APPENDIX D

IMAGINE 128 VGA SPECIFICATION

Appendix D: Imagine 128 VGA Specification

D.1 Internal VGA

IMAGINE has an internal VGA core. VGA operation is enabled by configuration jumpers. CJ[37:36] are pulled down/up respectively to set the PCI device class to VGA. No VGA operation is enabled if the device class is not set to VGA.

D.1.1 Supported Modes

The core supports all standard VGA modes as well as some extended modes. The following table shows the modes that will be supported in the IMAGINE 128:

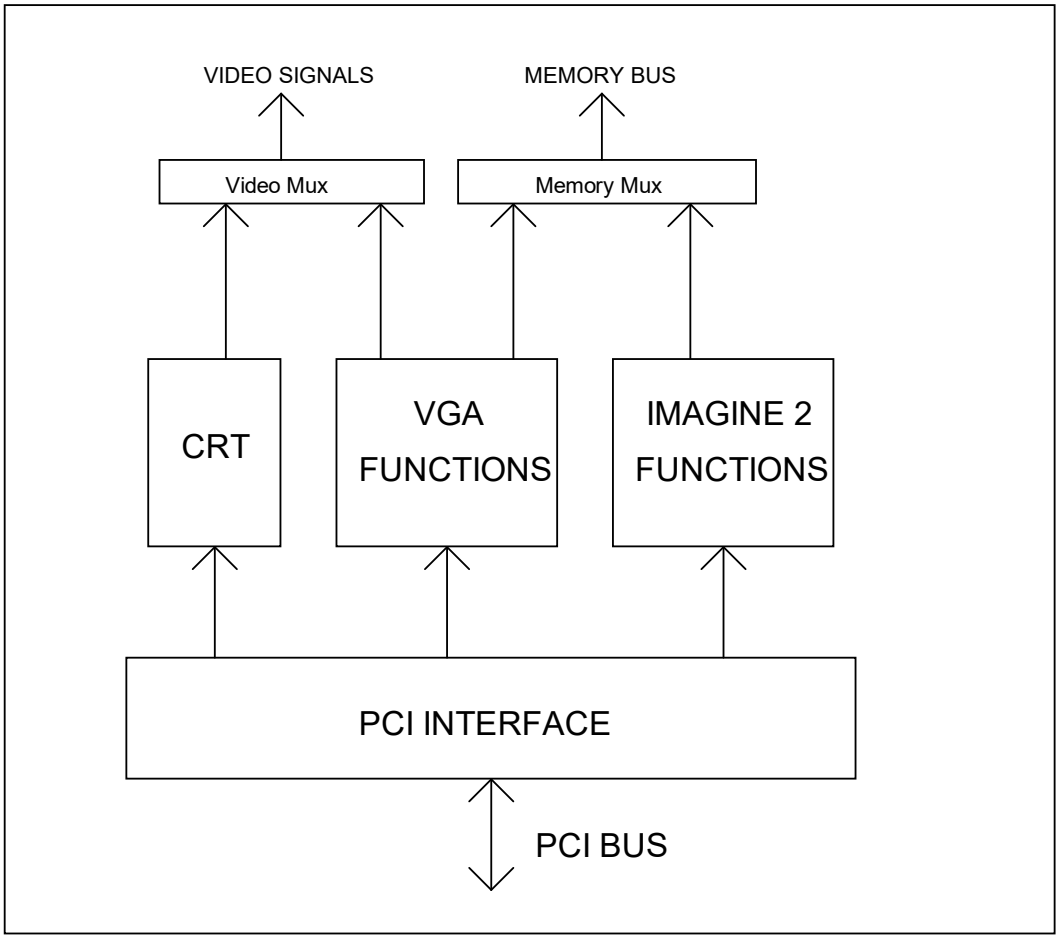
Standard VGA Mode Table

Mode No.	Screen Format	Colors	Mode Type
00 / 01	320 X 200	16	Text
00* / 01*	320 X 350	16	Text
00+ / 01+	360 X 400	16	Text
02 / 02	640 X 200	16	Text
02* / 03*	640 X 350	16	Text
02+ / 03+	720 X 400	16	Text
04 / 05	320 X 200	4	Graphics
06	640 X 200	2	Graphics
07	720 X 350	2	Text
07+	720 X 400	2	Text
0D	320 X 200	16	Graphics
0E	640 X 200	16	Graphics
0F	640 X 350	2	Graphics
10	640 X 350	16	Graphics
11	640 X 480	2	Graphics
12	640 X 480	16	Graphics
13	320 X 240	256	Graphics

*EGA Compatibility using 14 line font +Enhanced VGA modes using 16 line font

D.2 IMAGINE II VGA Architecture

The block diagram below shows how the VGA core subsystem interfaces into the rest of the IMAGINE 3 Design:



D.2.1 IMAGINE VGA Architecture

The VGA subsystem in IMAGINE consists of three parts: the VGA core, the VGA host interface, and the VGA memory interface. The VGA host interface accepts decoded PCI cycles from the PCI Host Interface block. The decoded cycles are then presented to the VGA core which processes the cycles. The VGA core will then request memory cycles to the VGA memory interface. The VGA memory interface in turn will generate the actual memory timing. The VGA subsystem runs off of the memory controller clock. The VGA video timing is generated from the CRT clock.

D.2.2 IMAGINE VGA Operation

The VGA core subsystem in IMAGINE is controlled by the VGA_CTRL register, the PCI device class that is determined by the state of CJ[37:36], and the internal VGA enable jumper (CJ[54]). The programmer interface for the VGA_CTRL register in internal VGA mode is shown at the end of this document. Before enabling VGA, the chip must be initialized for normal operation: enable the appropriate register block and memory window decode in CONFIG1, set the memory bus control in CONFIG2, and program the appropriate wait states also in CONFIG2.

On power up, the VGA_CTRL register is reset to all zero, indicating that all VGA cycle decode on the PCI Bus is disabled. If the PCI device class was set to VGA, the following settings in the VGA_CTRL register will enable VGA operation:

Set the VGA_EN = 1. This bit enables PCI decode of VGA cycles. If this bit is left in its default state of 0, all VGA decode, both memory and I/O, is disabled.

Set MEM_MUX = 1. Setting this to 1 enables the VGA generated memory timing signals to be present on the memory bus. Leaving this bit a 0 allows IMAGINE memory signals to be present on the memory bus. RAMDAC and PROM timing are always generated from the IMAGINE memory controller, not from the VGA subsystem.

Set VID_MUX = 1. Setting this to 1 enables the VGA generated video timing and data signals to be present on the sync/blank and DRAM DAC pins. Leaving this bit a 0 allows IMAGINE generated video timing and data signals to be present on the sync/blank and DRAM DAC pins. In DDC mode, this pin has no effect.

Set MEM_EN = 1. This enables VGA memory decode.

Set VDE = 1. This enables VGA DAC access.

D.2.3 VGA Decode

The VGA subsystem in IMAGINE has a number of options that determine what I/O and memory space is decoded on the PCI bus. For any VGA decode to be enabled, both the device class must be set to VGA (via configuration jumper) and the VGA_EN must be set to 1. If either of these conditions is not true, then no memory or I/O decode will be enabled. Please note that this means that VGA DAC decode (IO x3C6 - x3C9) will also be completely disabled.

Memory decode is based on the VGA mode. The VGA will decode one of the following ranges:

A:0000 - B:FFFF

A:0000 - A:FFFF

B:0000 - B:FFFF

B:8000 - B:FFFF

All VGA memory decode may be disabled by setting the MEM_EN bit in VGA_CTRL to 0. Doing so will not affect the I/O decode. This feature is provided so that linear memory window 1 may be programmed to decode VGA memory space. This can be achieved by programming the MW1 registers through I/O space. Scratch registers are also provided in I/O space so the original MW1 values may be preserved. The I/O mapping of the VGA_CTRL, MW1, SCRATCH, and DAC registers is attached at the end of this document.

VGA I/O decode is also dependent on the mode. The following I/O locations are decoded for the VGA subsystem. An X indicates “B” for monochrome modes and “D” for color modes:

3C0, 3C1, 3C2, 3C4, 3C5, 3CA, 3CC, 3CE, 3CF

3XA, 3X4, 3X5

I/O 3C6 - 3C9 are decoded to the normal DAC decode logic and not to the VGA subsystem

Imagine 2 VGA Control Register

read /write at address = PCIB5 + 0x30

all registers default to 0 on reset.

15	14	13	12	11	10	9	8
offset[7]	offset[6]	offset[5]	offset[4]	offset[3]	offset[2]	offset[1]	offset[0]

7	6	5	4	3	2	1	0
vde	sh_3c2	stretch	vga_buf	mem_en	vid_mux	vga_en	mem_mux

offset[7:0] In VGA mode, this field represents an 8 kilobyte offset into the VGA frame buffer. The VGA frame buffer is only 64 bits wide, so on a 4 Mbyte board, there is only 2 Mbytes of VGA frame buffer accessible. (In SGRAM configuration the offset is always set to 0x00 regardless the settings in this register).

vde This bit controls whether VGA DAC accesses are decoded. If VGA DAC accesses are enabled, then the cycles will be "owned" or "snooped' based on the PCI DAC snoop bit in the PCI command register. All DAC accesses are routed to the external RAMDAC.

vde = 1 VGA DAC cycles are decoded
vde = 0 VGA DAC cycles are ignored (default)

sh_3c2 This bit allows writes to VGA register x3C2 bit[0] to be shadowed to an internal register. This allows quick switching between the "B" and "D" register sets. If x3C2 shadowing is disabled, register set decode comes from the VGA core.

sh_3c2 = 1 Shadowing of x3C2 enabled
sh_3c2 = 0 Shadowing of x3C2 disabled (default)

stretch This bit lengthens all memory cycles allowing for faster memory clock. See timing diagrams for VGA cycles at EDO and Window RAM configuration. For SGRAM mode set this bit to 0.

stretch = 1 extended cycles
stretch = 0 normal cycles (default)

Note: No other register in Imagine3 affects timing of memory cycles generated in VGA mode.

vga_buf This bit selects the Imagine memory buffer in which the VGA frame buffer is stored.

vga_buf = 1 Store VGA Image in the Virtual Buffer
vga_buf = 0 Store VGA Imagine in the Display Buffer (default)

Imagine 2 VGA Control Register (*continued)*

read /write at address = PCIB5 + 0x30

all registers default to 0 on reset.

15	14	13	12	11	10	9	8
offset[7]	offset[6]	offset[5]	offset[4]	offset[3]	offset[2]	offset[1]	offset[0]

7	6	5	4	3	2	1	0
vde	sh_3c2	stretch	vga_buf	mem_en	vid_mux	vga_en	mem_mux

mem_en This bit enables overall VGA decode of the A and B segment of memory. If enabled, the appropriate portion of A or B segment will be decoded, depending on the VGA mode currently active. If disabled, the VGA subsystem will not decode any memory space, although I/O access will still be enabled.

mem_en = 1 VGA Memory decode enabled
mem_en = 0 VGA Memory decode disabled (default)

vid_mux This bit controls whether VGA data and syncs or high resolution video data and synchs are output on the DRAM DAC interface.

vid_mux = 1 VGA data and syncs to DAC
vid_mux = 0 Normal data and syncs to DAC (default)

vga_en This is the VGA master decode enable. If this bit is 0, all VGA decode on the PCI Bus is disabled.

vga_en = 1 Enable PCI VGA Decode
vga_en = 0 Disable PCI VGA Decode (default)

mem_mux This bit controls whether the VGA or high resolution subsystem has access to the frame buffer. Even when the VGA has access to the frame buffer, access to the DAC and PROM is still available through the high resolution subsystem.

mem_mux = 1 VGA has frame buffer access
mem_mux = 0 Normal Frame buffer access (default)

Imagine 2 Extended I/O Registers

Address (rbase_io +)	Name
0x2C	MON_INFO
0x30	VGA_CTRL
0x40	MW1_CTRL
0x44	MW1_ADDR
0x48	MW1_SZ
0x50	MW1_ORG
0x54	MW1_ORG
0x64	MW1_MASK
0x68	SCRATCH_1
0x6C	SCRATCH_2
0x70	SCRATCH_3
0x74	SCRATCH_4
0x78	Reserved
0x7C	Reserved
0x80	DAC Address 0
0x84	DAC Address 1
0x88	DAC Address 2
0x8C	DAC Address 3
0x90	DAC Address 4
0x94	DAC Address 5
0x98	DAC Address 6
0x9C	DAC Address 7
0xA0	DAC Address 8
0xA4	DAC Address 9
0xA8	DAC Address A
0xAC	DAC Address B
0xB0	DAC Address C
0xB4	DAC Address D
0xB8	DAC Address E
0xBC	DAC Address F