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 jumper. CP[28] is 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. The following table shows the modes that are supported in the IMAGINE 128:

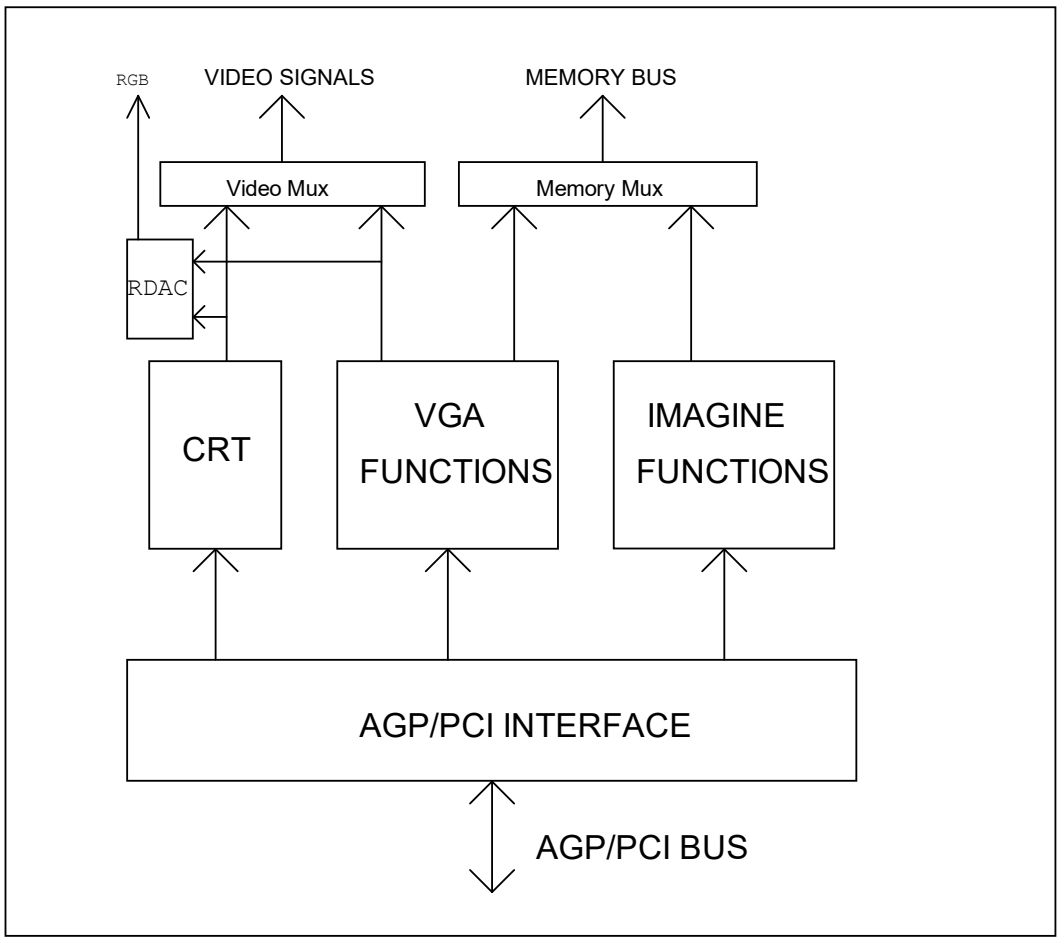
Standard VGA Mode Table

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 200	256	Graphics

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

D.2 IMAGINE/VGA Architecture

The block diagram below shows how the VGA core subsystem interfaces into the rest of the IMAGINE 128 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 programmer interface for the VGA_CTRL register 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 zero, indicating that all VGA cycle decode on the PCI Bus is disabled. If the PCI device class was set to VGA (CP[28]=1), the following settings in the VGA_CTRL register will enable VGA operation:

Set 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 VGA_MUX = 1. Setting this bit 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. Setting this bit to 1 also enables the VGA generated video timing and data signals to be present on the sync/blank pins and pixel data bus of an external/internal RAMDAC. Leaving this bit a 0 allows IMAGINE generated video timing to be present on the sync/blank pins of an external/internal RAMDAC.

In DDC1 mode, the VSYNC pin is not affected by this bit.

RAMDAC and PROM timing are always generated from the IMAGINE host bus controller, not from the VGA subsystem.

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 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, based on the VGA register GR6, bits [3:2]:

0xA0000 - 0xBFFFF

0xA0000 - 0xAFFFF

0xB0000 - oxB7FFF

0xB8000 - 0xBFFFF

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 DDC, 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 VGA Control Register

read /write at address = PCIB4 + 0x30

all registers default to 0 on reset.

7	6	5	4	3	2	1	0
vde	win_rst	stretch	reserved	mem_en	reserved	vga_en	vga_mux

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 nternal/external RAMDAC.

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

win_rst This bit is valid only in WRAM configuration (don't care with SGRAM memories). Normally this bit is set to 0
Any time when vga_mux bit is toggled (switching memory bus for Imagine or VGA core access) the onboard WRAM memories may need to be reset. In this case win_rst bit should be set to one for at least one full period of DRAM refresh cycles. With win_rst set to one, all DRAM refresh cycles (CAS before RAS) get converted to WRAM reset cycles. After that win_rst should be set again to zero.

stretch This bit lengthens all memory cycles allowing for faster memory clock.
See timing diagrams for VGA cycles 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 Imagine4 affects timing of memory cycles generated in VGA mode .

mem_en This bit enables overall VGA decode of the A000 and B000 segments of memory. If enabled, the appropriate portion of A000 or B000 segments will be decoded, depending on the setting of VGA register GR6[3:2]. 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)

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)

vga_mux This bit controls whether the VGA or Imagine high resolution subsystem has access to the frame buffer and the RAMDAC (syncs and video data).

vga_mux = 1 VGA subsystem has access to the frame buffer and DAC
vga_mux = 0 Imagine has Frame buffer/DAC access (default)

Imagine I/O Registers

Address (PCIB4 + x)	Name
0x2C	DDC
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
0x80	DAC0
0x84	DAC1
0x88	DAC2
0x8C	DAC3
0x90	DAC4
0x94	DAC5
0x98	DAC6
0x9C	DAC7
0xA0	DAC8
0xA4	DAC9
0xA8	DAC10
0xAC	DAC11
0xB0	DAC12
0xB4	DAC13
0xB8	DAC14
0xBC	DAC15