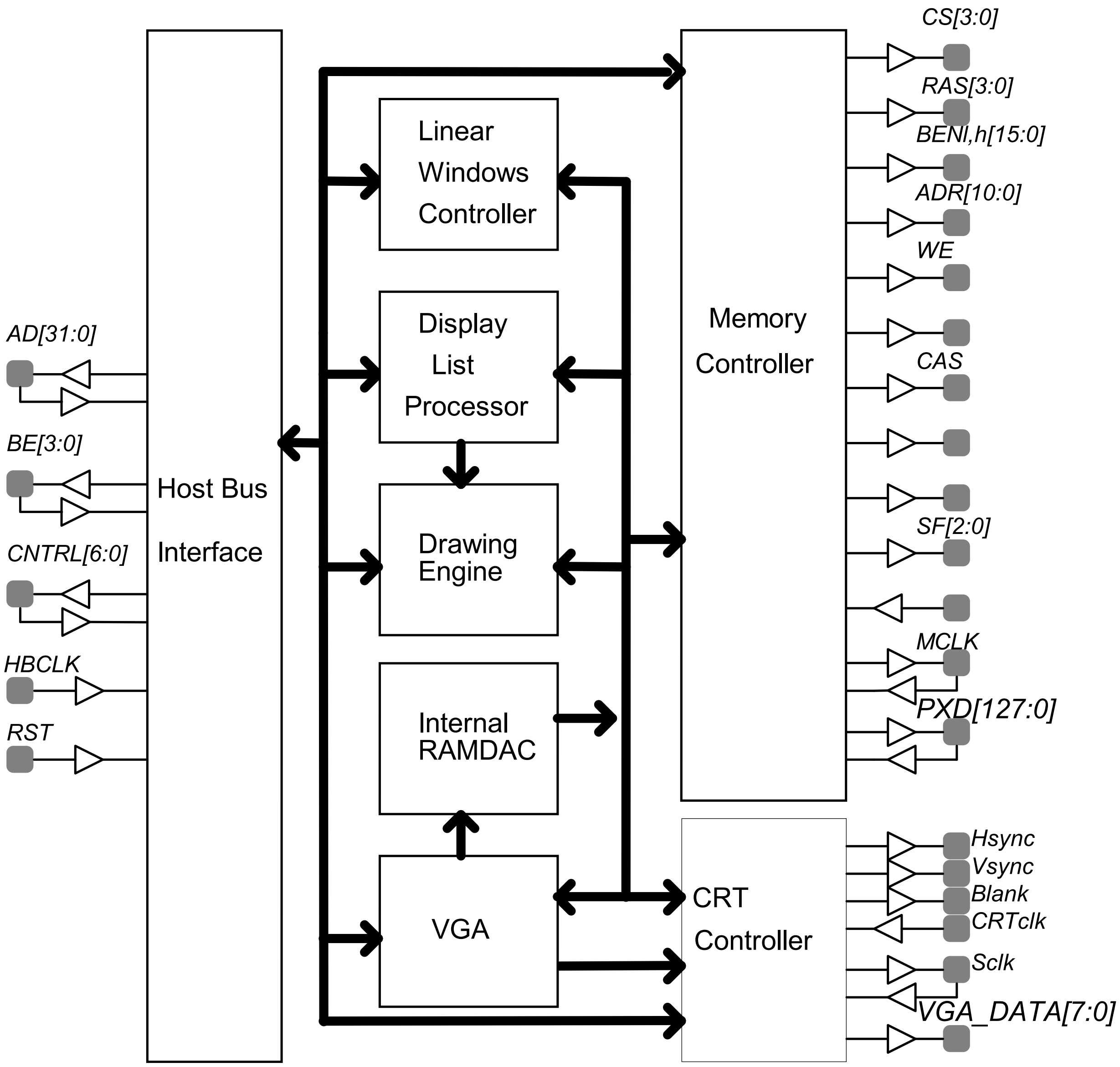
<DLP Description>

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. A free running frame counter which generates interrupts to the Host is also provided. This is useful for synchronizing bit map copies. **CRT Controller** also provides display refresh data for the internal (for SGRAM memory) or external (for WRAM memory) RAMDAC. There is an 8 bit VGA pixel data port when using an external RAMDAC and WRAM memory.

The **Memory Controller** arbitrates and controls all access to the local memory buffer by the Host Interface, the CRT controller, and the Drawing Engine. This unit provides support for WRAM or SGRAM memory.

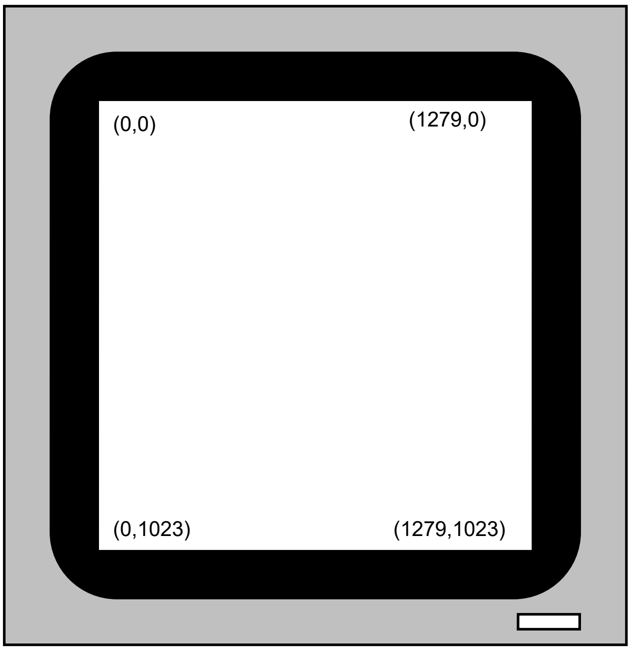
<VGA Description>

<Internal RAMDAC>


CS[3:0]
RAS[3:0]
BENl,h[15:0]
ADR[10:0]
WE
CAS
SF[2:0]
MCLK
PXD[127:0]
Hsync
Vsync
Blank
CRTclk
Sclk
VGA_DATA[7:0]
AD[31:0]
BE[3:0]
CNTRL[6:0]
HBCLK
RST
Host Bus Interface
Linear Windows Controller
Display List Processor
Drawing Engine
Internal RAMDAC
VGA
Memory Controller
CRT Controller

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.

The display buffer is 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 of the display buffer can be changed on a command by command basis. <Steve, logical buffers and pitches>

Two formats are available for Z coordinate space; they are 16-bit floating point (0-1 input range) full setup and 24-bit floating point (0-1 input range) full setup.