5

Register Set

Section 5: REGISTER SET

Undefined bits in the registers are always read back as zero and should be written as zero.

5.1 Addressing Configuration Registers

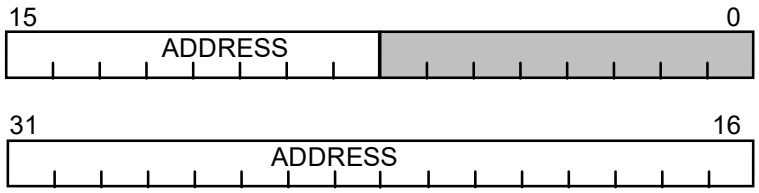
The addressing configuration group of registers are I/O mapped. The address of these registers is defined as follows. The value written into PCI base address Register 4 is concatenated with the fixed address of the register to determine the system address of the register.

5.1.1 Register Base Address for the Global Register Block {PCIB4, (0x0000)}

Name: RBASE_G

Type: I/O space read write

This register defines the start address for the memory mapped global register block. This register will be written with the same value as Base Address Register 4.



Bits	Name	Default	Function
RBASE_G[31:8]	ADDRESS		Address decode value

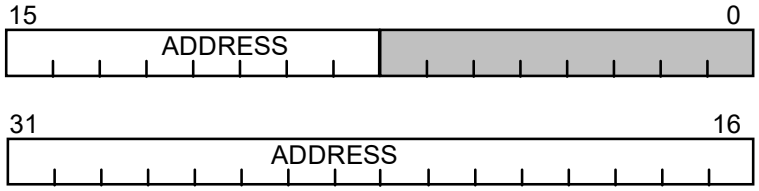
Note: PCIB4 = The value written into PCI Base 4 Register.

5.1.2 Memory Windows™ Register Block Base Address Register {PCIB4, (0x0004)}

Name: RBASE_W

Type: I/O space read write

This register defines the start address for the Memory Windows configuration register block. This register will be written with the same value as Base Address Register 4 plus an 8 kilobyte offset.



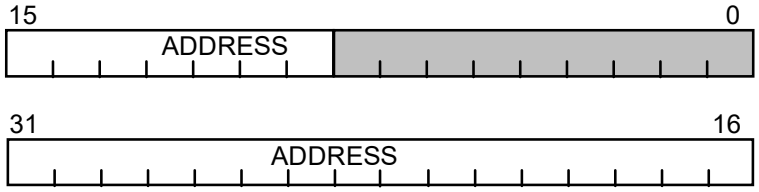
Bits	Name	Default	Function
RBASE_W [31:8]	ADDRESS		Address decode value

5.1.3 Register Base Address Drawing Engine {PCIB4,(0x0008)}

Name: RBASE_D

Type: I/O space read write

This register defines the start address for the memory mapped register block. This register is written with the same value as Base Address Register 4 plus a 16 kilobyte offset.



Bits	Name	Default	Function
RBASE_D[31:8]	ADDRESS		Address decode value

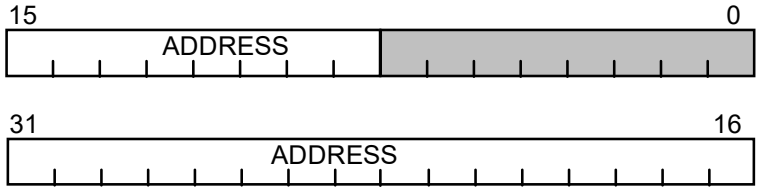
Note: PCIB4 = The value written into PCI Base 4 Register.

5.1.4 Register Base Address Global Interrupt Registers {PCIB4,(0x0010)}

Name: RBASE_I

Type: I/O space read write

This register defines the start address for the memory mapped global interrupt register block. This register will be written with the same value as Base Address Register 4 plus a 32 kilobyte offset.



Bits	Name	Default	Function
RBASE_I[31:8]	ADDRESS		Address decode value

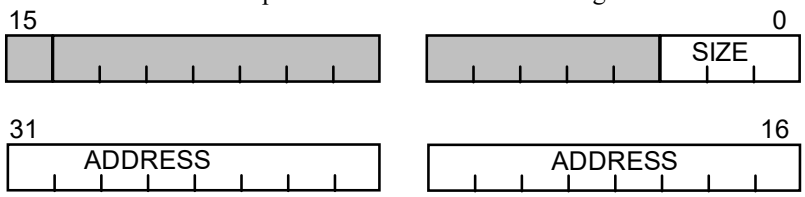
Note: PCIB4 = The value written into PCI Base 4 Register.

5.1.5 Register Base Address/Size EPROM registers {PCIB4, (0x0014)}

Name: RBASE_E

Type: I/O space read write

This register defines the start address for the memory mapped EPROM.. This register will be written with the same value as the expansion ROM Base Address Register and the size will be set to 64KB.



Bits	Name	Default	Function
RBASE_E[31:16]	ADDRESS	0x0000	EPROM Address decode value
RBASE_E[2:0]	SIZE	0x1	EPROM Size 64 KB (**read only**)

Note: PCIB4 = The value written into PCI Base 4 Register.

5.2 VGA DAC Shadow Registers

VGA DAC accesses are routed to the external RAM DAC in two ways: snooping and owning. When snooping is enabled, VGA DAC writes will be routed to the RAM DAC, but ignored on the bus. Read cycles will be ignored. If snooping isn't enabled, then IMAGINE 128 will own the VGA DAC cycles. IMAGINE 128 will claim VGA DAC cycles on the bus. Both reads and writes will be routed to the RAM DAC. Snooping is controlled by bit 5 in PCI configuration register 1. VGA DAC cycles decoded is controlled by the "vde" bit in the VGA_CTRL register, (*see section E.2.3)*.

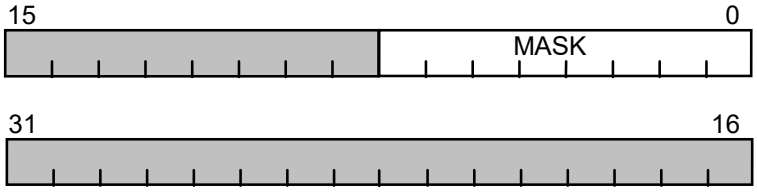
(See Tom on organization of DAC section).

5.2.1 Pixel Mask Registers 0 x 03C6

Name: PEL_MASK

Type: I/O or Memory mapped read write

The contents of this register are logically AND'ed with the pixel input to the VGA color palette.

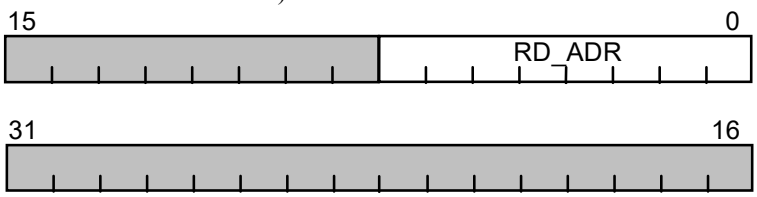


5.2.2 Read Address Register 0x03C7

Name: RD_ADR

Type: I/O or Memory mapped read write

This register defines the read address of the VGA color palette. This register is auto incrementing (with 3 successive reads of 0x3C9).

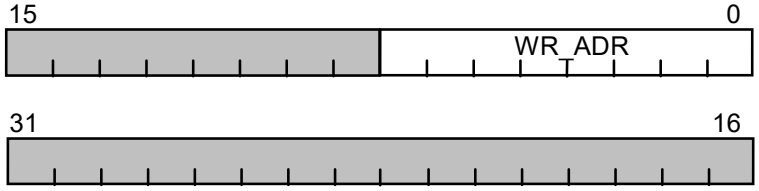


5.2.3 Write Address Register 0x03C8

Name: WR_ADR

Type: I/O or Memory mapped read write

This register defines the starting write address of the VGA color palette. This register is auto incrementing (with 3 successive writes of 0x3C9)..

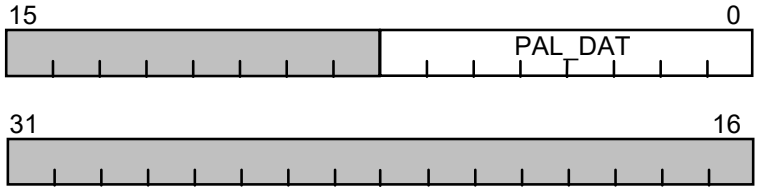


5.2.4 Palette Data Register 0x03C9

Name: PAL_DAT

Type: I/O or Memory mapped read write

This register contains data to be read from or written to the VGA color palette.



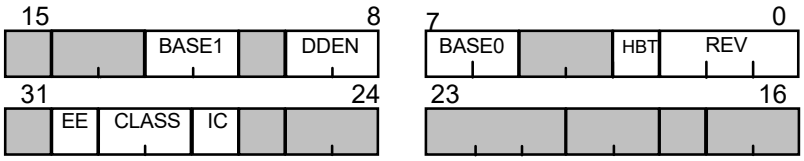
5.3 Miscellaneous I/O Registers

5.3.1 ID Register {PCIB4,(0x0018)}

Name: ID

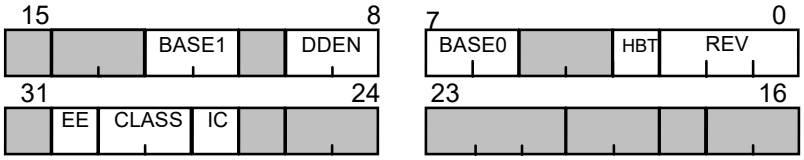
Type: I/O space mapped read only

The ID Register contains the hardwired chip revision as well as configuration information. Configuration pins, as indicated, are used to initialize most of the fields in this register.



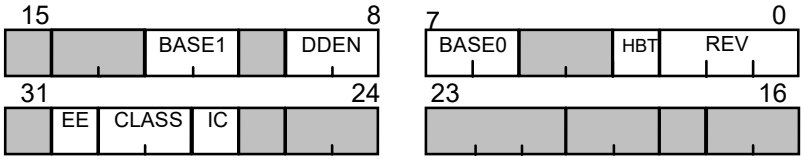
BITS	NAME	Config. pin	VALUE	DESCRIPTION
ID[2:0]	REV			Chip Revision
ID[3]	HBT			Host bus type
		CP[23]=0	0x0	AGP Bus
		CP[23]=1	0x1	PCI Local Bus
ID[7:6]	BASE0			PCI Base 0 Address Register Size
				(Linear Memory Window 0)
		CP[31:30] = 00	0x0	4 Megabyte Memory Space Requested
		CP[31:30] = 01	0x1	8 Megabyte Memory Space Requested
		CP[31:30] = 10	0x2	16 Megabyte Memory Space Requested
		CP[31:30] = 11	0x3	32 Megabyte Memory Space Requested
ID[9:8]	DDEN			Display buffer density
		CP[27:26] = 00	0x0	Reserved
		CP[27:26] = 01	0x1	256K bits by N memory chips
		CP[27:26] =10	0x2	if SGRAM - 16Mbit chips if WRAM - 256Kbit chips in interleaved banks
		CP[27:26] = 11	0x3	Reserved

ID Register {PCIB4,(0x0018)} (Continued)



BITS	NAME	Config. Pin	VALUE	DESCRIPTION
ID[12:11]	BASE1			PCI Base 1 Address Register Size (Linear Memory Window 1)
		CP[31:30] = 00	0x0	4 Megabyte Memory Space Requested
		CP[31:30] = 01	0x1	8 Megabyte Memory Space Requested
		CP[31:30] = 10	0x2	16 Megabyte Memory Space Requested
		CP[31:30] = 11	0x3	32 Megabyte Memory Space Requested

ID Register {PCIB4,(0x0018)} (Continued)



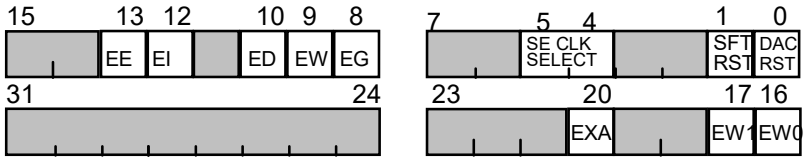
BITS	NAME	Config. Pin	VALUE	DESCRIPTION
ID[23:21]	BASEROM		0x1	PCI EPROM Base Address Register Size 64 Kilobyte Memory Space Requested
ID[27]	IC		1	PCI Interrupt Capability is on
ID[29:28]	CLASS			PCI Device Sub-Class
		CP[28] = 0 CP[28] = 1	0x0 0x2	VGA Other Display Controller
ID[30]	EE			EPROM Boot Decode Enable: This is the read of CP[29] for EPROM boot enable bit.
		CP[29] = 0 CP[29] = 1	0x0 0x1	EPROM Boot Decode disabled EPROM Boot Decode Enabled

5.3.2 Configuration Register One {PCIB4,(0x001C)}

Name: CONFIG1

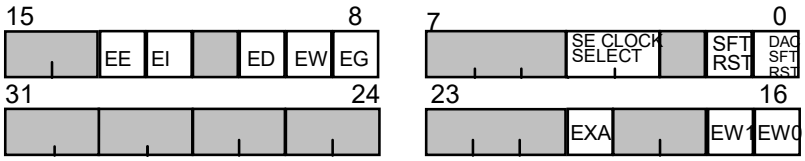
Type: I/O space mapped read write

The decoder enable bits allow certain address decode functions to be individually enabled. Before any address decode register is programmed, the corresponding decode enable bit should be set to 0 (disabled), and then set to 1 after the address range has been programmed. Bit[0] or Bit[1] allow for soft reset generation.



BITS	NAME	VALUE	DESCRIPTION
CONFIG1[0]	DAC_SOFT _RESET		DAC and SilverHammer Software reset bit.
		0	Default, No software reset to the DAC and SilverHammer.
		1	Software reset the DAC and SilverHammer.
CONFIG1[1]	SFT_RST		Software reset bit.
		0	No software reset generated. (default)
		1	Software reset generated.
CONFIG1[3:2]	Reserved		Reserved
CONFIG1[5:4]	SE_CLOCK _SELECT		Setup Engine Clock Source Select
		00	Default, Host Bus Clock as SE Clock
		01	Internally derived 2X Host Bus Clock (for use only in PCI 33MHz systems)
		10	Internally derived DE_CLK/2
		11	SE_CLK pin
CONFIG1[7:6]	Reserved	0	Reserved

Configuration Register One {PCIB4,(0x001C)} (Continued)



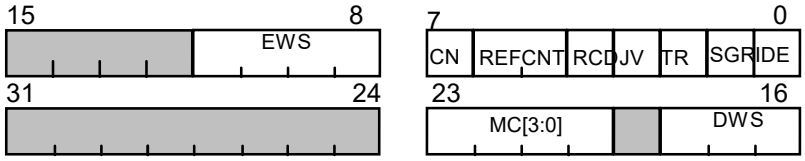
BITS	NAME	VALUE	DESCRIPTION
CONFIG1[20:8]	ENABLE DECODER	0 1 (Defaults)	Disable Decode for a specific block Enable Decode for a specific block
CONFIG1[8]	EG	0	Enable Global Decoder.
CONFIG1[9]	EW	0	Enable Mem Windows Reg Decoder.
CONFIG1[10]	ED	0	Enable Drawing Engine Decoder.
CONFIG1[11]	Reserved	0	Reserved
CONFIG1[12]	EI	0	Enable Global Interrupt Decoder.
CONFIG1[13]	EE	CP[29]	Enable EPROM Decoder.
CONFIG1[15:14]	Reserved	0	Reserved
CONFIG1[16]	EW0	0	Enable Mem Window 0 Decoder.
CONFIG1[17]	EW1	0	Enable Mem Window 1 Decoder.
CONFIG1[19:18]	Reserved	0	Reserved
CONFIG1[20]	EXA	0	Enable XY window Decoder.
CONFIG1[31:21]	Reserved	0	Reserved

5.3.3 Configuration Register Two {PCIB4,(0x0020)}

Name: CONFIG2

Type: I/O space mapped read write

This register contains configuration information for IMAGINE 128 and peripheral devices supported by IMAGINE 128. All bits are write/read except SGR bit which specifies the memory type in the local buffer and is set by Configuration Pin CP[24].

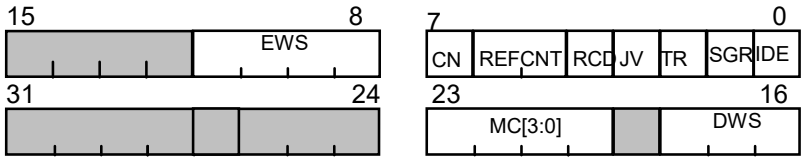


BITS	NAME	VALUE	DESCRIPTION
CONFIG2[0]	IDE (CP[25])		INTERNAL RAMDAC ENABLE
		0	Internal RAMDAC and PLLs disabled
		1	Internal RAMDAC and PLLs enabled
CONFIG2[1]	SGR (CP[24])		This bit (**read only**) defines the type of memory timing generated for the local buffers.
		0	Generate Window RAM timing.
		1	Generate SGRAM timing.
CONFIG2[2]	TRAL		Extended RAS low timing
		0	Normal timing (default)
		1	Extended timing
CONFIG2[3]	JV		This bit enables serial port read transfers to occur in all banks of Display buffer, regardless of the highest order address bit.
		0	Transfer cycles are generated to a single bank of Display buffer
		1	Transfer cycles are generated to all banks of Display buffer (default).
			This bit has a meaning with dual ported memories only (WINDOW RAM).

CONFIG2[4]	RCD		RAS to CAS delay select , this bit defines the delay between the falling edge of RAS and the falling edge of CAS.
		0	Short delay
		1	Normal delay (default)
			Typically this bit should be set to one.
CONFIG[6:5]	REFCNT		Refresh Count Select, these two bits define how often the DRAM refresh cycles occur.
		00	Generate DRAM refresh every 768 mclocks
		01	Generate DRAM refresh every 1024 mclocks
		10	Generate DRAM refresh every 1280 mclocks (default)
		11	Generate DRAM refresh every 3584 mclocks
CONFIG2[7]	CONT		Continuous Memory Cycle Enable This bit determines whether IMAGINE 128® will concatenate multiple memory request that fall within the same row into a single memory cycle.
		0	All new memory requests result in RAS precharge New memory request may be combined (default). Typically this bit should be set to 1.
		1	
CONFIG2[11:8]	EWS		EPROM Wait States : Each EPROM wait state adds one Memory Clock to the EPROM Access time. See IMAGINE 128® timing diagrams for more information.
		0x0	Zero EPROM wait states
		0x1	One EPROM wait states
		0x2	Two EPROM wait states
		0x3	Three EPROM wait states
		•	•
		•	•
		•	•
		.0xD	Thirteen EPROM wait states
		0xE	Fourteen EPROM wait states
		0xF	Fifteen EPROM wait states (Default)

CONFIG2[18:16]	DWS		DAC Wait States. These bits control WRITE/READ signals to internal and external RAMDAC as follows:
		0x0	WR/RD low = 3 clocks ; WR/RD high min. = 4 clocks
		0x1	WR/RD low = 3 clocks ; WR/RD high min. = 16 clocks
		0x2	WR/RD low = 5 clocks ; WR/RD high min. = 4 clocks
		0x3	WR/RD low = 5 clocks ; WR/RD high min. = 16 clocks
		0x4	WR/RD low = 7 clocks ; WR/RD high min. = 4 clocks
		0x5	WR/RD low = 7 clocks ; WR/RD high min. = 16 clocks
		0x6	WR/RD low = 9 clocks ; WR/RD high min. = 4 clocks
		0x7	WR/RD low = 9 clocks ; WR/RD high min. = 16 clocks
			Default is 0x3. Clocks refer to PCI interface clock. For optimum performance of the internal RAMDAC set this register to: 0x1 - if 33MHz PCI bus 0x3 - if AGP or 66MHz PCI bus

Configuration Register Two (Continued)



BITS	NAME	VALUE	DESCRIPTION
CONFIG2[20]	MC[0]		Memory Control 0 - Display Buffer Enable This bit will tristate all Display Buffer control signals (address, data, control).
		0	Display Buffer signals are tristate (*Default*)
		1	Display Buffer signals are driven
CONFIG2[21]	MC[1]		Memory Control 1 - RAM Refresh Control This bit controls whether DRAM refresh cycles are generated by IMAGINE 128
		0	RAM Refresh cycles generated (*Default*)
		1	RAM refresh disabled

CONFIG2[22]	MC[2]	0 1	Memory Control 2 - Data Sampling Control This bit controls the sampling of data during read cycles from a IMAGINE 128 memory buffer. Normal Data sampling (*Default*) Delayed Data sampling Typically this bit should be set to: 1 - for Window Ram configuration 0 - for SGRAM configuration
CONFIG2[23]	MC[3]	1 0	Memory Control 3 - Memory Control Skew This bit causes certain memory control signals to be skewed to allow more access time at higher MCLK rates. Normal memory control signals (*Default*) Typical value of this bit should remain zero.

SGRAM CONFIGURATION REGISTER {PCIB4,(0x0024}

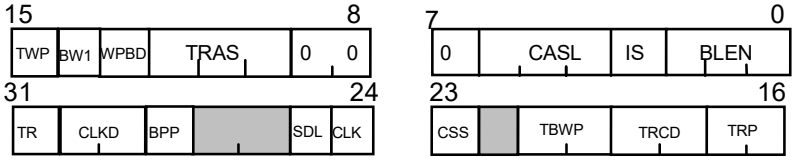
Name: SGR_CONFIG

Type: I/O space mapped read write

Programs SGRAM memories.

After a power up cycle, but before any of local memory buffers have been accessed for the very first time, the SGRAM memories must be initialized with a special configuration cycle.
This configuration cycle may be repeated at any time later if a new set of SGRAM parameters is required. For example: switching from VGA mode to a high resolution mode may require reprogramming SGRAM.

Bit [31] of this register is a "trigger bit". Any time a "one" is written to this bit, the configuration cycle begins and the other bits in this register define SGRAM parameters. No other cycles are allowed to occur at the time when the configuration cycle is pending.



BITS	NAME	DEFAULT VALUE	DESCRIPTION
SGR_CONFIG[2:0]	BLEN	0	Has to be always set to 000, (this setting programs SGRAM memories for burst length of one).
SGR_CONFIG[3]	IS	0	Interleaved or sequential access between banks. IS has to be set to 0 (sequential access).
SGR_CONFIG [6:4]	CASL	0	CAS Latency in memory clocks. (In VGA mode set CASL to 010)
SGR_CONFIG[9:7]	vendor special op	0	Write 000 to these bits.
SGR_CONFIG[12:10]	TRAS	0	RAS active time (minimum). Value in this register should be calculated based on tRAS parameter from SGRAM specification and the actual period of MCLK (memory clock). TRAS =(tRAS/MCLK) -1 Then round up the result to an integer number. Example: tRAS=70ns, MCLK period=11ns. TRAS=70/11-1=5.36=>6. Write TRAS=110.
SGR_CONFIG[13]	WPBD	0	0- writes per bit enabled 1 -writes per bit disabled
SGR_CONFIG[14]	BW1	0	0 - two clocks block writes 1 - one clock block writes
SGR_CONFIG[15]	TWP	0	0 - two clocks from write to precharge 1 -one clock from write to precharge

SGR_CONFIG[17:16]	TRP	00	Precharge cycle (in clocks) 00 - 3 clocks 01 - 1 clock 10 - 2 clock 11 - 3 clocks
SGR_CONFIG[19:18]	TRCD	00	TRCD time (in clocks) 00 - 3 clocks 01 - 1 clock 10 - 2 clocks 11 - 3 clocks
SGR_CONFIG[21:20]	TBWP	00	Block write to precharge time (in clocks) 00 - 2 clock 01 - 1 clock 10 - 3 clocks 11 - 3 clocks
SGR_CONFIG[23]	CSS	0	Chip select skew 0 -normal 1 -delayed
SGR_CONFIG[24]	CLK	0	This bit enables PLL controlling skew between internal and external clock. 0 - PLL disabled (a clock from REFCLK pin is sent instead) 1 - PLL enabled (enable this bit after memory clock synthesizer is properly programmed and stable)
SGR_CONFIG[25]	SDL	0	Read data sampling delay 0 - normal 1 - delayed This bit is OR-ed with CONFIG2[22] bit , so any of these two bits if set to one will delay sampling.
SGR_CONFIG[28]	BPP	0	This bit enables PLL controlling skew between internal and external clock. 0 - PLL enabled 1 - PLL disabled (delayed internal clock is sent to memories, but the skew is not controlled by the PLL)
SGR_CONFIG[30:29]	CLKD	0	Clock delay control 00 - 2 ns + 0 ns 01 - 2ns + 1ns 10 - 2ns + 2 ns 11 - 2 ns + 3 ns
SGR_CONFIG[31]	TR		A write only trigger bit. A “one” written to this bit initialize configuration cycle. The memory controller has to be in idle state to execute this cycle. (Turn off ram refresh, screen/CRT refresh, drawing engine has to be idle)

Bits[23:10] and bit[25] are "don't care" in VGA mode.

Typically the value in SGR_CONFIG[31:0] register should be set to:

8Mbit memories 128kx32x2 -8 (Samsung KM4132G271A-8)
0xA100_0020 in VGA mode
0xA108_9030 in Imagine mode (83MHz <clock <= 100 MHz).
0xA10A_8c20 in Imagine mode at 83 MHz and below

16Mbit memories 256kx32x2 -H (Samsung KM4132G512-H)
0xA100_0020 in VGA mode.
0xA11A_D020 in Imagine mode (83MHz <clock <= 100 MHz).

16Mbit memories 256kx32x2 -8 (Samsung KM4132G512-8)
0xA118_d430 for -8 chips 100 MHz up

5.3.4 SGRAM CONFIGURATION REGISTER {CJ,(0x0024}

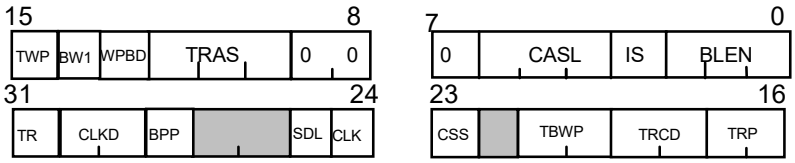
Name: SGR_CONFIG

Type: I/O space mapped read write

Programs SGRAM memories.

After a power up cycle , but before any of local memory buffers have been accessed for the very first time, the SGRAM memories must be initialized with a special configuration cycle.
This configuration cycle may be repeated at any time later if a new set of SGRAM parameters is required. For example: switching from VGA mode to a high resolution mode may require reprogramming SGRAM.

Bit [31] of this register is a "trigger bit". Any time a "one" is written to this bit, the configuration cycle begins and the other bits in this register define SGRAM parameters. No other cycles are allowed to occur at the time when the configuration cycle is pending.



BITS	NAME	DEFAULT VALUE	DESCRIPTION
SGR_CONFIG[2:0]	BLEN	0	Has to be always set to 000, (this setting programs SGRAM memories for burst length of one).
SGR_CONFIG[3]	IS	0	Interleaved or sequential access between banks. IS has to be set to 0 (sequential access).
SGR_CONFIG [6:4]	CASL	0	CAS Latency- Set CASL to: 011 in Imagine mode. 010 in VGA mode
SGR_CONFIG[9:7]	vendor special opr	0	Write 000 to these bits.

SGR_CONFIG[12:10]	TRAS	0	RAS active time (minimum). Value in this register should be calculated based on tRAS parameter from SGRAM specification and the actual period of MCLK (memory clock). TRAS =(tRAS/MCLK) -1 Then round up the result to an integer number. Example: tRAS=70ns, MCLK period=11ns. TRAS=70/11-1=5.36=>6. Write TRAS=110.
SGR_CONFIG[13]	WPB	0	0- writes per bit enabled 1 -writes per bit disabled
SGR_CONFIG[14]	BW1	0	0 - two clocks block writes 1 - one clock block writes
SGR_CONFIG[15]	TWP	0	0 - two clocks from write to precharge 1 -one clock from write to precharge
SGR_CONFIG[17:16]	TRP	00	Precharge cycle (in clocks) 00 - 3 clocks 01 - 1 clock 10 - 1 clock 11 - 3 clocks
SGR_CONFIG[19:18]	TRCD	00	TRCD time (in clocks) 00 - 3 clocks 01 - 1 clock 10 - 2 clocks 11 - 3 clocks
SGR_CONFIG[21:20]	TBWP	00	Block write to precharge time (in clocks) 00 - 1 clock 01 - 1 clock 10 - reserved 11 - reserved
SGR_CONFIG[23]	CSS	0	Chip select skew 0 -normal 1 -delayed
SGR_CONFIG[24]	CLK	0	This bit enables PLL controlling skew between internal and external clock. 0 - PLL disabled (a clock from REFCLK pin is sent instead) 1 - PLL enabled (enable this bit after memory clock synthesizer is properly programmed and stable)
SGR_CONFIG[25]	SDL	0	Read data sampling delay 0 - normal 1 - delayed This bit is OR-ed with CONFIG2[22] bit , so any of these two bits if set to one will delay sampling.
SGR_CONFIG[28]	BPP	0	This bit enables PLL controlling skew between internal and external clock. 0 - PLL enabled 1 - PLL disabled (delayed internal clock is sent to memories, but the skew is not controlled by the PLL)

SGR_CONFIG[30:29]	CLKD	0	Clock delay control 00 - 2 ns + 0 ns 01 - 2ns + 1ns 10 - 2ns + 2 ns 11 - 2 ns + 3 ns
SGR_CONFIG[31]	TR		A write only trigger bit. A "one" written to this bit initialize configuration cycle. The memory controller has to be in idle state to execute this cycle. (Turn off dram refresh, screen/CRT refresh, drawing engine has to be idle)

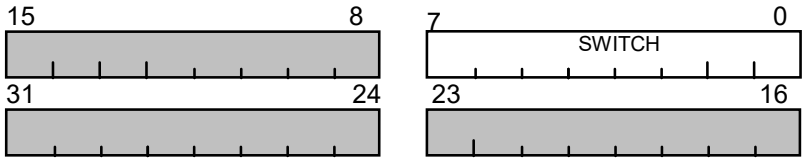
Typically the value in SGR_CONFIG[15:0] register should be set to:
0x1420 in VGA mode at 83 Mhz. 0x1020 in VGA mode at 60 Mhz.
0x1830 in Imagine mode at 100 Mhz. 0x1430 in Imagine mode at 83 Mhz.

5.3.5 Soft Switch Register {PCIB, (0x0028}

Name: SOFT_SW

Type: I/O space mapped read write

This register contains the data that was written to the external soft switch register. If no switches are present, this can be used as a general purpose register.

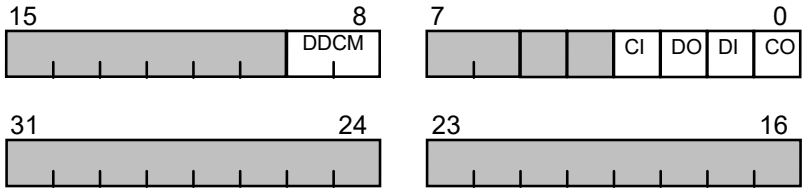


BITS	NAME	DEFAULT VALUE	DESCRIPTION
SOFT_SW[7:0]	SWITCH	0x00	This register contains the same data as the external soft switch register.

5.3.6 DDC Register {PCIB4, (0x002C)}

Name: DDC

Type: I/O mapped read write



Bits	Name	Value	Function
DDC[0]	C0		DDC CLK OUT. In DDC1 mode, this bit connects to the monitor vsync signal. In DDC2B mode, this bit connects to the SCL line.
		0x0	Forces clock signal low.
		0x1	Forces clock signal to "Z" External pull-up resistor will cause the clock to stay high if no other device is driving this signal low.
DDC[1]	DI		DDC Data Line Status (read only) (SDA). This bit shows actual value on the serial data line (DDC1 and/or DDC2B).
DDC[2]	DO		DDC2B Data OUT. (SDA) In DDC2B mode this bit connects to the Serial Data line. This bit is N/A in DDC1 mode.
		0x0	Forces data signal low.
		0x1	Forces data signal to "Z" External pull-up resistor will cause the data signal to stay high if no other device is driving this signal low.
DDC[3]	CI		DDC2B Clock Line Status (read only). (SCL) This bit shows actual value on the serial clock line (SCL) .
DDC[9:8]	DDCM		DDC Mode.
		0x0	DDC is disabled.
		0x1	DDC1 is enabled.
		0x2	DDC2B is enabled.
		0x3	DDC is disabled.

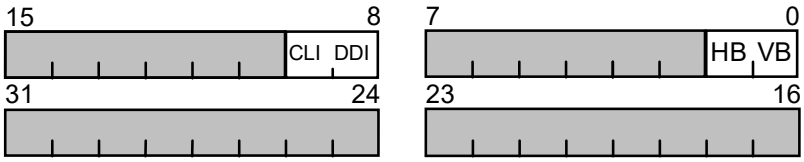
5.4 Global Interrupt Registers

5.4.1 Global Interrupt Register RBASE_I+(0x0000)

Name: GINTP

Type: Memory mapped read write

This register contains the concatenation of all the IMAGINE 128 interrupts. The drawing engine interrupts in this register are read only and must be cleared in their respective interrupt registers.



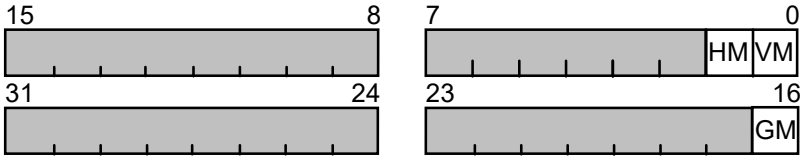
Bits	Name	Value	Function
GINTP[0]	VB_INT	0 1	Vertical blank count match has not occurred Vertical blank count match has occurred See also INT_VCNT register in CRT Registers section
GINTP[1]	HB_INT	0 1	Horizontal blank count match has not occurred Horizontal blank count match has occurred See also INT_HCNT register in CRT Registers section
GINTP[8]	DD_INT	0 1	Drawing engine operation not complete. Drawing engine operation is complete.
GINTP[9]	CL_INT	0 1	Drawing engine clip interrupt has not occurred. Drawing engine clip interrupt has occurred.

5.4.2 Global Interrupt Mask Register RBASE_I+(0x0004)

Name: GINTM

Type: Memory mapped read write

This register contains the interrupt mask bits for the "VB_MSK" and the "HB_MSK" interrupts.



Bits	Name	Value	Function
GINTM[0]	VB_MSK	 0 1	Vertical Blank Mask Vertical blank count interrupt disabled. (*Default*) Vertical blank count interrupt enabled.
GINTM[1]	HB_MSK	 0 1	Horizontal Blank Mask Horizontal blank count interrupt disabled. (*Default*) Horizontal blank count interrupt enabled.

GINTM[16]	GM		Global Interrupt mask
		0	Disable all interrupts. (*Default*)
		1	Enable un-masked interrupts

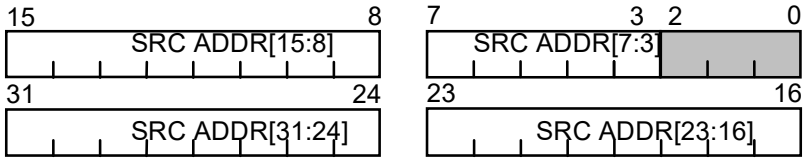
5.5 AGP DMA Registers

DMA registers are located in both I/O and RBASE_I space. The T2R IV is capable of generating DMA reads from system memory to local memory using AGP bus transactions.

5.5.1 DMA Source Address Register PCIB4 or RBASE_I + (0x00D0)

Name: DMA_SRC

Type: Memory or I/O mapped read write

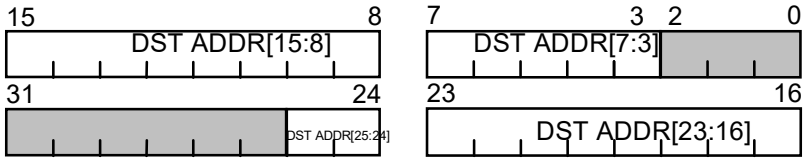


Bits	Name	Value	Function
DMA_SRC[2:0]	Reserved		Reserved
DMA_SRC[31:3]	SRC_ADDR		Source Address - address of 1st Qword in system memory.

5.5.2 DMA Destination Address Register PCIB4 or RBASE_I + (0x00D4)

Name: DMA_DST

Type: Memory or I/O mapped read write

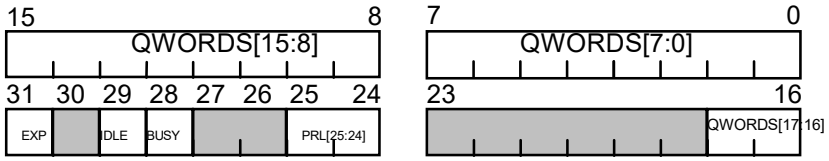


Bits	Name	Value	Function
DMA_DST[2:0]	Reserved		Reserved
DMA_DST[25:3]	DST_ADDR		Destination Address - address of 1st Qword in local memory.
DMA_DST[31:26]	Reserved		Reserved

5.5.3 DMA Command Register PCIB4 or RBASE_I + (0x00D8)

Name: DMA_CMD

Type: Memory or I/O mapped read write



Bits	Name	Value	Function
DMA_CMD[17:0]	QWORDS		Number of Qwords requested
DMA_CMD[23:18]	Reserved		Reserved
DMA_CMD[25:24]	PRL		Preferred Request Length for AGP transfers
		00	4 Qwords
		01	8 Qwords
		10	16 Qwords
		11	32 Qwords
			Note: A PRL = "00" is an illegal combination for 2x transfers. ("00" will be effectively promoted a "01")
DMA_CMD[27:26]	Reserved		Reserved
DMA_CMD[28]	DMAC_PIPELINE		DMAC_PIPELINE is **read only.**
		0	DMAC can accept another request.
		1	DMAC cannot accept another request.
DMA_CMD[29]	DMAC_STATUS		DMAC_STATUS is **read only**.
		0	DMAC is not idle
		1	DMAC is idle.
DMA_CMD[30]	Reserved		Reserved
DMA_CMD[31]	EXP		Expedite
			Note: Do not start a high priority DMA request if the DMAC is busy processing a low priority request.
		0	Low Priority
		1	High Priority

5.6 PCI Bus Master Registers

PCI Bus Master registers are located in both I/O and RBASE_I space. The T2R IV is capable of generating a specific PCI Bus Master Write to system memory of event status.

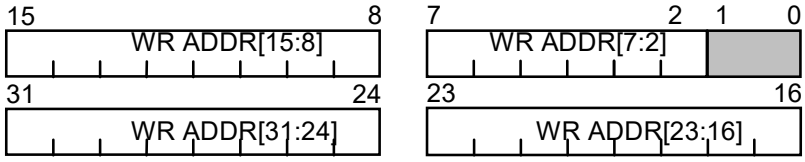
5.6.1 PCI Bus Master Write Address Register PCIB4 or RBASE_I + (0x00E0)

Name: PCI_BMWA

Type: Memory or I/O mapped read write

Description:

Address where Status Table is written.



Bits	Name	Value	Function
PCI_BMWA[1:0]	Reserved		Reserved
PCI_BMWA[31:2]	WR_ADDR		Write Address (system memory physical address)

Status Table: (Format of the Status DWord written into WR_ADDR)

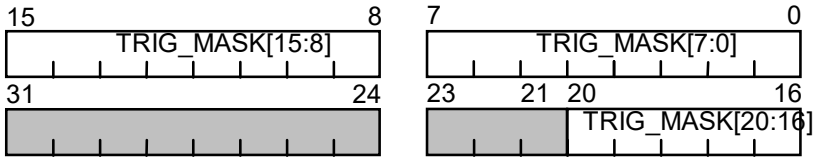
Bits	Status Name	Reference	Description
[0]	DEB	FLOW[0]	Drawing Engine is busy
[1]	MCB	FLOW[1]	Memory Controller is busy
[2]	CLP	FLOW[2]	Clipping on previous command
[3]	PRV	FLOW[3]	Previous Command still executing
[4]	RPB	FLOW[4]	Rendering Pipeline and/or Pixel Cache Busy
[5]	DEPF	BUSY[0]	Drawing Engine Command Pipeline is full.
[6]	DLPB	DL_ADR[30]	DLP is busy
[7]	DMAC_PIPELINE	DMA_CMD[28]	DMAC pipeline is full.
[8]	DMAC_STATUS	DMA_CMD[29]	DMAC is idle
[9]	HBINT	GINTP[1]	HB Interrupt occurred
[10]	VBINT	GINTP[0]	VB Interrupt occurred
[11]	VBLANK	DB_ADR[29]	Vertical Blanking time
[30:12]	Reserved	N/A	RESERVED
[31]	REPORTED	N/A	PCI Bus Master always writes a 1. Software must clear this bit in order to detect a new write.

5.6.2 PCI Bus Master Trigger Mask Register PCIB4 or RBASE_I + (0x00E4)

Name: PCI_BMTM

Type: Memory or I/O mapped read write

Description:
IMAGINE 128™'s PCI bus master can be triggered on separate rising or falling edge events. The appropriate mask bit must be cleared (reset to 0) in order for a bus master cycle to run for the given event. When an unmasked trigger event occurs, the bus master will write the **current** state of the following status bits to WR ADDR. (See **Status Table**)



Bits	Name	Value	Function
PCI_BMTM[20:0]	TRIG_MASK		Trigger Mask for PCI Bus Master
PCI_BMTM[31:21]	Reserved		Reserved

Trigger Mask Table:

Note: A rising edge mask means that an event trigger will occur when the referenced status bit transitions from a 0 value to a 1 value. Falling edge indicates a 1 to 0 transition.

Bits	Name	Value	Function	Status Table Bit
[0]	DMAC_PIPELINE_FE_MASK	1	DMAC Pipeline falling edge mask	7
[1]	DMAC_PIPELINE_RE_MASK	1	DMAC Pipeline rising edge mask	7
[2]	DMAC_STATUS_FE_MASK	1	DMAC status falling edge mask	8
[3]	DMAC_STATUS_RE_MASK	1	DMAC status rising edge mask	8
[4]	DEB_FE_MASK	1	Drawing Engine busy falling edge mask	0
[5]	DEB_RE_MASK	1	Drawing Engine busy rising edge mask	0
[6]	HBINT_RE_MASK	1	HB Interrupt rising edge mask	9
[7]	VBINT_RE_MASK	1	VB Interrupt rising edge mask	10
[8]	VBLANK_FE_MASK	1	V Blank rising edge mask	11
[9]	VBLANK_RE_MASK	1	V Blank falling edge mask	11
[10]	MCB_FE_MASK	1	Memory Controller busy falling edge mask	1
[11]	MCB_RE_MASK	1	Memory Controller busy rising edge mask	1
[12]	CLP_RE_MASK	1	Clipping bit rising edge mask	2
[13]	PRV_FE_MASK	1	Previous Command bit falling edge mask	3
[14]	PRV_RE_MASK	1	Previous Command bit rising edge mask	3
[15]	RPB_FE_MASK	1	Rendering Pipeline/Pixel Cache busy bit falling edge mask	4
[16]	RPB_RE_MASK	1	Rendering Pipeline/Pixel Cache busy bit rising edge mask	4
[17]	DEPF_FE_MASK	1	Drawing Engine Pipeline Full falling edge mask	5
[18]	DEPF_RE_MASK	1	Drawing Engine Pipeline Full rising edge mask	5
[19]	DLPB_FE_MASK	1	DLP busy falling edge mask	6
[20]	DLPB_RE_MASK	1	DLP busy rising edge mask	6
[31:21]	Reserved		Reserved	N/A

5.7 RAM DAC Registers

The following 16 registers provide access to a RAMDAC (external or internal). The first four registers maintain VGA LUT compatibility and are therefore also accessible from VGA I/O space *(See Section 5.2 "VGA DAC Shadowing")*. When accessing these registers as T2R IV addresses, the VGA snooping enable bit in the PCI command register has no effect.

The T2R IV implements a generic DAC 8-bit register interface. It does it through 16 DAC registers, which map to actual internal or external DAC registers through a register select (RS[3:0]) mechanism. External DACs are connected to the T2R IV peripheral bus (see Appendix B) and external DAC board designs will connect the peripheral bus address lines (PA[3:0]) to the DAC register selects (RS[3:0]). See the external DAC data sheet for register descriptions (indexed by RS[3:0]). For the T2R IV internal DAC, see Section ??.

The T2R IV DAC registers may be accessed through the following I/O or memory addresses:

DAC Register	RS[3:0]	T2R IV addresses	VGA address
DAC0	0000	PCIB4+80h, RBASEG+70h, RBASEG+00h	3C8h
DAC1	0001	PCIB4+84h, RBASEG+74h, RBASEG+04h	3C9h
DAC2	0010	PCIB4+88h, RBASEG+78h, RBASEG+08h	3C6h
DAC3	0011	PCIB4+8Ch, RBASEG+7Ch, RBASEG+0Ch	3C7h
DAC4	0100	PCIB4+90h, RBASEG+80h, RBASEG+10h	
DAC5	0101	PCIB4+94h, RBASEG+84h, RBASEG+14h	
DAC6	0110	PCIB4+98h, RBASEG+88h, RBASEG+18h	
DAC7	0111	PCIB4+9Ch, RBASEG+8Ch, RBASEG+1Ch	
DAC8	1000	PCIB4+A0h, RBASEG+90h	
DAC9	1001	PCIB4+A4h, RBASEG+94h	
DAC10	1010	PCIB4+A8h, RBASEG+98h	
DAC11	1011	PCIB4+ACh, RBASEG+9Ch	
DAC12	1100	PCIB4+B0h, RBASEG+A0h	
DAC13	1101	PCIB4+B4h, RBASEG+A4h	
DAC14	1110	PCIB4+B8h, RBASEG+A8h	
DAC15	1111	PCIB4+BCh, RBASEG+ACh	

5.8 CRT Registers

The CRT registers specify the CRT timing sent to the DAC. All of the horizontal parameters are in terms of the VCLK, whereas the vertical parameters are in terms of horizontal lines. VCLK is defined as the number of pixels delivered to the DAC in one clock cycle.

E.g:

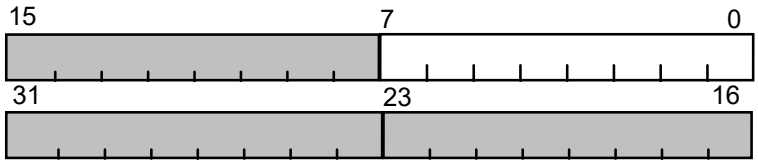
640 pixels at 16bpp with a 64 bit DAC bus width is 640/(64/16) = 160 VCLKs

Vertical Interrupt Count Register RBASE_G+(0x0020)

Name: INT_VCNT

Type: memory mapped read write.

This register defines the vertical field count. The contents of this register is compared with the field counter. When they match, a host interrupt is generated and the field counter is reset to zero. The range of this register is 0 to 255.



Bits	Name	Value	Function
INT_VCNT[7:0]	VCNT	0x00 0x01 ... 0xfe 0xff	Interrupt on every vertical field. Interrupt on every other vertical field. ... Interrupt on every 255 vertical fields. Interrupt on every 256 vertical fields.

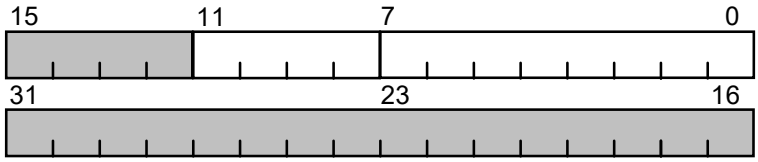
Horizontal Interrupt Count Register RBASE_G+(0x0024)

Name: INT_HCNT

Type: memory mapped read write.

This register defines a horizontal line before which an interrupt is generated.

A value 0x0ff in HCNT, for example, means that an interrupt will be generated at the beginning of the horizontal blank between line number 254 and 255.

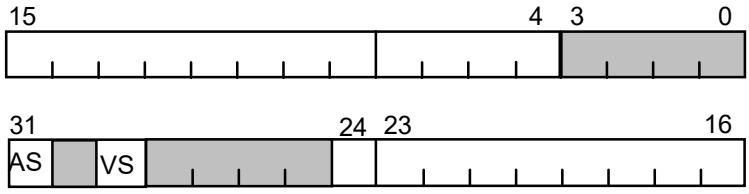


CRT Display Start Address (RBASE_G + 0x0028)

Name: DB_ADR

Type: Memory mapped read write

Display Start Address specifies the linear address of the first pixel to be displayed after vertical refresh (the upper left corner of screen). The start address is 16/128/256-byte aligned in SGRAM/WINRAM/WINRAM interleaved modes respectively.



Bits	Name	Value	Function
DB_ADR[24:0]	Start Address		Display start address. Display start address can be changed at any time, however, changes have no effect until following vertical blank interval. At the time a value in DB_ADDR[24:0] gets actually accepted, bit AS gets cleared to zero. Memory Type — Alignment WRAM — 128-byte aligned SGRAM — 16-byte aligned WRAM (interleaved) — 256-byte aligned
DB_ADR[29]	VS		Vertical Blank Status (read only)
		0	Vertical Blank is 0 (screen blanked)
		1	Vertical Blank is 1 (screen active)
DB_ADR[31]	AS		Display Address Status (read only) Any write into DB_ADDR[24:0] sets this bit to one. At the time of accepting a new value in DB_ADDR[24:0], the AS bit gets cleared.
		0	A write into DB_ADDR[24:0] has been synchronized.
		1	A write into DB_ADDR[24:0] hasn't been synchronized yet

To avoid a “display tearing effect” when switching the display start address, a new address is actually passed to CRT controller only once per frame during vertical blank period.

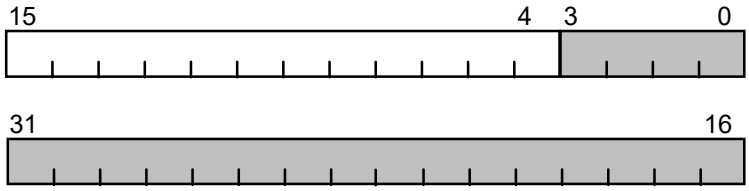
DB_ADDR[24:0] can be written to at any time, however, to avoid skipping frames, the address should be updated only if AS bit reads zero. For proper operation, the most significant byte of the address (bit[24]), has to be always updated along with the other address bits.

Display Buffer Pitch (RBASE_G + 0x002C)

Name: DB_PTCH

Type: Memory mapped read write

This register defines the display pitch of the Display buffer. It is the address offset between two vertically adjacent pixels. The display pitch is 16 byte aligned. *For WINDOW RAM align the pitch value to 128 bytes (256 bytes for interleaved configuration).*

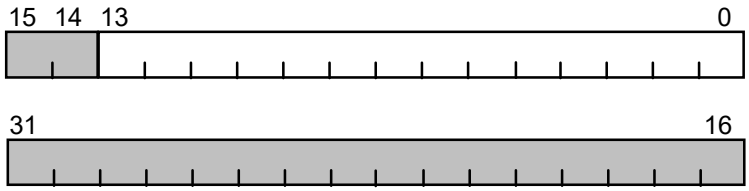


CRT Horizontal Active Line (RBASE_G + 0x0030)

Name: CRT_HAC

Type: Memory mapped read write

The horizontal active line register specifies the horizontal resolution in terms of CRT clock periods. The value in this register equals the number of CRT clocks (VCLK) during the active part of one line.

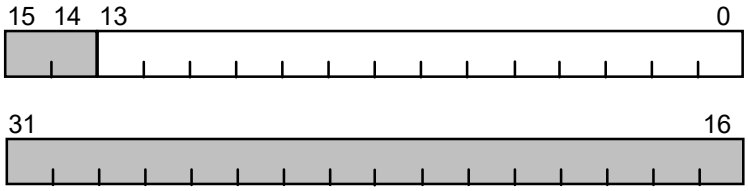


CRT Horizontal Blank Width (RBASE_G + 0x0034)

Name: CRT_HBL

Type: Memory mapped read write

This register specifies width of horizontal blank in number of CRT clocks (VCLK).

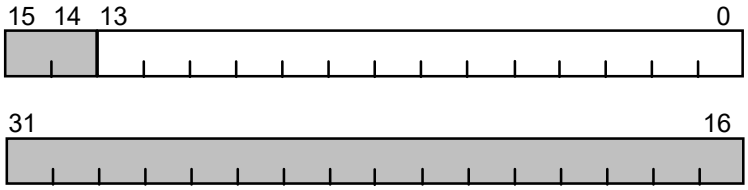


CRT Horizontal Front Porch Width (RBASE_G + 0x0038)

Name: CRT_HFP

Type: Memory mapped read write

This register specifies the width of horizontal front porch in number of CRT clocks (VCLK). The value of CRT_HFP is equal to the number of CRT clock periods between the beginning of horizontal blank and the following horizontal sync impulse. The smallest value is 0.

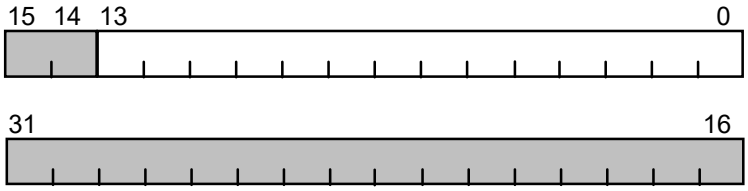


CRT Horizontal Sync Width (RBASE_G + 0x003C)

Name: CRT_HS

Type: Memory mapped read write

This register specifies the width of the horizontal sync impulse in number of CRT clock periods (VCLK) The smallest value for non-interlaced displays is 1.

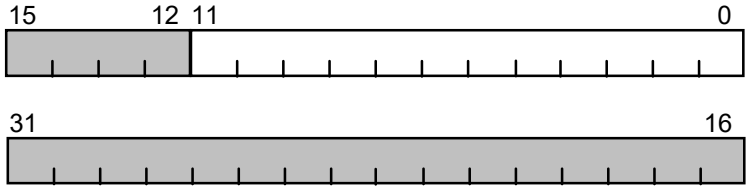


CRT Vertical Field Active (RBASE_G + 0x0040)

Name: CRT_VAC

Type: Memory mapped read write

Vertical Field Active register specifies the vertical resolution in number of lines. It is the number of displayed lines during one frame.

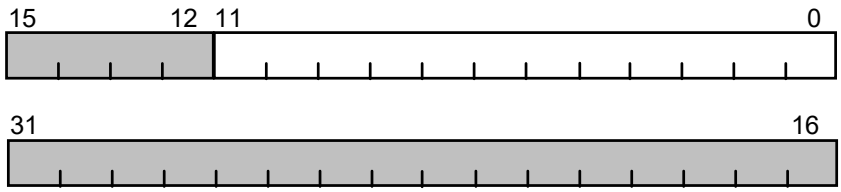


CRT Vertical Blank Width (RBASE_G + 0x0044)

Name: CRT_VBL

Type: Memory mapped read write

This register specifies the number of lines blanked during one frame.

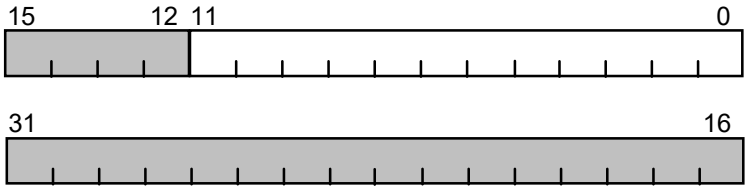


CRT Vertical Front Porch Width (RBASE_G + 0x0048)

Name: CRT_VFP

Type: Memory mapped read write

This register specifies the width of the vertical front porch. The units of this register are: number of lines .

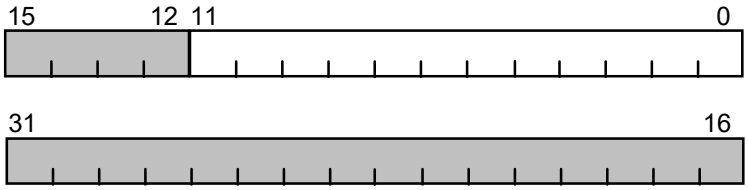


CRT Vertical Sync Width (RBASE_G + 0x004C)

Name: CRT_VS

Type: Memory mapped read write

This register specifies the width of the vertical sync impulse The units of this register are: number of lines .



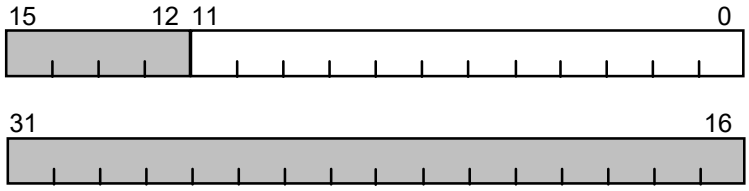
CRT Line Counter (RBASE_G + 0x0050)

Name: CRT_LCNT

Type: Memory mapped read only

This read only register shows the actual value of the display line counter.

A zero corresponds to beginning of a vertical blank period. A value greater than the number of lines in a vertical blank (see CRT_VBL register) indicates active portion of video, etc.

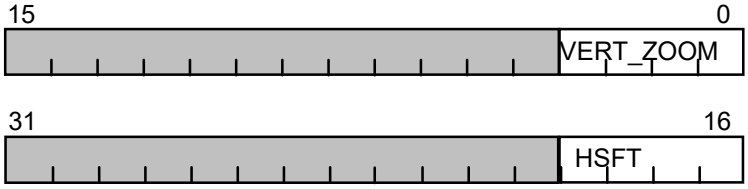


CRT Display Buffer Zoom Factor (RBASE_G + 0x0054)

Name: CRT_ZOOM

Type: Memory mapped read write

This register defines the vertical zoom and horizontal shift factors for the display output stage.



Bits	Name	Value	Function
CRT_ZOOM[3:0]	VERT_ZOOM[3:0]	0x0	No Zoom
		0x1	Zoom 2X
		0x2	Zoom 3X
		0x3	Zoom 4X
		0x4	Zoom 5X
		.....	.....
		0xd	Zoom 14X
		0xe	Zoom 15X
		0xf	Zoom 16X

Note: Vertical Zoom has no effect in line sequential stereo mode. See CRT_1CON[30].

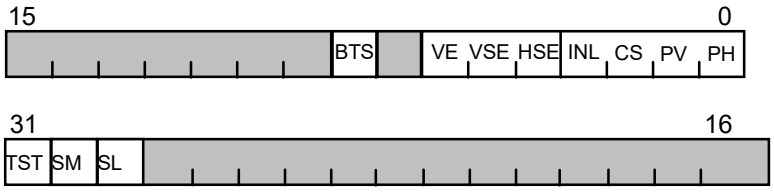
Bits	Name	Value	Function
CRT_ZOOM [19:16]	HSFT[3:0]	0x0	No Divide
		0x1	Divide By 2
		0x3	Divide by 4
		0x7	Divide by 8
		0xF	Divide by 16

HSFT[3:0] bits cause simple right shifting of all horizontal sync parameters. Using this feature for zoom purpose may require (in some applications) dividing VCLK by the same factor.

CRT Configuration Register 1 (RBASE_G + 0x0058)

Name: CRT_1CON

Type: Memory mapped read write

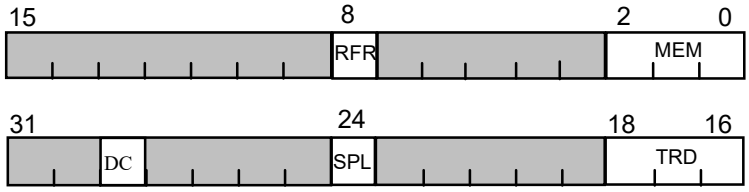


Bits	Name	Value	Function
CRT_1CON[0]	PH		Horizontal Sync Polarity.
		0	Negative horizontal sync.
		1	Positive horizontal sync.
CRT_1CON[1]	PV		Vertical Sync Polarity.
		0	Negative vertical sync.
		1	Positive vertical sync.
CRT_1CON[2]	CS		Composite Sync Enable. The polarity of composite sync is **always negative**. If composite sync is enabled, then Vertical Sync becomes a one clock wide impulse (always negative).
		0	Separate HSYNC and VSYNC are generated
		1	Composite sync is output on HSYNC
CRT_1CON[3]	INL		This bit must be set to 0.
		0	
CRT_1CON[4]	HSE		Horizontal Sync Enable
		0	HSYNC disabled (HSYNC = 1)
		1	Normal HSYNC generation
CRT_1CON[5]	VSE		Vertical Sync Enable
		0	VSYNC disabled (VSYNC = 1)
		1	Normal VSYNC generation
CRT_1CON[6]	VE		Video Enable
		0	BLANK disabled (BLANK= 0)
		1	Normal BLANK generation
CRT_1CON[8]	BTS		SCLK Pin direction
		0	SCLK is an input to IMAGINE 128®
		1	SCLK is an output from IMAGINE 128®: The CRT clock (VCLK) is AND-ed with an internal blank signal and then output to SCLK pin to be used as WINDOW RAM shift clock.
CRT_1CON[29]	SL		
		0	EMGNT# pin used as left/right line indicator.
		1	This setting should be used only with SM bit=1.
CRT_1CON[30]	SM	0	Normal CRT operation
		1	Enables line sequential stereo mode.
CRT_1CON[31]	TST	0	CRT Test Mode Bit Normal Operation.
		1	Test Mode.

CRT Configuration Register 2 (RBASE_G + 0x005C)

Name: CRT_2CON

Type: Memory mapped read write



Bits	Name	Value	Function
CRT_2CON[2:0]	MEM[2:0]		Window RAM configuration:
		0x0	write 000 for non-interleaved config.
		0x4	write 100 for interleaved configuration
		all other	Reserved
CRT_2CON[8]	RFR		Enables screen refresh (transfers to serial registers of WINRAM buffer or to CRT FIFO in SGRAM mode)
		0	- refresh disabled
		1	- refresh enabled
CRT_2CON[18:16]	TRD		"Memory to register" transfer delay
		0x0	no delay
		0x1	1 VCLK delay
		0x2	2 VCLKs delay
		0x3	3 VCLKs delay
		0x4	4 VCLKs delay
		0x5	5 VCLKs delay
		0x6	6 VCLKs delay
		0x7	7 VCLKs delay

continued..

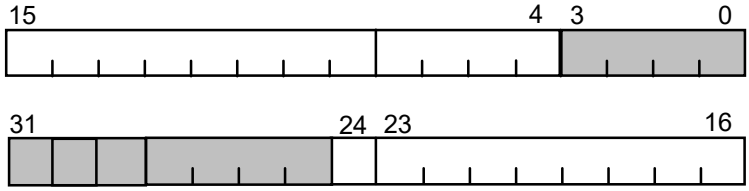
Bits	Name	Value	Function
CRT_2CON[24]	SPL	0	This bit should be 0.
CRT_2CON[29]	DC	0x0 0x1	Display Configuration Dual ported memories (WINRAM) Single ported memories(SGRAM)

CRT Display Start Address2 (RBASE_G + 0x0060)

Name: DB_ADR2

Type: Memory mapped read write

Display Start Address2 specifies the linear address of the secondary display buffer used only in stereo mode. Value in this register is valid only if SM bit in CRT_1CON register is set to one. The start address is 16/128/256 byte aligned in SGRAM/WINRAM/WINRAM interleaved modes respectively.



Bits	Name	Value	Function
DB_ADR2[24:0]	Start Address2		Display start address.

If stereo mode is set, the first line after vertical blank and then every other line will be displayed from the primary display buffer defined by DB_ADR register. Second line after vertical blank and then every other line will be displayed from the secondary display buffer defined by DB_ADR2. For example, the CRTC will output line 0 from DB_ADR, line 0 from DB_ADR2, line 1 from DB_ADR, line 1 from DB_ADR2, etc.

5.7 Memory Windows™ Configuration Registers

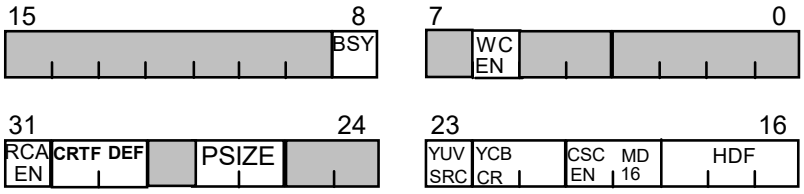
The Memory Windows™ configuration registers set the size, control, system memory location and local buffer offset of the Memory Windows™. There are two identical sets of these memory mapped registers. The address of these registers is set by the value written to the RBASE_W register.

5.7.1 Memory Window Control Register

window zero **RBASE_W + (0x0000)**
window one **RBASE_W + (0x0028), PCIB4 + (0x0040)**

Name: MW0_CTRL, MW1_CTRL,

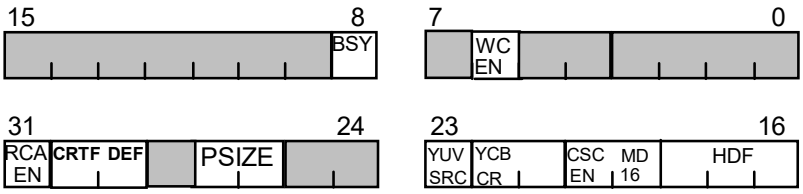
Type: Memory mapped read write



KEVIN TO SUPPLY INTRODUCTION!

Memory Window Control Register (Continued)

window zero **RBASE_W + (0x0000)**
window one **RBASE_W + (0x0028) , PCIB4 + (0x0040)**

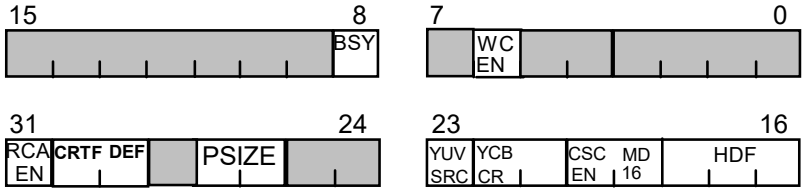


Bits	Name	Value	Function
MWX_CTRL[5:0]	Reserved	0x0	Reserved
MWX_CTRL[6]	WCEN	0x0 0x1	Enable write caching Disable write caching
MWX_CTRL[7]	Reserved	0x0	Reserved
MWX_CTRL[8]	BSY	 0x0 0x1	Memory Window Busy **(read only)** Memory Window is busy Memory Window is not busy

MWX_CTRL[16]	HDF[0]	0x0 0x1	Bit ordering is not modified Swaps bits within each byte
MWX_CTRL[17]	HDF[1]	0x0 0x1	Byte ordering is not modified Swap bytes within each word
MWX_CTRL[18]	HDF[2]	0x0 0x1	Word ordering is not modified Swap words within each DWORD
MWX_CTRL[19]	MD16	 0x0 0x1	16 Bits per pixel mode 0=5RED,5GREEN,5BLUE 1=5RED,6GREEN,5BLUE
MWX_CTRL[20]	CSC_EN	 0x0 0x1	Color space conversion enable Disable color space conversion Enable color space conversion
MWX_CTRL[21]	Reserved	0x0	Reserved
MWX_CTRL[22]	YCBCR	 0X0 0X1	YCBCR Selector (color space format) YCBCR YUV
MWX_CTRL[23]	YUV_SRC	 0x0 0x1	YUV source format. YUV 422. (Default) YUV 444.

Memory Window Control register (Continued)

window zero RBASE_W + (0x0000)
window one RBASE_W + (0x0028) , PCIB4 + (0x0040)



Bits	Name	Value	Function
MWX_CTRL[25:24]	Reserved	 0x0	Reserved.
MWX_CTRL[27:26]	PSIZE[1:0]	 0x0 0x1 0x2 0x3	Destination pixel size, number of bits per pixel Eight bits per pixel Sixteen bits per pixel Thirty-two bits per pixel Reserved
MWX_CTRL[28]	Reserved	0x0	Reserved
MWX_CTRL[29]	DEF	 0 1	Drawing Engine Flush Cache on 2D/3D Trigger Access Enable (default). Disable.
MWX_CTRL[30]	DSE	 0 1	CRT Flush on Vertical Blank Interrupt Enable(default). Disable.

MWX_CTRL[31]	RCA_EN		Read cache enable.
		0	Enable read caching. (Default)
		1	Disable read caching.

5.7.2 Memory Windows Address Registers

window zero RBASE_W + (0x0004)
window one RBASE_W + (0x002C) , PCIB4 + (0x0044)

Name: MW0_AD, MW1_AD

Type: Memory mapped read write

The address contained in this register specifies the start of system memory space that is mapped to IMAGINE 128® local memory. The granularity of this register is determined by the memory window size that is specified by the SIZE field which is described in section 4.8.3. In the case of the smallest memory window which is 4 Kb, MWX_AD[31:12] are decoded for the memory window. For an 8 KB memory window, MWX_AD[31:13] are decoded. This pattern continues to the maximum window size of 32 MB, where MWX_AD[31:25] are decoded.

The system BIOS is responsible for allocation of all PCI system resources. The allocation process is facilitated by the PCI base registers described in section 3. IMAGINE 128® will request 4, 8, 16, or 32 megabytes of memory space for each of the memory windows based on the configuration jumper settings. The system will then allocate memory to IMAGINE 128® by writing the starting address of the memory block for memory window 0 to base address register 0 and the starting address of the memory block for memory window 1 to base address register 1. These values will be shadowed to MW0_AD and MW1_AD respectively. Software can then program each of the memory windows to reside anywhere within the allocated memory block. The size of each window can then be set from a minimum of 4 kilobytes to a maximum determined by the amount of memory allocated by the PCI system

Bits	Name	Value	Function
MWX_AD[31:12]	ADDRESS		Address decode value

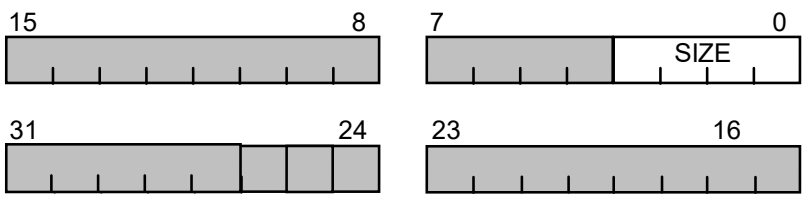
5.7.3 Memory Window Size

window zero RBASE_W + (0x0008)
window one RBASE_W + (0x0030) , PCIB4 + (0x0048)

Name:MW0_SZ, MW1_SZ

Type: Memory mapped read write

This register defines the size of a Memory window. The size of a memory window should be set equal to or less than the amount of memory allocated by the PCI system.



Bits	Name	Value	Function
MWX_SZ_DIB[3:0]	SIZE	0x0	Size equals 4K
		0x1	Size equals 8K
		0x2	Size equals 16K
		...	...
		0xD	Size equals 32MB
		0xE	RESERVED
		0xF	RESERVED

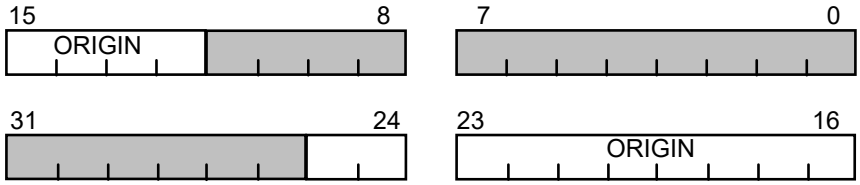
5.7.4 Memory Window Origin

window zero RBASE_W + (0x0010 or 0x0014)
window one RBASE_W + (0x0038 or 0x003C) , PCIB4 + (0x0050 or 0x0054)

Name: MW0_ORG,MW1_ORG

Type: Memory mapped read write

This register specifies the starting address of a memory window within a selected buffer or buffers. The number of valid bits in this register depends on the size of the memory window. As the size of the memory window increases, fewer bits will be taken from this register and more will be taken from the host address. In the case of a 4 kilobyte memory window, the address presented to the memory controller will be a concatenation of bits[24:12] from this register with the lower 12 bits coming from the host. An 8 kilobyte memory window will cause bits[24:13] to be taken from this register with the lower 13 bits coming from the host. A 32 megabyte memory window would not use any bits from this register.



Bits	Name	Value	Function
MWX_ORG[25:12]			Upper order memory window address bits
	MWX_ORG[24:12]		Upper order address for 4 kilobyte window
	MWX_ORG[24:13]		Upper order address for 8 kilobyte window
	MWX_ORG[24:14]		Upper order address for 16 kilobyte window
	•		•
	•		•
	•		•
	MWX_ORG[24:23]		Upper order address for 8 megabyte window
	MWX_ORG[24]		Upper order address for 16 megabyte window

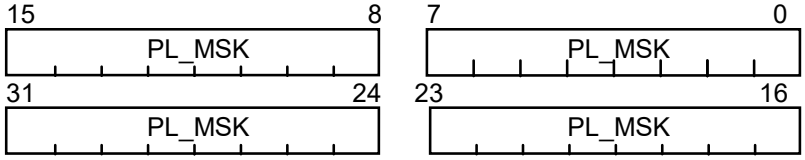
5.7.5 Memory Window Plane Mask

window zero RBASE_W + (0x0024)
window one RBASE_W + (0x004C) , PCIB4 + (0x0064)

Name: MW0_MASK, MW1_MASK,

Type: Memory mapped read write

The MASK register allows for selective writing of pixel planes to the Display Buffer. Using this feature, bit maps may be partitioned into logical groups for selective updating. A bit value of zero disables writing to a specific plane while a one enables writing to that plane. In eight bit per pixel mode, the plane mask in MASK[7:0] should be replicated four times into the entire register. In sixteen bits per pixel mode, MASK[15:0] should be replicated two times into the entire register.



Bits	Name	Value	Function
MWX_MASK [31:0]	PL_MSK		Plane Masking Register

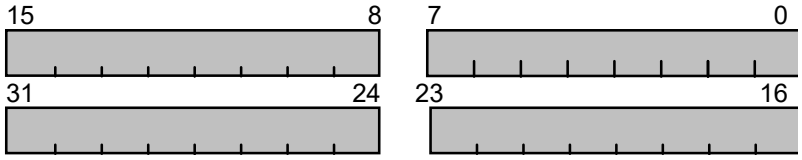
5.7.7 Memory Window Flush Trigger

RBASE_W + (0x0054)

Name: MWC_FLSH

Type: Memory mapped write, always reads back as zero

This register when written triggers a flush of the memory windows cache. The value of the data written to this register is a don't care.



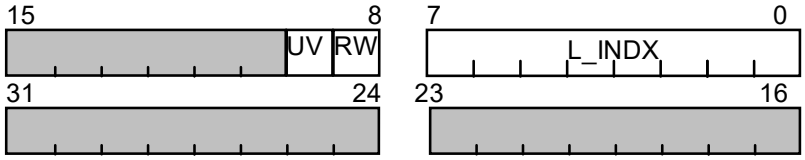
5.7.8 YUV LUT Index Register

RBASE_W + (0x0058)

Name: YUV_ADR

Type: Memory mapped read, write

This is a test register is used to indirectly address the YUV LUT.



Bits	Name	Value	Function
YUV_ADR[7:0]	L_INDX	0x00 to 0xff	Address pointer into the color space conversion LUT.
YUV_ADR[8]	RE	 0x0 0x1	LUT read enable bit. 0 = Normal operation. (Default) 1 = Read back mode.

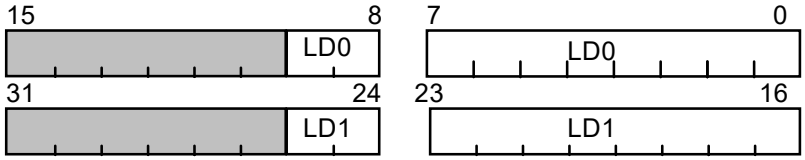
YUV_ADR[9]	UV	0x0 0x1	LUT select. 0 = Select V LUT. (Default) 1 = Select U LUT.

5.7.9 YUV LUT Data Register

RBASE_W + (0x005C)

Name: YUV_DAT
Type: Memory mapped read only

This is a test register. It returns the value of the LUT location pointed to by YUV_ADR.



Bits	Name	Value	Function
YUV_DAT[9:0]	LD0		LUT read data L_INDX.
YUV_ADR[25:16]	LD1		LUT read data L_INDX + 1.

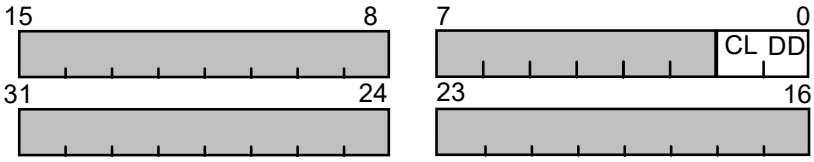
5.8 Drawing Engine Command and Parameter Registers

The following group of registers are memory mapped and are used to issue commands and parameters to the drawing engine. The address of these registers is offset by RBASE_D.

5.8.1 Interrupt Register RBASE_D + (0x0000)

Name: INTP

Type: Memory mapped read write

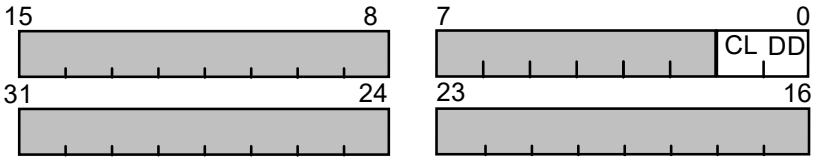


Bits	Name	Value	Function
INTP[0]	DD_INT	0 1	Drawing operation not completed Drawing operation is completed
INTP[1]	CL_INT	0 1	Clip interrupt has not occurred Clip interrupt has occurred
INTP[31:2]	Reserved	0x0	Reserved

5.8.2 Interrupt Mask Register RBASE_D + (0x004)

Name: INTM

Type: memory mapped read write

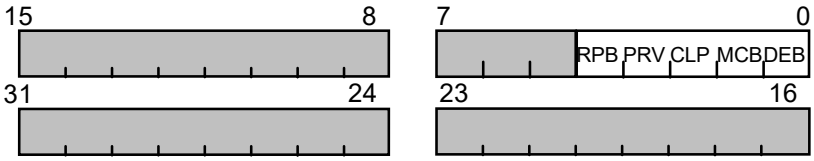


Bits	Name	Value	Function
INTM[0]	DD_MSK	0 1	Drawing completed interrupt disabled Drawing completed interrupt enabled
INTM[1]	CL_MSK	0 1	Clip interrupt disabled Clip interrupt enabled
INTM[31:2]	Reserved	0x0	Reserved

5.8.3 Flow Control Register RBASE_D +(0x0008)

Name: FLOW

Type: read only



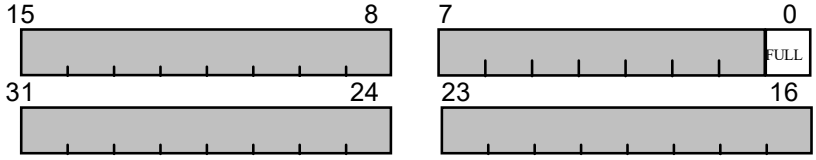
Bits	Name	Value	Function
FLOW[0]	DEB	0 1	Drawing engine is not busy Drawing engine is busy
FLOW[1]	MCB	0 1	Memory controller is not busy Memory controller is busy
FLOW[2]	CLP	0 1	No clipping was applied to previous drawing command Clipping was applied to previous drawing command
FLOW[3]	PRV	 0 1	Previous Command Busy The previously triggered command has completed The previously triggered command is still executing
FLOW[4]	RPB	0 1	Renderer Pipeline and Pixel Cache both not busy. Renderer Pipeline and/or Pixel Cache busy. **Note:** The Drawing engine will complete before the renderer and pixel cache complete. It is possible that the memory controller may be read as not busy before the pipeline and cache are flushed.

Note: Drawing Engine Busy (DEB) and Renderer Pipeline Busy (RPB) are independent busy status bits.

5.8.4 BUSY Register RBASE_D +(0x000C)

Name: BUSY

Type: memory mapped read only



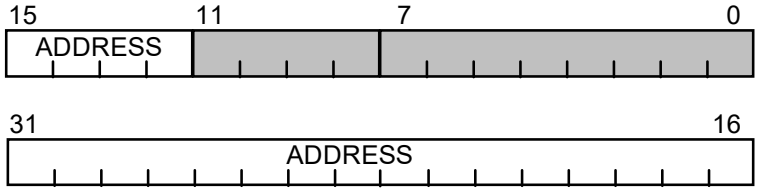
Bits	Name	Value	Function
BUSY[0]	FULL	0	IMAGINE 128³ is ready to accept a new command. *Command pipeline not full.*
		1	IMAGINE 128³ is not ready to accept a new command. *Command pipeline is full.*

5.8.5 XY Window Address RBASE _D + (0x0010)

Name: XYW_AD

Type: Memory mapped read write

This register defines the system address range that the host interface will decode for an X-Y memory window. Before programming this register, the EXA bit in CONFIG1 should be set to 0, preventing spurious X-Y window decode.

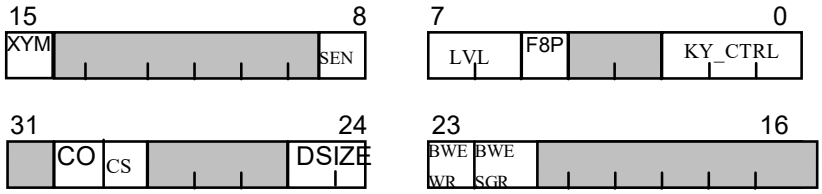


Bits	Name	Value	Function
XYW_AD[31:12]	ADDRESS		Address decode value

5.8.6 Buffer Control Register RBASE _D + (0x0020)

Name: BUF_CTRL

Type: Memory mapped read write



This register contains the buffer control bits for the drawing engine.

The source enable bits (SEN) select whether SilverHammer's internal Drawing engine cache is selected, or if the on board memory is selected.

Cache On (CO) is meaningful only when the source data is to be read from IMAGINE 128's internal 2D data cache. Setting CO to 1 indicates that the 2D cache always will contain valid data. This is useful when software partitions the cache into two halves, guaranteeing that one half of the cache contains valid data. Resetting CO to O is done by software to indicate it is no longer using the 2D cache.

The Destination Pixel Size bits define the destination pixel depth and format for the current drawing operation. The source pixel size and format is 32bpp, 8888 format and all drawing operations will be performed with this format. Right before alpha blending, z compare and ROP's, the pixel will be re-formatted to the specified destination pixel size and format. Texture cache formats are defined separately in the TEX_CTRL register, but will be expanded to 32bpp, 8888 format before any operation is performed on the texels. This preserves precision during pixel and texel operations.

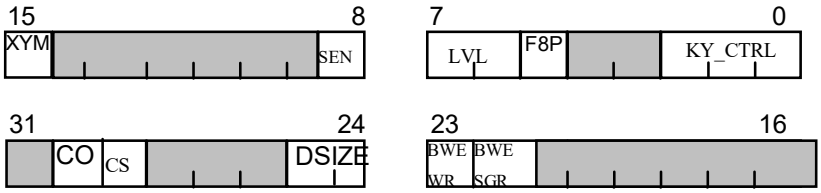
The Key Control field is used in conjunction with the DE_KEY register to set up a 2D transparency operation. 3D keying is selected in the 3D_CTRL register and is independent of 2D color keying. 2D keying is not reliable if doing a 3D operation involving texture-mapping, specular, fog, or texture filtering. 3D keying should be used only under these circumstances.

The Cache Select (CS) bit is used to select host bus writing access to either the XY windows or the user programmable Fog Tables.

Buffer Control Register RBASE _D + (0x0020) (Continued)

Name: BUF_CTRL

Type: Memory mapped read write

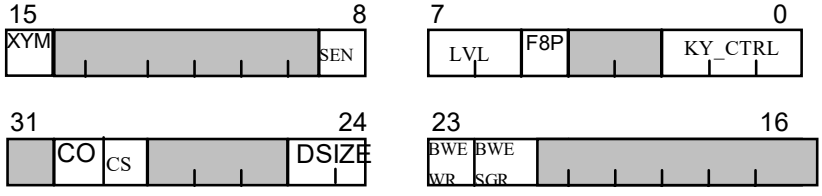


Bits	Name	Value	Function
BUF_CTRL[2:0]	KY_CTRL		Key Control
		0xx	No Keying
		100	Source Keying, mask on key color
		101	Destination Keying, mask on key color
		110	Source Keying, mask on NOT key color
		111	Destination Keying, mask on NOT key color
BUF_CTRL[4:3]	Reserved	0	Reserved
BUF_CTRL[5]	F8P		Force 8 Page Mode.
		0	Drawing Engine runs 8 & 16 page accesses
		1	Drawing Engine Runs 8 page access only
BUF_CTRL[7:6]	LVL		Line Vertical Limit This bit sets the limit to the number of vertical requests which the pixel cache can make:
		0x0	0 = no limit (4 pages)
		0x1	1 = 1 page
		0x2	2 = 2 pages
		0x3	3 = 3 pages.
BUF_CTRL[8]	SEN		Source Read Enable.
		0x0	Enable reads from external memory.
		0x1	Enable reads from IMAGINE 128® cache.
BUF_CTRL[14:9]	Reserved	0x0	Reserved
BUF_CTRL[15]	XYM		Origin Mode
		0x0	Normal linear origin mode
		0x1	XY origin mode
BUF_CTRL[21:16]	Reserved	0x0	Reserved
BUF_CTRL[22]	BWESGR		Block Write Enable (For SGRAM)
		0x0	Block writes disabled
		0x1	Block writes enabled
BUF_CTRL[23]	BWEWR		Block Write Enable (For WRAM)
		0x0	Block writes disabled
		0x1	Block writes enabled
BUF_CTRL[25:24]	DSIZE[1:0]		Destination Data Pixel Size and Format - Number of Bits per Pixel.
		0x0	8 bits per pixel - 332 Format (Blue Channel)
		0x1	16 bits per pixel - 1555 Format
		0x2	32 bits per pixel - 8888 Format
		0x3	16 bits per pixel - 565 Format

Buffer Control Register RBASE _D + (0x0020) (Continued)

Name: BUF_CTRL

Type: Memory mapped read write



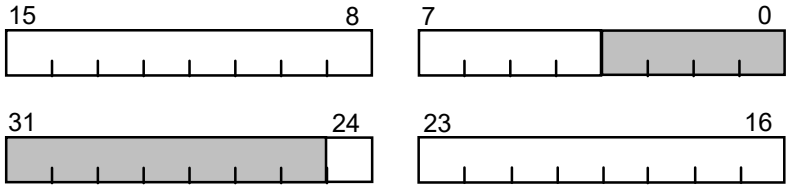
BUF_CTRL[28:26]	Reserved	0x0	Reserved
BUF_CTRL[29]	CS		Cache Select
		0	XY Windows
		1	User Programmable Fog Tables
BUF_CTRL[30]	CO		Cache On **(Needs Clarification!!!)** This bit should be set when source data is read from the cache and the cache contains the appropriate data. This bit is **not reset** by the drawing engine after a command completes, therefore the cache is always assumed to have valid data when CO is set. This setting is useful when the cache is partitioned into halves by software.
		0	Cache has not yet been written with the correct data
		1	Cache contains appropriate data for the next command
BUF_CTRL[31]	Reserved	0x0	Reserved

5.8.7 Drawing Engine Source Origin RBASE_D + (0x0028)

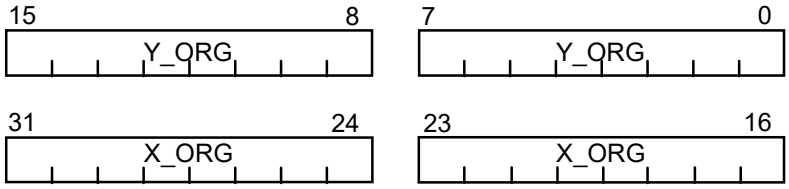
Name: DE_SORG

Type: Memory mapped read write

This register defines the linear or XY address offset into the selected buffer or buffers for drawing engine read cycles. In linear mode this is a 128 bit word address when accessing the Display Buffers. In XY mode this is a pixel address offset.



Bits	Name	Value	Function
DE_SORG[24:4]	SORGL		Linear origin mode



Bits	Name	Value	Function
DE_SORG[31:0]	SORGXY		XY origin mode

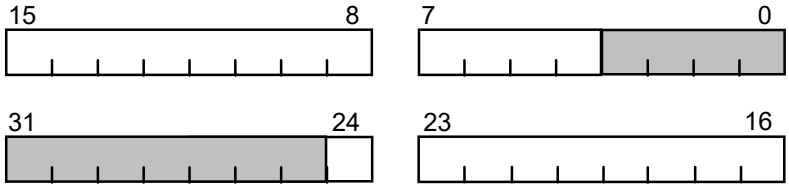
Note: Do we support this XY mode anymore? Jim

5.8.8 Drawing Engine Destination Origin RBASE_D + (0x002C)

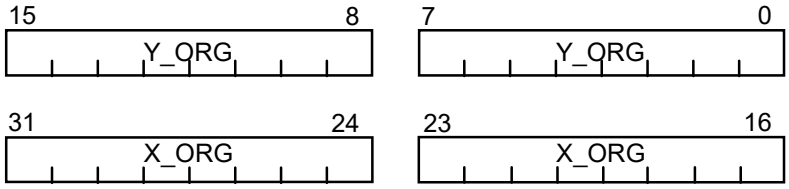
Name: DE_DORG

Type: Memory mapped read write

This register defines the linear or XY address offset into the selected buffer or buffers for drawing engine write cycles. In linear mode this is a 128 bit word address when accessing the Display Buffers. In XY mode this is a pixel address offset.



Bits	Name	Value	Function
DE_DORG[24:4]	DORGL		Linear origin mode.



Bits	Name	Value	Function
DE_DORG[31:0]	DORGXY		XY origin mode.

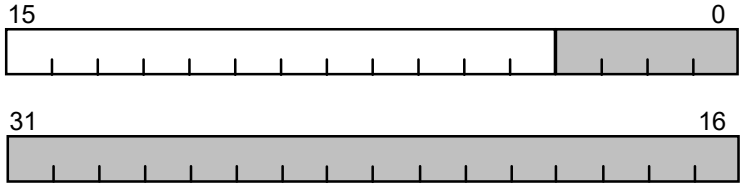
Note: Do we support this XY mode anymore? Jim

5.8.9 DE Texture Pitch RBASE_D + (0x0038)

Name: DE_TPTCH

Type: Memory mapped read write

This register defines the pitch of the texture map in LOD0 in bytes. All subsequent texture maps have a pitch equal to the previous texture/2, with a minimum size of 1. It is aligned to a 16 byte boundary.

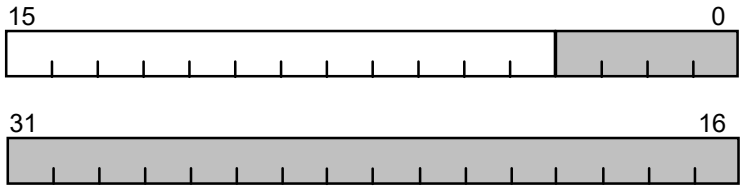


5.8.10DE Z Buffer Pitch RBASE_D + (0x003C)

Name: DE_ZPTCH

Type: Memory mapped read write

This register defines the pitch of the Z buffer in bytes. It is aligned to a 16 byte boundary.

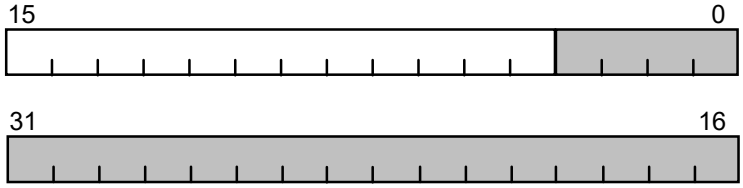


5.8.11 DE Source Pitch RBASE_D + (0x0040)

Name: DE_SPTCH

Type: Memory mapped read write

This register defines the pitch of the source bit map in bytes. It is aligned to a 16 byte boundary.

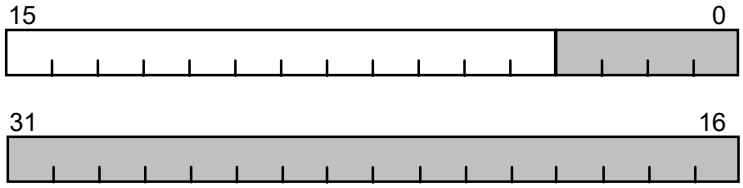


5.8.12 DE Destination Pitch RBASE_D + (0x0044)

Name: DE_DPTCH

Type: Memory mapped read write

This register defines the pitch of the destination bit map in bytes. It is aligned to a 16 byte boundary.

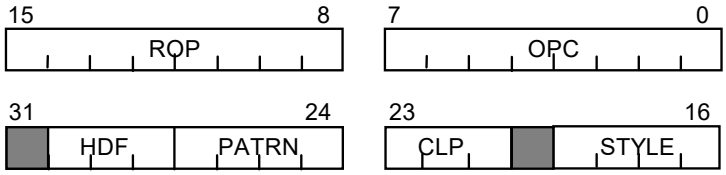


5.8.13 Command Register RBASE_D +(0x0048)

Name: CMD

Type: Memory mapped read write

The drawing engine command register can be accessed as a single 32 bit register or individual fields can be accessed in their own register space as described in the following pages.



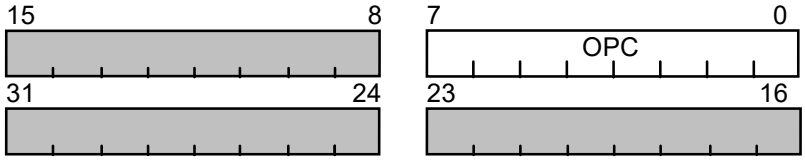
Bits	Name	Function
CMD[7:0]	OPC	Drawing command opcode
CMD[15:8]	ROP	Drawing Raster or logical operation
CMD[19:16]	STYLE	Solid, Transparency, Stipple control
CMD[20]	Reserved	Reserved
CMD[23:21]	CLP	Clipping control
CMD[27:24]	PATRN	No last, pat reset, area pattern, and pattern cache enable.
CMD[30:28]	HDF	Host Data Format

5.8.14

5.8.15 Command Opcode RBASE_D +(0x0050)

Name: CMD_OPC

Type: Memory mapped read write



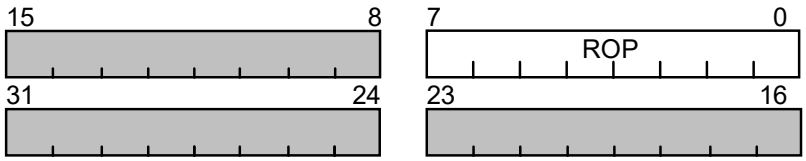
Value	Name	Function
0x00	NOOP	Transfer parameters no drawing
0x01	BITBLT	BIT BLT command
0x02	LINE	Line command
0x03	ELINE	Error Line
0x04	Reserved	Reserved
0x05	PLINE	Poly line command.
0x06	RXFER	Host X-Y window read image transfer
0x07	WXFER	Host X-Y window write image transfer
0x08	LINE_3D	3D Line command, Texture mapped, Z buffer, Gouraud shaded, setup
0x09	TRIAN_3D	3D Triangle Command, Full floating point setup, Texture Mapped, Perspective Corrected, Z buffered, Gouraud Shaded.
0x0A	TEX_INV	Invalidate Texture Cache
0x0B	LD_PAL	Palette Load Command
0x0C---0xFF	Reserved	For future expansion (no action taken)

5.8.16 Command Raster Operation RBASE_D +(0x0054)

Name: CMD_ROP

Type: Memory mapped read write

The ROP field sets the logical operation for all drawing commands.

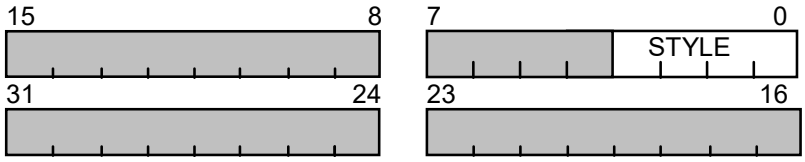


Value	Name	Function
0x00	clear	0 (zero)
0x01	nor	(NOT source) AND (NOT destination)
0x02	andInverted	(NOT source) AND destination
0x03	copyInverted	(NOT source)
0x04	andReverse	source AND (NOT destination)
0x05	invert	(NOT destination)
0x06	xor	source XOR destination
0x07	nand	(NOT source) OR (NOT Destination)
0x08	and	source AND destination
0x09	equiv	(NOT source) XOR destination
0x0A	noop	Destination
0x0B	orInverted	(NOT source) OR destination
0x0C	copy	source
0x0D	orReverse	source OR (NOT destination)
0x0E	or	source OR destination
0x0F	set	1 (one)
0x10 – 0xFF	RESERVED	reserved for future expansion

5.8.17 Line/Fill Style Register RBASE_D +(0x0058)

Name: CMD_STYLE

Type: Memory mapped read write

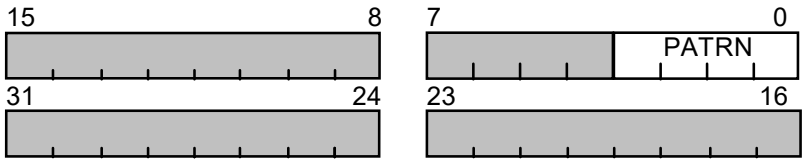


Bits	Name	Value	Function
CMD_STYLE[0]	SOLID		Solid When SOLID is set to one the source of the data for all operations is the FORE register.
CMD_STYLE[1]	TRNSP		Transparency When TRNSP is set to one during a line , stippled bit blt, stippled write transfer or stippled triangles a background condition will cause the destination data to be unmodified. Transparency overrides solid.
CMD_STYLE[3:2]	STPLE	 0x0 0x1 0x2 0x3	Stipple No stipple Reserved Packed stipple (padded to 32 bit boundary) Packed stipple (padded to 8 bit boundary)
CMD_STYLE[4]	Reserved	0x0	Reserved.

5.8.18 Patterning Register RBASE_D +(0x005C)

Name: CMD_PATRN

Type: Memory mapped read write

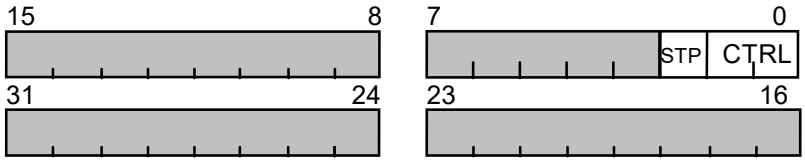


Bits	Name	Function
CMD_PATRN[1:0]	APAT[1:0]	Area Pattern When APAT is set to two it forces the source area pattern to be a 32x32 screen locked pattern.
	0x0	Area pattern mode is off.
	0x1	Area pattern mode equals 8x8.
	0x2	Area pattern mode equals 32x32.
	0x3	RESERVED
CMD_PATRN[2]	NLST	No last pixel When NLST is set to one during a line operation it forces the last pixel in the line not to be drawn.
	0	Draw the last pixel in a line
	1	Do not draw the last pixel in a line
CMD_PATRN[3]	PRST	Pattern reset bit When PRST is set to a one during a line operation, the pattern pointers will be reset before each line is started.
	0	Don't reset pattern pointers at the start of each line
	1	Reset pattern pointers at the start of each line

5.8.19 Clipping Control Register RBASE_D +(0x0060)

Name: CMD_CLP

Type: Memory mapped read write



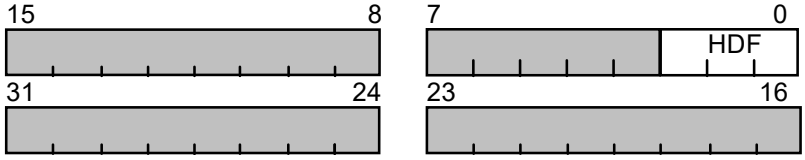
Bits	Name	Value	Function
CMD_CLP[1:0]	CCTRL[1:0]	0x0	No clipping
		0x1	No clipping
		0x2	Draw inside rectangle
		0x3	Draw outside rectangle
CMD_CLP[2]	CSTOP	0	Don't stop on clip boundary
		1	Stop on clip boundary

5.8.20 Host Data Format Register RBASE_D +(0x0064)

Name: CMD_HDF

Type: Memory mapped read write

This register controls how host data is passed to or from the drawing engine cache. It doesn't affect drawing engine registers. If all bits within this register are set to 0, the host data is unmodified. Setting the WORD_SWAP bit will cause the upper word of data to be exchanged with the lower word. The BYTE_SWAP bit will reverse the bytes within each word. The BIT_SWAP bit will reverse the bit ordering within a byte: in byte 0, bits[7:0] will be passed as bits[0:7]. All combinations of "SWAP" bits are valid.



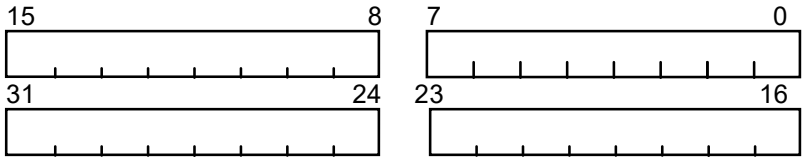
Bits	Name	Value	Function
CMD_HDF[0]	BIT_SWAP	0x0 0x1	Bit ordering is not modified Swap bits within each byte
CMD_HDF[1]	BYTE_SWAP	0x0 0x1	Byte ordering is not modified Swap bytes within each word
CMD_HDF[2]	WORD_SWAP	0x0 0x1	Word ordering is not modified Swap words within each dword

5.8.21 Foreground RBASE_D +(0x068)

Name: FORE

Type: Memory mapped read write

The FORE register holds the foreground color used during line drawing and stipple operations as well as any operation with the SOLID bit set. This register is 32 bits at 32 bits per pixel, 16 bits at 16 bits per pixel, and 8 bits at 8 bits per pixel.

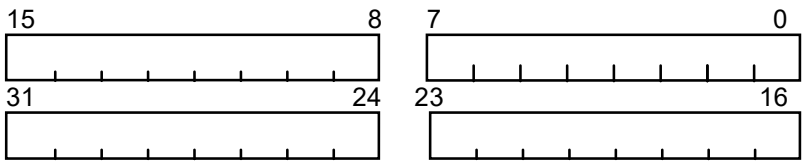


5.8.22 Background RBASE_D +(0x006C)

Name: BACK

Type: Memory mapped read write

The BACK register holds the background color used during patterned line and stipple operations. During these operations, a zero value in the pattern or stipple will select the BACK register as the source to be written out as the pixel value. This register is not used if transparency is set. This register is 32 bits at 32 bits per pixel, 16 bits at 16 bits per pixel, and 8 bits at 8 bits per pixel.

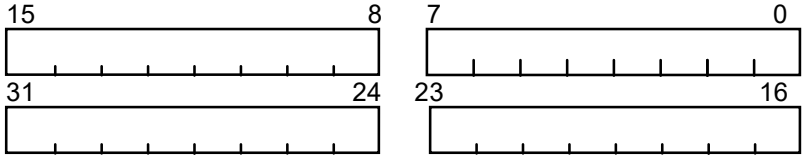


5.8.23 Plane Mask RBASE_D +(0x0070)

Name: MASK

Type: Memory mapped read write

The MASK register allows for selective writing of pixel planes. Using this feature, bit maps may be partitioned into logical groups for selective updating. A bit value of zero disables writing while a one enables writing to the corresponding plane. **(Ask Jim about masking for different bit depths)**

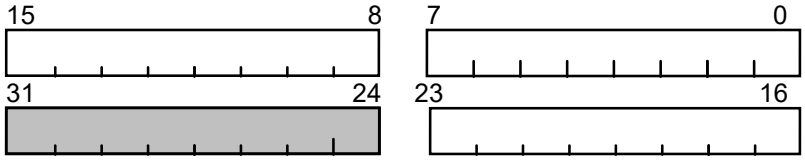


5.8.24 Color Key Register Mask RBASE_D +(0x0074)

Name: DE_KEY

Type: Memory mapped read write

The DE_KEY register provides a 24 bit register for a 2D color key compare. The 2D Key control register is located in BUF_CTRL.

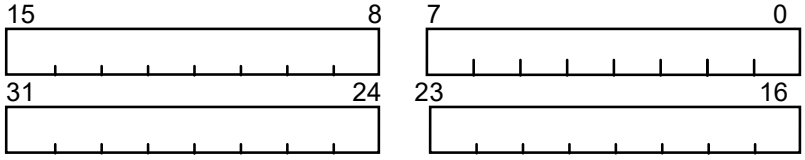


5.8.25 Line Pattern Register RBASE_D +(0x0078)

Name: LPAT

Type: Memory mapped read write

The LPAT register holds a 32 bit pattern in effect during line drawing. A one in the LPAT register corresponds to rendering in the foreground color while a zero corresponds to the background color or destination color if transparency is enabled. LPAT[0] represents the first bit of the pattern in a line assuming that the pattern reset bit is set in the CMD_PAT register. Solid lines may be drawn by setting the pattern register to 0xffffffff and setting Shade bit to Zero or by setting the SOLID bit in the CMD_STYLE register.

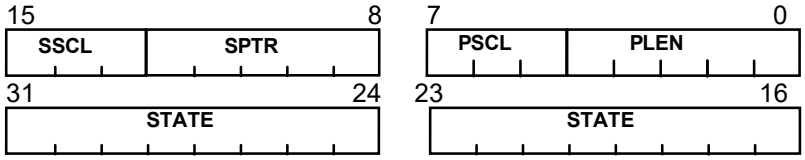


5.8.26 Line Pattern Control Register RBASE_D +(0x007C)

Name: PCTRL

Type: Memory mapped read write

The pattern control register allows the modifying of the pattern in LPAT by specifying a scaling factor, a length, and specifying a starting condition. The SSCL field (Starting Scale factor) allows for accurate alignment of scaled patterns.



Bits	Name	Value Function
PCTL[4:0]	PLEN	Pattern length. The length assumes a scale factor of 1x. 0x0 32 bit long pattern 0x1 1 bit long pattern ... 0x1f 31 bit long pattern
PCTL[7:5]	PSCL	Pattern scale factor (stretch) 0x0 x1 scaling 0x1 x2 scaling ... 0x7 x8 scaling
PCTL[12:8]	SPTR	Pattern offset 0x00 1st bit of pattern is LPAT[0] 0x01 1st bit of pattern is LPAT[1] ... 0x1f 1st bit of pattern is LPAT [31]
PCTL[15:13]	SSCL	Initial scale offset 0x0 no offset 0x1 offset by one ... 0x7 offset by seven
PCTL[31:16]	STATE	State always contains the current state of the pattern controller. By reading the contents of the field and moving it directly to {SSCL,SPTR,PSCL,PLEN} the line pattern will continue from exactly where it left off. This is very useful for context switching.

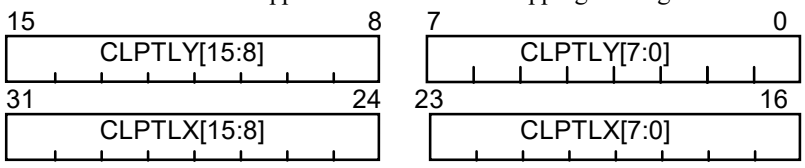
(Jim, what is the intention of the STATE field in PCTL)

5.8.27 Top Left of Clip Area RBASE_D + (0x0080)

Name: CLPTL

Type: Memory mapped read write

The CLPTL defines the upper left corner of the clipping rectangle.



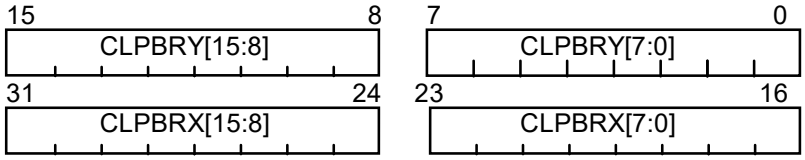
Bits	Name	Function
CLPTL[15:0]	CLPTLY	Y coordinate of the top left corner of the clipping rectangle.
CLPTL[31:16]	CLPTLX	X coordinate of the top left corner of the clipping rectangle.

5.8.28 Bottom Right of Clip Area RBASE_D + (0x0084)

Name: CLPBR

Type: Memory mapped read write

The CLPBR defines the bottom right corner of the clipping rectangle.



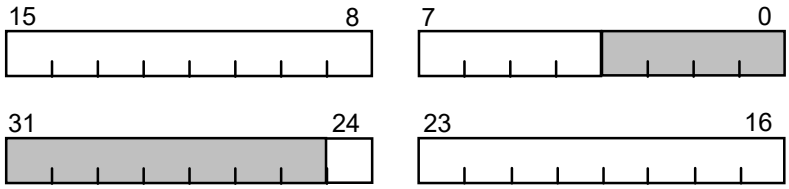
Bits	Name	Function
CLPBR[15:0]	CLPBRY	Y coordinate of the bottom right corner of the clipping rectangle.
CLPBR[31:16]	CLPBRX	X coordinate of the bottom right corner of the clipping rectangle.

5.8.29 Drawing Engine Z Origin RBASE_D + (0x0100)

Name: DE_ZORG

Type: Memory mapped read write

This register defines the linear address offset into the selected buffer or buffers for drawing engine Z cycles. This is a 16 byte-aligned address when accessing the Display Buffer.



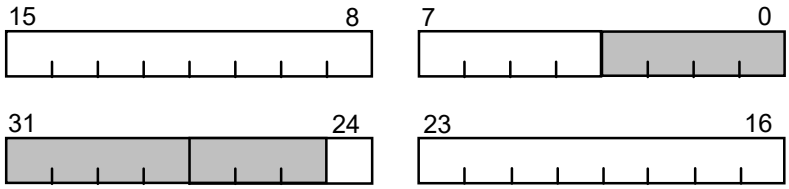
Bits	Name	Value	Function
DE_ZORG[24:4]	ZORGL		Linear origin of the Z buffer

5.8.30 Drawing Engine MipMap Origins RBASE_D + (0x00D0 through 0x00F4)

Name: LOD0_ORG, LOD1_ORG, LOD2_ORG, LOD3_ORG, LOD4_ORG, LOD5_ORG, LOD6_ORG, LOD7_ORG, LOD8_ORG, LOD9_ORG.

Type: Memory mapped read write

This register defines the linear address offset into the selected buffer or buffers for Texture mapping with Mip-Mapping enabled. This is a 128 bit word address when accessing the Display Buffer. LOD0_ORG must be loaded with the origin of the largest mipmap (or the single texture map origin in non-mipmapped mode) with the smallest mipmap in a higher numbered origin. Each mipmap must be a power of 2 smaller than the previous mipmap with the largest mipmap size specified in TEX_CNTRL and the number of mipmaps also specified. The Texture Pitch (TPTCH) must also be specified for LOD0. SilverHammer then calculates the pitches for all subsequent mipmap levels. The pitch must be at a 128 bit boundary, so when the pitch falls below 128, the pitch still is 128. Therefore, a 2x2 texture at 32 bpp still would take up 2x128, (1/2 is not used). (Pixels are packed in the lower order bits, i.e. in 32bpp, the 2 pixels will be loaded in [63:0] in each 128-bit row.) SilverHammer supports up to 10 LOD's to be used (512x512 to 1x1) when mipmapping is enabled. Rectangular power of 2 Mipmaps are supported.



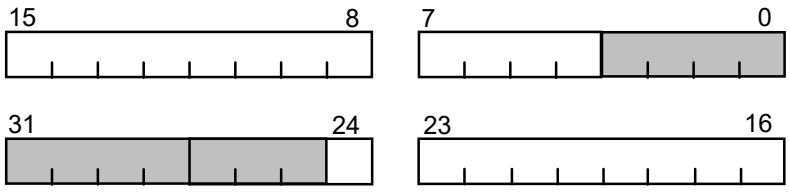
Bits	Name	Value	Function
LOD0_ORG[24:4]	LOD0_ORG		Linear origin mode for LOD0
LOD1_ORG[24:4]	LOD1_ORG		Linear origin mode for LOD1
LOD2_ORG[24:4]	LOD2_ORG		Linear origin mode for LOD2
LOD3_ORG[24:4]	LOD3_ORG		Linear origin mode for LOD3
LOD4_ORG[24:4]	LOD4_ORG		Linear origin mode for LOD4
LOD5_ORG[24:4]	LOD5_ORG		Linear origin mode for LOD5
LOD6_ORG[24:4]	LOD6_ORG		Linear origin mode for LOD6
LOD7_ORG[24:4]	LOD7_ORG		Linear origin mode for LOD7
LOD8_ORG[24:4]	LOD8_ORG		Linear origin mode for LOD8
LOD9_ORG[24:4]	LOD9_ORG		Linear origin mode for LOD9

Drawing Engine Palette Origin RBASE_D + (0x0118)

Name: DE_TPALORG

Type: Memory mapped read write

This register defines the linear address offset into the selected buffer or buffers for the texture cache palette. This is a 128 bit word address when accessing the Display Buffer.



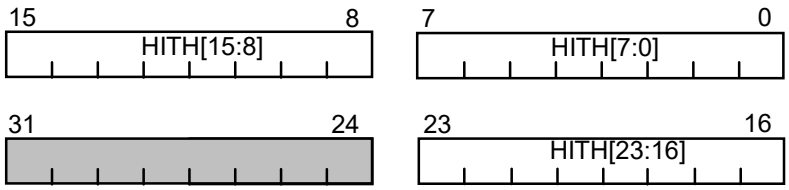
Bits	Name	Value	Function
DE_TPALORG[24:4]	TPAL_ORG		Linear origin of the Texture Palette

5.8.33 HITHER Clip Plane RBASE_D + (0x011C)

Name: HITH

Type: Memory mapped read write

The HITH defines the front most clipping plane.



Bits	Name	Function
HITH[23:0]	HITH	Z coordinate hither clipping plane.

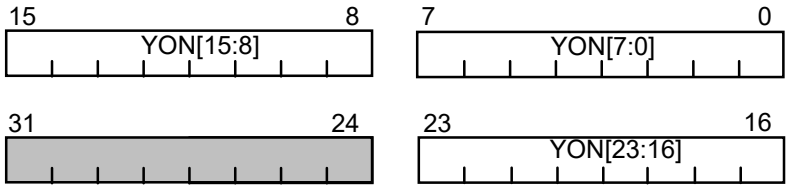
* 23:0 for 24 bit Z, or 15:0 for 16 bit Z.

5.8.34 YON Clip plane RBASE_D + (0x0120)

Name: YON

Type: Memory mapped read write

The YON defines the back most clipping plane.



Bits	Name	Function
YON[23:0]	YON	Z coordinate YON clipping plane.

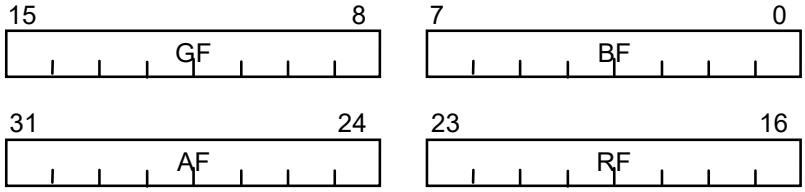
* 23:0 for 24 bit Z, or 15:0 for 16 bit Z.

5.8.36 Fog Color Register RBASE_D + (0x0124)

Name: FOG_COL

Type: Memory mapped read write

The color of the fog which is present across a primitive. It is specified in an ARGB format.



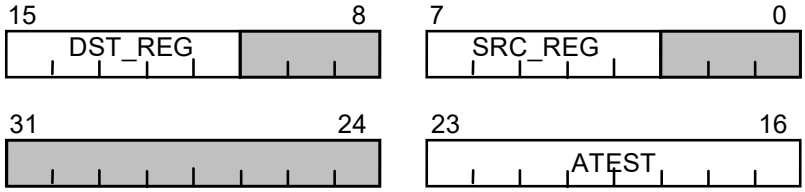
Bits	Name	Function
FOG_COL[7:0]	BF	Blue component of the Fog Color
FOG_COL[15:8]	GF	Green component of the Fog Color
FOG_COL[23:16]	RF	Red component of the Fog Color
FOG_COL[31:24]	AF	Alpha component of the Fog Color

5.8.37 Alpha Register RBASE_D + (0x0128)

Name: ALPHA_REG

Type: Memory mapped read write

The Alpha register contains the alpha test value for alpha compare mode. It also contains the source and destination alpha values that are used for alpha blending under the control of ACNTRL.



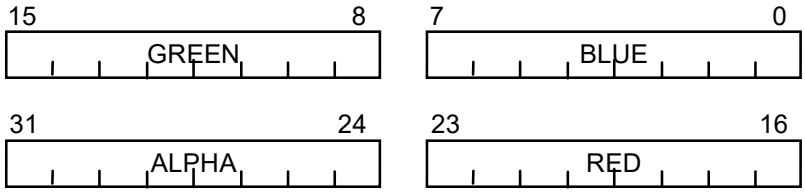
Bits	Name	Function
ALPHA[7:0]	SRC_REG	Alpha source register for blending. Lower three bits are considered 0.
ALPHA[15:8]	DST_REG	Alpha destination register for blending. Lower 3 bits are considered 0.
ALPHA[23:16]	ATEST	Alpha test register for alpha compare
ALPHA[31:24]	Reserved	Reserved

5.8.39 Texture Border Color Register RBASE_D + (0x012C)

Name: TBORD_COL

Type: Memory mapped read write

The Texture border color register contains the RBG value to clamp the color value to when Bordering is turned on in TEX_CTRL and the texture map U,V coordinates extend beyond the size of the texture map.

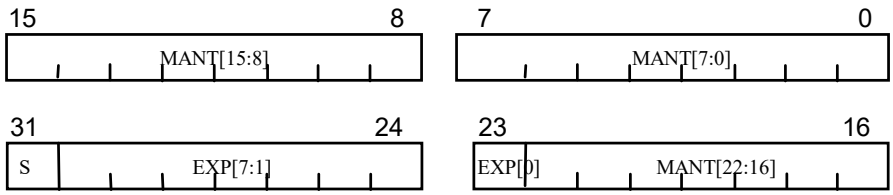


5.8.40 Floating Point Interface to the color registers RBASE_D + (0x0130 – 0x015C)

Name: V0_A_FP, V0_R_FP, V0_G_FP, V0_B_FP, V1_A_FP, V1_R_FP, V1_G_FP, V1_B_FP, V2_A_FP, V2_R_FP, V2_G_FP, V2_B_FP,

Type: Memory mapped write only

In SilverHammer, an interface is in place which will allow the three triangle vertex color registers to be loaded using floating point colors. This removes software overhead when updating vertices and removes the float to fixed conversion that software must do for these registers. These registers are write only. These floating point color registers can be used at any time, for a given command, it should either all be floating point or all integer.

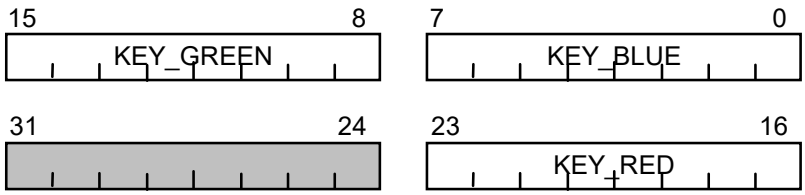


5.8.41 3D Color key compare registers RBASE_D + (0x0160, 0x0164)

Name: KEY_3D_LOW, KEY_3D_HIGH

Type: Memory mapped read write

SilverHammer allows 3D color keying to be performed on a range of colors. This range is specified in the two 3D color key registers. The range can be either between the two key colors, or outside the range of the two key colors. In texture mapping, 3D color keying is always done on the nearest Texel prior to any additive operations or filtering.

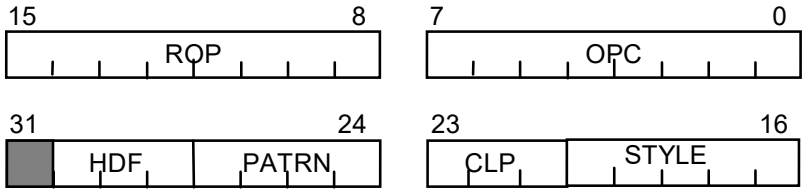


5.8.42 Command Register RBASE_D +(0x0168)

Name: CMD

Type: Memory mapped read write

The drawing engine command register can be accessed as a single 32 bit register or individual fields can be accessed in their own register space as described in the following pages.



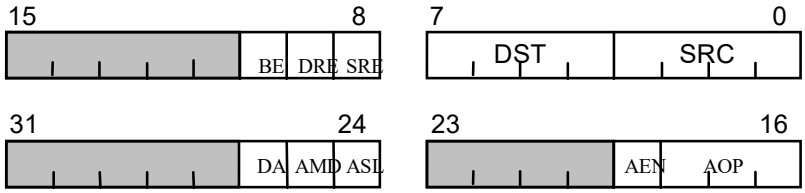
Bits	Name	Function
CMD[7:0]	OPC	Drawing command opcode
CMD[15:8]	ROP	Drawing Raster or logical operation
CMD[20:16]	STYLE	Solid, Transparency, Stipple control
CMD[23:21]	CLP	Clipping control
CMD[27:24]	PATRN	No last, pat reset, area pattern, and pattern cache enable.
CMD[30:28]	HDF	Host Data Format
CMD[31]	Reserved	Reserved

5.8.43 Command Register RBASE_D +(0x016C)

Name: ACNTRL

Type: Memory mapped read write

The alpha control register sets up the Imagine 4 blending unit for use with the drawing engine commands.



Note: DST and SRC use only 3 bits. Should we get rid of the 4th bit. Frank.

Bits	Name	Value	Function
ACNTRL[3:0]	SRC	see below	Blending Source Function
ACNTRL[7:4]	DST	see below	Blending Destination Function
ACNTRL[8]	SRE	 0x0 0x1	Source Register Enable Use Alpha in the Pixel Use source alpha register

ACNTRL[9]	DRE		Destination Register Enable
		0x0	Use Alpha in the Pixel
		0x1	Use destination alpha register
ACNTRL[10]	BE		Blending Enable
		0x0	Disable blending
		0x1	Enable blending
ACNTRL[15:11]	Reserved	0x0	Reserved
ACNTRL[18:16]	AOP		Alpha operator
		0x0	Never.
		0x1	Always.
		0x2	Less than.
		0x3	Less than or equal.
		0x4	Equal.
		0x5	Greater than or equal.
		0x6	Greater than.
		0x7	Not Equal.
ACNTRL[19]	AEN		Alpha Compare
		0x0	Disabled
		0x1	Enabled
ACNTRL[23:20]	Reserved	0x0	Reserved
ACNTRL[24]	ASL		Alpha Select
		0x0	Selects Texture Alpha
		0x1	Selects Vertex Alpha
ACNTRL[25]	AMD		Alpha Modulate
		0x0	Disabled
		0x1	Enabled
ACNTRL[26]	DAB		Decal Alpha Blend Mode
		0x0	Disabled
		0x1	Enabled
ACNTRL[31:27]	Reserved	0x0	Reserved

The Imagine 4 alpha blending modes follow the standard OpenGL conventions which are outlined in the following two tables:

Source Blending Table

Source Factor	Definition	Operation
0x0	SRC_ZERO	(0,0,0,0)
0x1	SRC_ONE	(1,1,1,1)
0x2	SRC_DST_COLOR	(Ad,Rd,Gd,Bd)
0x3	SRC_ONE_MINUS_DST	(1,1,1,1)-(Ad,Rd,Gd,Bd)
0x4	SRC_SRC_ALPHA	(As,As,As,As)
0x5	SRC_ONE_MINUS_SRC_ALPHA	(1,1,1,1)-(As,As,As,As)
0x6	SRC_DST_ALPHA	(Ad,Ad,Ad,Ad)
0x7	SRC_ONE_MINUS_DST_ALPHA	(1,1,1,1)-(Ad,Ad,Ad,Ad)

Destination Factor Table

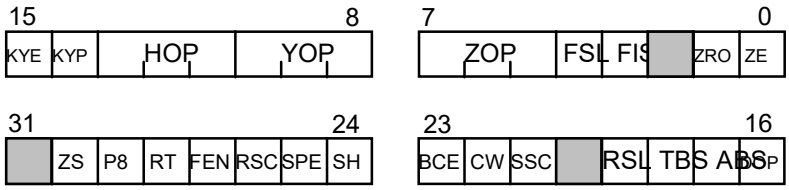
Destination Factor	Definition	Operation
0x0	DST_ZERO	(0,0,0,0)
0x1	DST_ONE	(1,1,1,1)
0x2	DST_SRC_COLOR	(As,Rs,Gs,Bs)
0x3	DST_ONE_MINUS_SRC	(1,1,1,1)-(As,Rs,Gs,Bs)
0x4	DST_SRC_ALPHA	(As,As,A,s,As)
0x5	DST_ONE_MINUS_SRC_ALPHA	(1,1,1,1)-(As,As,As,As)

0x6	DST_DST_ALPHA	(Ad,Ad,Ad,Ad)
0x7	DST_ONE_MINUS_DST_ALPHA	(1,1,1,1)-(Ad,Ad,Ad,Ad)

5.8.44 3D Control Register RBASE_D +(0x0170)

Name: 3D_CTRL

Type: Memory mapped read write

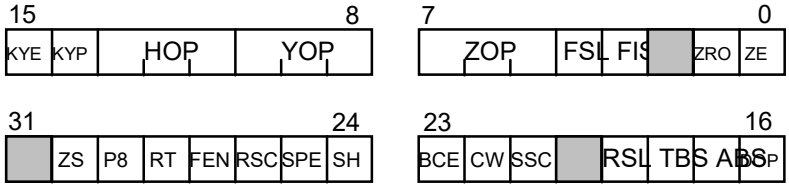


Bits	Name	Value	Function
3D_CTRL[0]	ZE		Z buffer enable bit.
		0x0	No Z buffer update or compare.
		0x1	Z buffer compare enabled, and updated if ZRO = 0.
3D_CTRL[1]	ZRO		Z read only mode.
		0	Updating of Z buffer enabled.
		1	Z buffer in read only mode.
3D_CTRL[2]	Reserved	Reserved	Reserved
3D_CTRL[3]	FIS		Fog Index Select
		0x0	Selects 1/w as Index to Fog Table
		0x1	Selects Z as Index to Fog Table
3D_CTRL[4]	FSL		Fog Selector
		0x0	Vertex Fogging
		0x1	Table based Fogging
3D_CTRL[7:5]	ZOP		Z operator
		0x0	Never.
		0x1	Always.
		0x2	Less than.
		0x3	Less than or equal.
		0x4	Equal.
		0x5	Greater than or equal.
		0x6	Greater than.
		0x7	Not Equal.
3D_CTRL[10:8]	YOP		YON operator
		0x0	Never.
		0x1	Always.
		0x2	Less than.
		0x3	Less than or equal.
		0x4	Equal.
		0x5	Greater than or equal.
		0x6	Greater than.
		0x7	Not Equal.

3D Control Register RBASE_D + (0x0170) ***Continued***

Name: 3D_CTRL

Type: Memory mapped read write



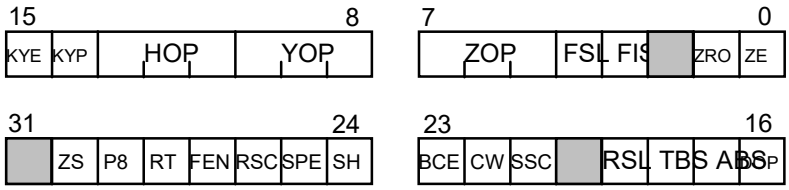
3D_CTRL[13:11]	HOP	 0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7	Hither operator Never. Always. Less than. Less than or equal. Equal. Greater than or equal. Greater than. Not Equal.
3D_CTRL[14]	KYP	 0x0 0x1	3D Key Polarity Inclusive (In range) Exclusive (Out of Range) NOTE: 3D and 2D keying are not exclusive.
3D_CTRL[15]	KYE	 0x0 0x1	3D Key Enable 3D Key Disabled 3D Key Enabled
3D_CTRL[16]	DOP	 0x0 0x1	Dither Operator No Dithering 8x8
3D_CTRL[17]	ABS	 0x0 0x1	Alpha Blend Select Disabled Enabled
3D_CTRL[18]	TBS	 0x0 0x1	Texture Blend Select Disabled Enabled
3D_CTRL[19]	RSL	 0x0 0x1	RGB Select Selects Texture RGB Selects Vertex RGB
3D_CTRL[20]	Reserved	Reserved	Reserved
3D_CTRL[21]	SSC	 0x0 0x1	XY Setup Spatial Center 0.0 Centered 0.5 Centered
3D_CTRL[22]	CW	 0x0 0x1	CW/CCW selector for culling CW CCW
3D_CTRL[23]	BCE	 0x0 0x1	Backface culling Enable Disabled Enabled

3D_CTRL[24]	SH	 0x0 0x1	Gouraud Shading Disabled Enabled
3D_CTRL[25]	SPE	 0x0 0x1	Specular Lighting Disabled Enabled
3D_CTRL[26]	RSC	 0x0 0x1	UV Renderer Spatial Center 0.0 Centered 0.5 Centered

3D Control Register RBASE_D + (0x0170) ***Continued***

Name: 3D_CTRL

Type: Memory mapped read write



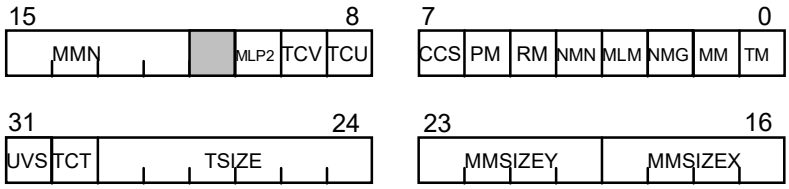
3D_CTRL[27]	FEN		Fog/Texture Composite Mode ***Note: When Mipmap Linear is enabled, this mode bit becomes the Texture Composite Mode bit.***
		0x0 0x1	Disabled Enabled
3D_CTRL[28]	RT		Rectangle
		0x0 0x1	Draw triangle in normal mode. Draw rectangle (P0, P1 and P2 must be sorted)
3D_CTRL[29]	P8		8bpp Palettized video support
		0x0 0x1	8bpt normal mode 8bpt palette index mode*
3D_CTRL[30]	ZS		Z Scaling mode
		0x0 0x1	Z is not scaled Z is scaled
3D_CTRL[31]	Reserved	Reserved	Reserved

**** TEX_CNTRL[29:24] must be set to 0x0D (8-bit index).***

5.8.45 Texture Mapping Control Register RBASE _(D) + (0x0174)

Name: TEX_CNTRL

Type: Memory mapped read write



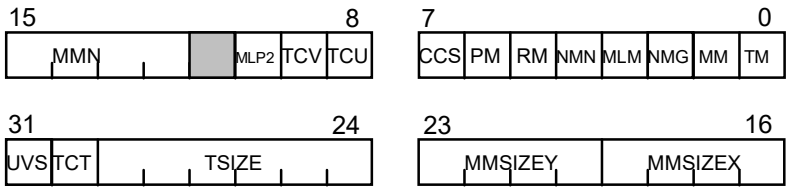
Bits	Name	Value	Function
TEX_CTRL[0]	TM		Texture Mode
		0x0 0x1	Texture mapping disabled Texture mapping enabled
TEX_CTRL[1]	MM		Mipmap mode
		0x0 0x1	Mipmapping disabled Mipmapping enabled

TEX_CTRL[2]	NMG	0x0 0x1	Nearest Mode Magnification Disabled Enabled
TEX_CTRL[3]	MLM	0x0 0x1	MipMap Linear Mode Disabled Enabled
TEX_CTRL[4]	NMN	0x0 0x1	Nearest Mode Minification Disabled Enabled
TEX_CTRL[5]	RM	0x0 0x1	RGB Modulation Disabled Enabled
TEX_CTRL[6]	PM	0x0 0x1	Perspective Correction Mode Disabled Enabled
TEX_CTRL[7]	CCS	0x0 0x1	Clamp Color Selector Use Edge Color Use Border Color
TEX_CTRL[8]	TCU	0x0 0x1	Texture Clamp in U Disabled Enabled
TEX_CTRL[9]	TCV	0x0 0x1	Texture Clamp in V Disabled Enabled
TEX_CTRL[10]	MLP2	0x0 0x1	MipMap Linear Pass Pass 1 Pass 2

Texture Mapping Control Register RBASE_D +(0x0174) ***Continued***

Name: TEX_CNTRL

Type: Memory mapped read write



TEX_CTRL[15:12]	MMN		Number of Mipmaps
		0x0	1 mipmap
		0x1	2 mipmaps
		0x2	3 mipmaps
		0x3	4 mipmaps
		0x4	5 mipmaps
		0x5	6 mipmaps
		0x6	7 mipmaps
		0x7	8 mipmaps
		0x8	9 mipmaps
		0x9	10 mipmaps
		0xA-0xF	Reserved
TEX_CTRL[19:16]	MMSIZEX		MipMap X size
		0x0	1 texel
		0x1	2 texels
		0x2	4 texels
		0x3	8 texels
		0x4	16 texels
		0x5	32 texels
		0x6	64 texels
		0x7	128 texels
		0x8	256 texels
		0x9	512 texels
		0xA- 0xF	Reserved
TEX_CTRL[23:20]	MMSIZEY		MipMap Y size
		0x0	1 texel
		0x1	2 texels
		0x2	4 texels
		0x3	8 texels
		0x4	16 texels
		0x5	32 texels
		0x6	64 texels
		0x7	128 texels
		0x8	256 texels
		0x9	512 texels
		0xA- 0xF	Reserved

Texture Mapping Control Register RBASE_D +(0x0174) ***Continued***

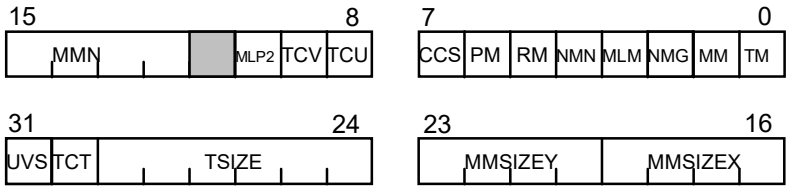
Note: Clarification on bpt. Frank

TEX_CTRL[29:24]	TSIZE		Texture map pixel format
			Palettized
		0x0	1 bpt 1555
		0x1	1 bpt 0565
		0x2	1 bpt 4444
		0x3	1 bpt 8888
		0x18	1 bpt 8332
		0x4	2 bpt 1555
		0x5	2 bpt 0565
		0x6	2 bpt 4444
		0x7	2 bpt 8888
		0x19	2 bpt 8332
		0x8	4 bpt 1555
		0x9	4 bpt 0565
		0xA	4 bpt 4444
		0xB	4 bpt 8888
		0xE	4 bpt 8332
		0x1A	8 bpt 1555
		0xF	8 bpt 0565
		0x1C	8 bpt 4444
		0x1D	8 bpt 8888 (Exact mode, must set nearest)
		0x3E	8 bpt 8888 (auto convert to 4444)
		0x3F	8 bpt 8888 (auto convert to 0565)
		0x1E	8 bpt 8332
			Non-Palettized
		0xC	8 bpt 1232
		0xD	8 bpt 0332 (when 3D_CTRL [29] = 0)
		0xD	8-bit index (when 3D_CTRL [29] = 1)
		0x10	16 bpt 4444
		0x11	16 bpt 1555
		0x12	16 bpt 0565
		0x13	16 bpt 8332
		0x14	32 bpt 8888
			OpenGL modes
		0x20	Alpha4
		0x21	Alpha8
		0x24	Luminance4
		0x25	Luminance8
		0x28	Luminance4_Alpha4
		0x29	Luminance6_Alpha2
		0x2A	Luminance8_Alpha8
		0x2C	Intensity4
		0x2D	Intensity8
		0x30	RGBA2
		Everything else	Reserved
TEX_CTRL[30]	TCT		Texture Cache Tile Mode
		0x0	Disabled
		0x1	Enabled

Texture Mapping Control Register RBASE_D +(0x0174) *Continued*

Name: TEX_CNTRL

Type: Memory mapped read write



TEX_CTRL[31]	UVS	0x0 0x1	U-V Scaling Scale U and V Disabled Scale U and V Enabled

5.8.46 3D Command Trigger Register RBASE_D + (0x01DC)

Name: 3D_TRIG

Type: Memory mapped read write

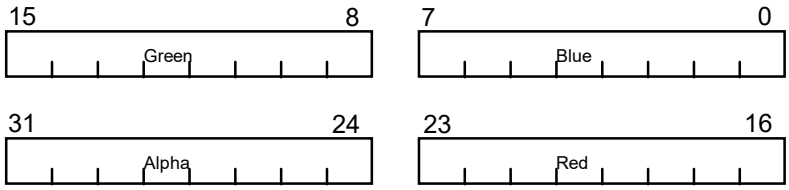
The 3D trigger register causes a 3D command to be loaded and executed when written to. The register needs to only be written to, no data need be supplied, and no data is contained within.

5.8.47 OpenGL Blend Color Register RBASE_D + (0x01E0)

Name: GLBLENDC

Type: Memory mapped read write

This color register is used in OpenGL for its Texture Blend Function. It is specified in an ARGB format.



Bits	Name	Function
GLBLENDC[7:0]	Blue	Blue component of the OpenGL Blend Color
GLBLENDC[15:8]	Green	Green component of the OpenGL Blend Color
GLBLENDC[23:16]	Red	Red component of the OpenGL Blend Color
GLBLENDC[31:24]	Alpha	Alpha component of the OpenGL Blend Color

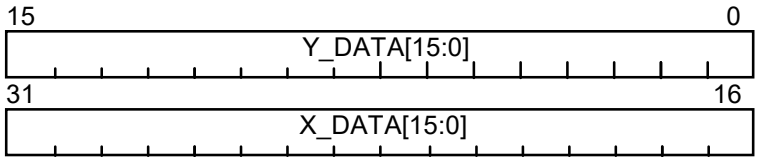
5.8.48 XY Parameter Registers RBASE_D + (0x0088 through 0x0098)

Name: XY0, XY1, XY2, XY3, XY4

Type: Memory mapped read write

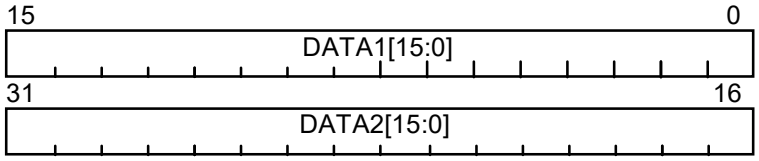
The XY parameter registers are general purpose registers used to hold different data values based on the type of command to be executed. The Command set chapter describes the use of these registers for each command. The data in these registers is one of two data formats.

X-Y FORMAT



Bits	Function
Y_DATA[15:0]	Y coordinate of a pixel location, or the Y portion of a dimension. Y_DATA is an integer value with range of +32767 to -32767.
X_DATA[31:16]	X coordinate of a pixel location, or the X portion of a dimension. X_DATA is an integer value with range of +32767 to -32767.

I FORMAT



The I format contains two general purpose 16 bit integer data values. (Data can be less than 16 bits as in direction bits for bit blt.)

XY1 is the trigger register for all 2D drawing operations. All other XY registers and parameter registers should be programmed before writing to XY1. Although XY1 can be written to as a byte, word, or Dword, the most significant byte of XY1 must be written for a command to be triggered.

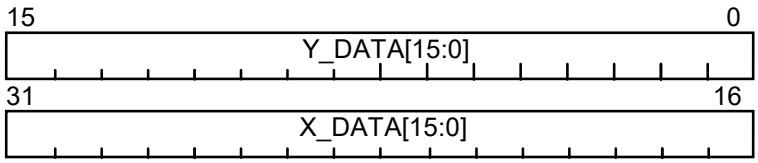
5.8.49 CP Parameter Registers RBASE_D + (0x0178 through 0x01D8)

Name: CP0 through CP24

Type: Memory mapped read write

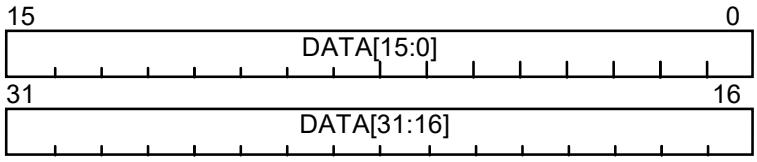
The CP parameter registers are general purpose registers used to hold different data values based on the type of command to be executed. The Command set chapter describes the use of these registers for each command. The data in these registers is one of three data formats.

X-Y FORMAT



Bits	Function
Y_DATA[15:0]	Y coordinate of a pixel location, or the Y portion of a dimension. Y_DATA is an integer value with range of +32767 to -32767.
X_DATA[31:16]	X coordinate of a pixel location, or the X portion of a dimension. X_DATA is an integer value with range of +32767 to -32767.

I FORMAT

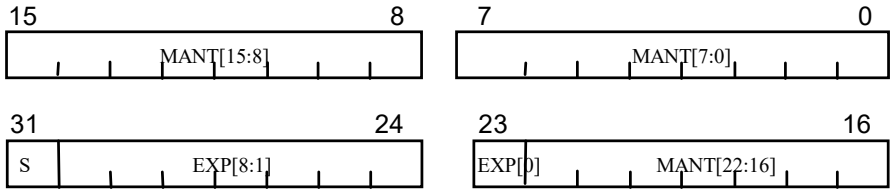


The I format contains one general purpose 32 bit integer data values.

3D_Trigger is the trigger register for all 3D operations. All other CP registers and parameter registers should be programmed before writing to 3D_TRIGGER.

Float Format

IEEE Single Precision Floating Point



S = Sign Bit
EXP = 8-bit Biased Exponent
MANT = 23 Bit MANTISSA with Implied Leading 1.

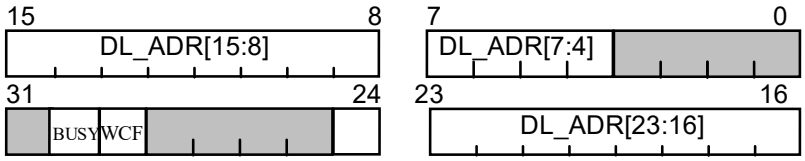
5.9 Display List Processor Registers

5.9.1 Display List Processor Address Register RBASE_D + (0x00F8)

Name: DL_ADR

Type: Memory mapped read write

This register defines the address in local RAM at which the display list processor will begin execution. A start address is written into DL_ADR at which time the DLP will accept end addresses which are associated with the start address. The list can be started or stopped at any time and the end address updated asynchronously. The DLP can also accept a new start address as a "pending list". Once the new start address is written, the DLP will continue executing the current list and seamlessly switch to the new list when the executing list is completed. All new end addresses written after a start address are associated with the new list, and the executing list is NOT accessible. Only 2 start addresses can be loaded at any time and Imagine will issue a PCI bus retry if a third start address is attempted.



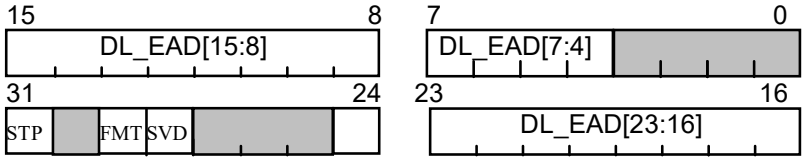
Bits	Name	Value	Function
DL_ADR[3:0]	Reserved	0x0	Reserved
DL_ADR[24:4]	DL_ADR		Display List Start Address
DL_ADR[28:25]	Reserved	0	Reserved
DL_ADR[29]	WCF	 0x0 0x1	Wait for Linear windows cache to flush - Wait for cache flush (wait for linear window's cache to flush before executing the list.) Enabled Disabled
DL_ADR[30]	BUSY	 0x0 0x1	Display List Busy Not Busy Busy
DL_ADR[31]	Reserved	0x0	Reserved

5.9.2 Display List Processor Control Register RBASE_D + (0x00FC)

Name: DL_CNTRL

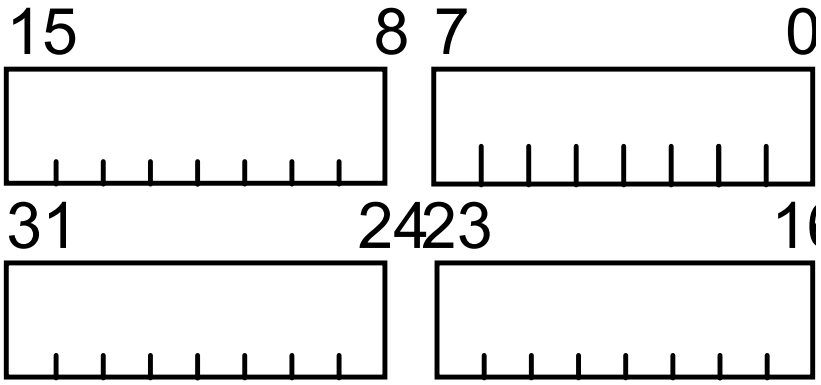
Type: Memory mapped read write

This register contains the control bits and the termination address of the display list.



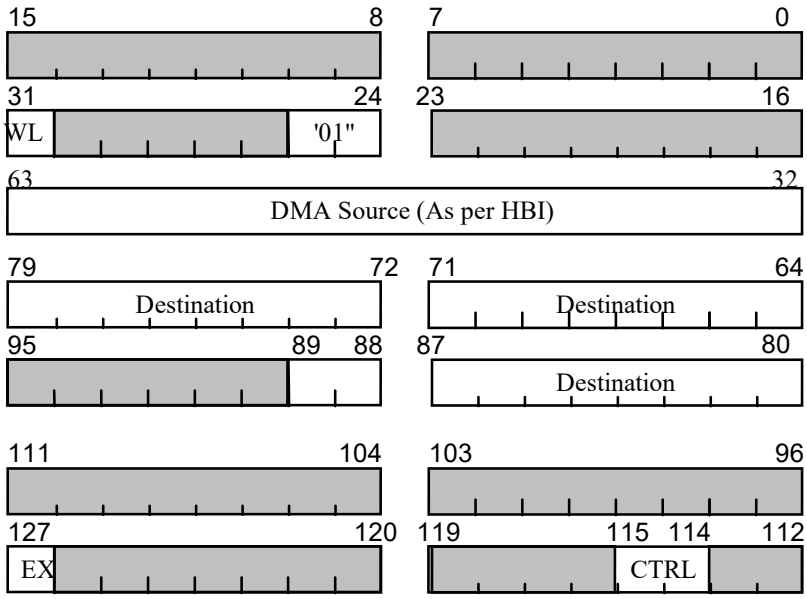
Bits	Name	Value	Function
DL_CNTRL[3:0]	Reserved	0x00	Reserved
DL_CNTRL[24:4]	DL_EAD		Display List End Address, When the Display List Processor reaches this address it terminates execution and sets the DL_STP bit.
DL_CNTRL[27:25]	Reserved	0x0	Reserved
DL_CNTRL[28]	DL_SVD		Display List Source Enable Override
		0x0	Use DL_SEN **(??? Frank…does this mean local buffer address in DL_ADR?)**
		0x1	Use AGP DMA **(AGP System Memory Address?)**
DL_CNTRL[29]	DL_FMT		Display List Format
		0x0	Display List Format 0, Write register defined in display list.
		0x1	Display List Format 1, Write only XY0,XY2,XY3,XY1.
DL_CNTRL[30]	Reserved	0x0	Reserved
DL_CNTRL[31]	DL_STP		Display List Processor Stop. This read write **(?)** bit is used to control the operation of the DLP.
		0x0	The DLP is currently running.
		0x1	And Busy is zero the DLP is idle.**(Frank, please clarify)**

5.9.3 Display List Instruction Word, Format Zero No DMA "DL_FMT = 0 & DL_IW[25:24] =00"



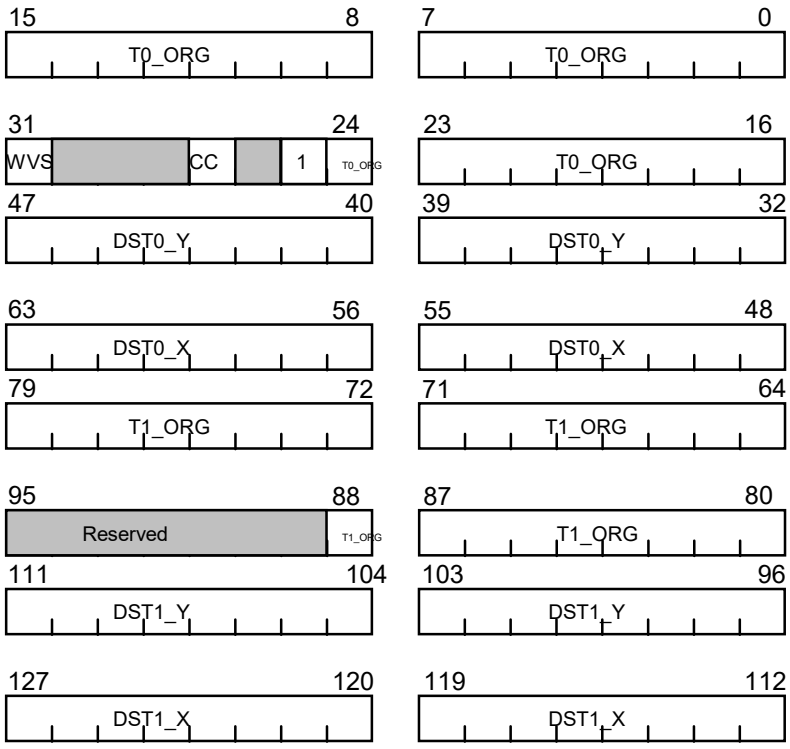
Bits	Name	Value	Function
DL_IW[7:0]	AAD		Address for register A data.
DL_IW[15:8]	BAD		Address for register B data.
DL_IW[23:16]	CAD		Address for register C data.
DL_IW[25:24]	SELECT	0x0	Must be 0 for this format
DL_IW[27:26]	WCNT		Word Count
		0x0	3 Words
		0x1	1 Word
		0x2	2 Words
		0x3	3 Words
DL_IW[28]	SA		Register A Address Selector
		0x0	DE registers 0x000 through 0x0FC
		0x1	DE registers 0x100 through 0x1FC
DL_IW[29]	SB		Register B Address Selector
		0x0	DE registers 0x000 through 0x0FC
		0x1	DE registers 0x100 through 0x1FC
DL_IW[30]	SC		Register C Address Selector
		0x0	DE registers 0x000 through 0x0FC
		0x1	DE registers 0x100 through 0x1FC
DL_IW[31]	WV		Wait for vertical blank
		0x0	Do not wait for vertical interrupt before executing current command.
		0x1	Wait for vertical interrupt before executing current command.
DL_IW[63:32]	ADAT		Data to be written to register addressed by AAD
DL_IW[95:64]	BDAT		Data to be written to register addressed by BAD
DL_IW[127:96]	CDAT		Data to be written to register addressed by CAD

5.9.4 Display List Instruction Word, Format Zero DMA "DL_FMT = 0 & DL_IW[25:24] = 01"



Bits	Name	Value	Function
DL_IW[23:0]	Reserved	0x0	Reserved
DL_IW[25:24]	SELECT	0x01	Must be 01 for this format **(Fix Diagram)**
DL_IW[30:26]	Reserved	0x0	Reserved
DL_IW[31]	WL	 0x0 0x1	Wait for AGP DMA done Don't wait for current (and previous DMA's) to finish Wait for pending and this DMA to finish
DL_IW[63:32]	SRC		AGP DMA Source (see DMA_SRC)
DL_IW[89:64]	DST		AGP DMA destination (see DMA_DST)
DL_IW[113:90]	Reserved	0x0	Reserved
DL_IW[115:114]	CTRL		AGP DMA Control (see DMA_CMD)
DL_IW[126:116]	Reserved	0x0	Reserved
DL_IW[127]	EX	 0x0 0x1	Expedite Normal Expedited

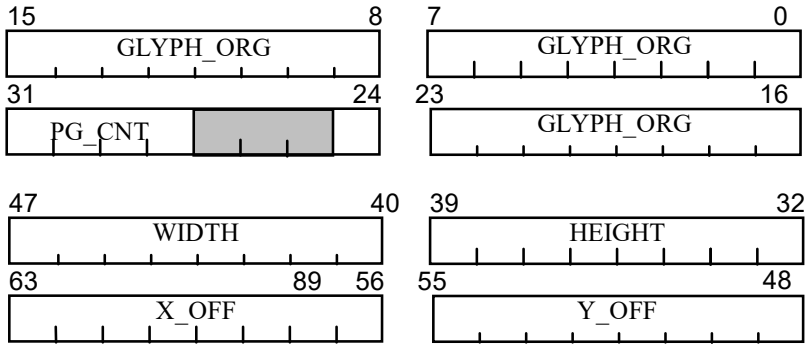
5.9.5 Display List Instruction Word, Format Zero TEXT "DL_FMT = 0 & DL_IW[25] = 1"



Bits	Name	Value	Function
DL_IW[24:0]	T0_ORG		Table 0 Origin
DL_IW[25]	1	0x1	Must be '1' for this format
DL_IW[26]	Reserved	0x0	Reserved
DL_IW[27]	CC	 0x0 0x1	Character Count One character Two characters
DL_IW[30:28]	Reserved	0x0	Reserved
DL_IW[31]	WVS	 0x0 0x1	Wait for vertical sync after this character Disabled Enabled
DL_IW[47:32]	DST0_Y		Destination 0 Y
DL_IW[63:48]	DST0_X		Destination 0 X
DL_IW[88:64]	T1_ORG		Table 1 Origin
DL_IW[95:89]	Reserved	0x0	Reserved
DL_IW[111:96]	DST1_Y		Destination 1 Y
DL_IW[127:112]	DST1_X		Destination 1 X

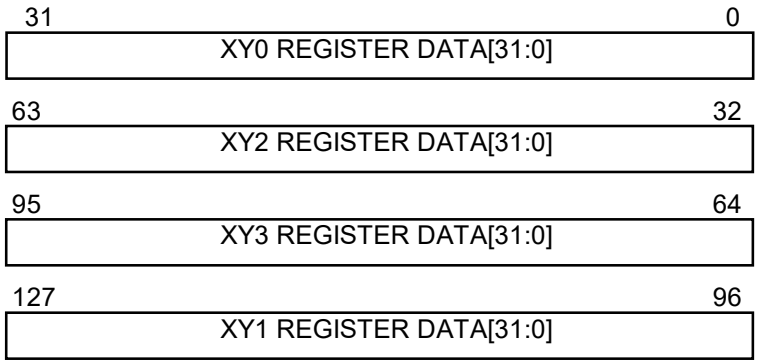
5.9.6 Text Table Format

The DLP uses the concept of “text tables” which contain information about character glyphs stored in off screen memory. A table entry is 64 bits long and can be packed such that two entries can exist in a 128 bit memory word. The origin in the DLP instruction points to the 64 bit memory word which contains the text entry. The definition of the table is as follows:



Bits	Name	Value	Function
TEXT[24:0]	GLYPH_ORG		Glyph Origin loaded into SORG
TEXT[27:25]	Reserved	0x0	Reserved
TEXT[31:28]	PG_CNT		Page Count loaded into XY3
TEXT[39:32]	Height		Height loaded into XY2
TEXT[47:40]	Width		Width, Loaded into XY2
TEXT[55:48]	Y_OFF		Y offset, subtract from dest_y
TEXT[63:56]	X_OFF		X Offset, add to dest_x

5.9.7 Display List Instruction Word, Format One "DL_FMT = 1"



Bits	Name	Value	Function
DL_IW[31:0]	XY0_DAT		Data to be written to register XY0
DL_IW[63:32]	XY2_DAT		Data to be written to register XY2
DL_IW[95:64]	XY3_DAT		Data to be written to register XY3
DL_IW[127:96]	XY1_DAT		Data to be written to register XY1

5.9.8 DLP Notes

- Waiting after DMA will cause the DLP to hold the DE busy and will not execute another command until the DMA pipeline is empty
- HBI DMA cannot be used at the same time as DLP DMA. The DMA pipeline must be empty prior to executing a DLP DMA list.
- Text mode only updates the SORG, XY2, XY3, and XY1 registers. All other registers (including command) must be set prior to DLP text operations.
- Text, DMA and 3 register writes may all be mixed in the same physical list. The 4 register mode must be executed in its own list.