STPC Client Programming Manual

Issue 2.2

August 28, 2000

STMicroelectronics

1.TABLE OF CONTENTS

2.LIST OF FIGURES

3.LIST OF TABLES

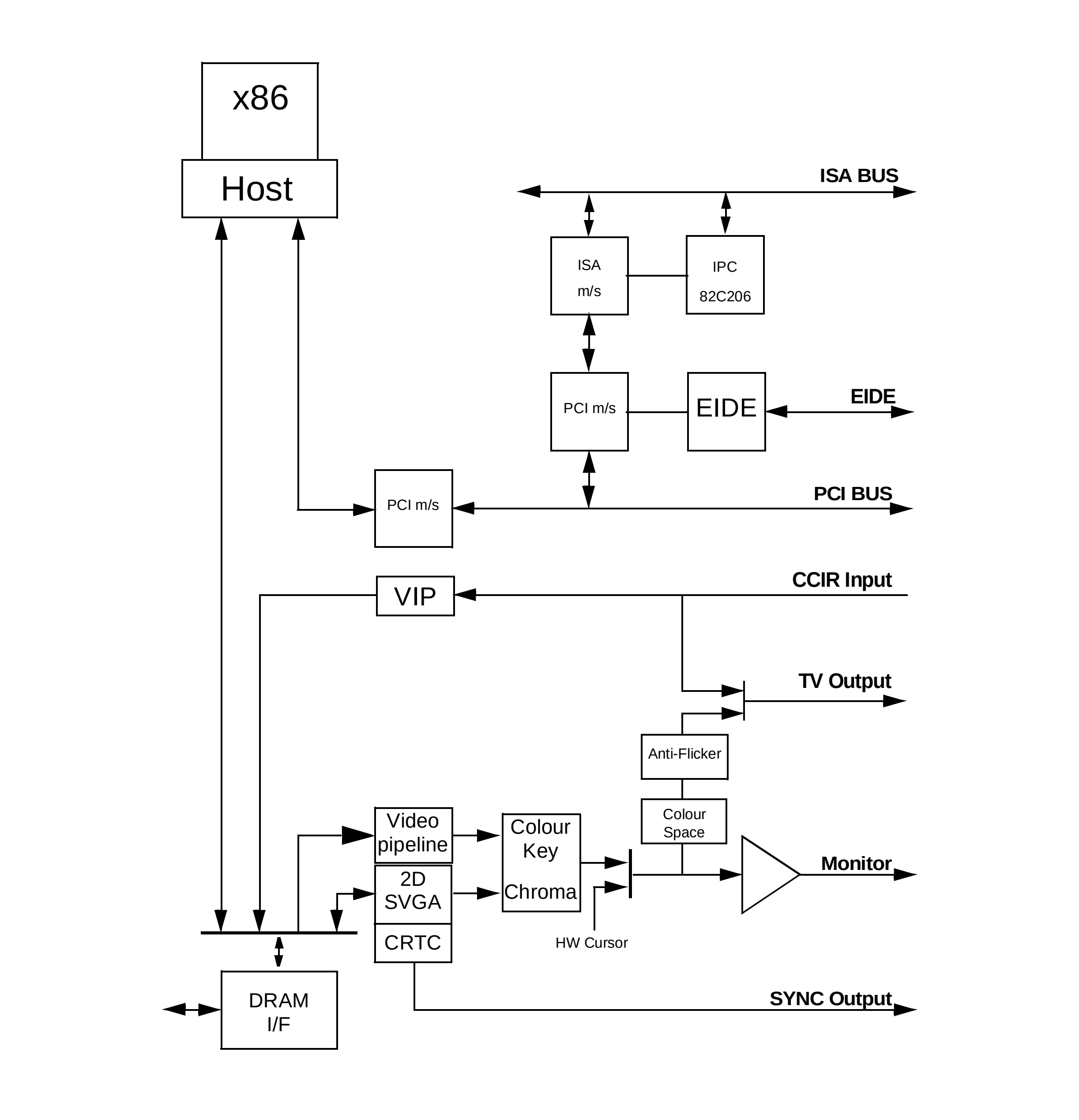
4. INTRODUCTION

This programming manual details the register sets for the STPC Client Device. The manual is split into chapters each dedicated to a function.

This documents contains all the information required to program and configure the STPC Client.

In order to use this manual to the full, you may want to make reference to the STPC and X86 core Datasheets.

Figure 4-1. Functionnal description



5. HOW TO USE THIS MANUAL

5.1 INTRODUCTION

This manual provides full technical documentation for the STPC device. It is recommended that the reader is familiar with the x86 series processors and PC compatible architectures before reading this document. Many terms are related directly to the PC architecture.

The manual itself is split into chapters. These chapters hold the information for a particular functional block of the device. For example, the chapter titled “Memory Access” gives the memory map of the STPC device, the memory architecture and interface to the external DRAM modules.

5.2 SPECIFIC NOTES

5.2.1 RESERVED BITS

Write mode 1 is a subset of Write Mode 0. No CPU-supplied write data is used. The read data latched from a previous read operation is written. The bit mask is disabled. The map-masks are implemented as they are for Write Mode 0.

Many bits in the register descriptions are noted as reserved. These bits are not internally connected, physically not present or are used for testing purposes. In all cases these bits should be set to a ‘0’ when writing to a register with reserved bits. When reading from a register with reserved bits, these specific bits should be masked from the data value before action is taken on the data.

Any functionality found by setting the reserved bits to levels other than ‘0’ cannot and will not be guaranteed on future revisions of the circuit design. Thus it is not recommended to use the bits marked as reserved in any way different from noted above.

5.2.2 SIGNAL ACTIVE STATE

The pound symbol (#) following a signal name indicates that when the signal is in its active (asserted) state, the signal is at a logic low level. When the “#” is not present at the end of a signal name, the logic high level represents the active state.

5.2.3 HEXADECIMAL NOTATION

In this manual Hexadecimal (Hex) numbers (numbers to the base 16: [0-9,A-F]) are denoted by the postfix ‘h’.

For example a memory address 783A hexadecimal will be written 783Ah.

5.2.4 ENDIAN

In common with the x86 architecture, values in memory are little-endian, that is the lower part of the memory contains the least significant Byte.

For an 8-bit value

N	7	6	5	4	3	2	1	0

For a 16-bit (word) value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8

For a 24-bit value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8
N+2	23	22	21	20	19	18	17	16

For a 32-bit (long word) value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8
N+2	23	22	21	20	19	18	17	16
N+3	31	30	29	28	27	26	25	24

For a 64-bit (QUAD word) value

N	7	6	5	4	3	2	1	0
N+1	15	14	13	12	11	10	9	8
N+2	23	22	21	20	19	18	17	16
N+3	31	30	29	28	27	26	25	24
N+4	39	38	37	36	35	34	33	32
N+5	47	46	45	44	43	42	41	40
N+6	55	54	53	52	51	50	49	48
N+7	63	62	61	60	59	58	57	56

5.3 ISSUING NOTES

There are three levels identified; Advanced data, Preliminary data and Full production release.

Each level is identified in a specific way as follows.

Document Identification	Status	Definition	Release Identification
ADVANCED DATA	In design	This document based on the product specification. The information may be updated without notice. Large changes may still occur.	Release A, Release B...
PRELIMINARY DATA	Pre-production Data	This document contains preliminary data and may be updated without notice in order to improve the product features.	Issue 0.X.
FULL PRODUCTION DATA	Production Data	This is the finalized document and all test plans are completed. The information may be updated without notice in order to improve the product features.	Issue 1.X.

6. LIST OF REGISTERS

This chapter lists all the registers accessable by software.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
3.	**Power on strap Registers[3]**			0022h	
3.1.1	Strap Register 0	Strap0	Configuration	0023h	Index 04Ah
3.1.2	Strap Register 1	Strap1	Configuration		Index 04Bh
3.1.3	Strap Register 2	Strap2	Configuration		Index 04Ch
3.1.4	HCLK PLL Strap Register 0	HCLK_Strap	Configuration		Index 05Fh
7.5.	**Cache related registers**			0022h	
7.5.1.	Cache Architecture Register 0	Cash_Arc0	Configuration	0023h	Index 020h
7.5.2.	Cache Architecture Register 1	Cash_Arc1	Configuration		Index 021h
7.5.3.	Cache Architecture Register 2	Cash_Arc2	Configuration		Index 022h
7.6.	**Address decode related registers**			0022h	
7.6.1.	Memory Hole Control Register	Mem_Hole	Configuration	0023h	Index 024h
7.6.2.	Shadow Control Register 0	Shadow_0	Configuration		Index 025h
7.6.3.	Shadow Control Register 1	Shadow_1	Configuration		Index 026h
7.6.4.	Shadow Control Register 2	Shadow_2	Configuration		Index 027h
7.6.5.	Shadow Control Register 3	Shadow_3	Configuration		Index 028h
7.6.6.	VGA Decode Register	VGA_Dec	Configuration		Index 029h
7.7.	**Host DRAM controller registers**			0022h	
7.7.1.	DRAM Bank 0 Register	DRAM_B0	Configuration	0023h	Index 030h
7.7.2.	DRAM Bank 1 Register	DRAM_B1	Configuration		Index 031h
7.7.3.	DRAM Bank 2 Register	DRAM_B2	Configuration		Index 032h
7.7.4.	DRAM Bank 3 Register	DRAM_B3	Configuration		Index 033h
7.7.5.	Memory Bank Width Register	Mem_Width	Configuration		Index 034h
7.7.6.	DRAM Bank 0 Timing Parameter Register	DRAM_T0	Configuration		Index 035h
7.7.10.	Graphics Memory Size Register	Graph_Mem	Configuration		Index 036h
7.7.11.	Memory Type Register	Mem_Type	Configuration		Index 037h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
7.7.7.	DRAM Bank 1 Timing Parameter Register	DRAM_T1	Configuration		Index 038h
7.7.12.	DRAM Refresh Register	DRAM_Ref	Configuration		Index 039h
7.7.8.	DRAM Bank 2 Timing Parameter Register	DRAM_T2	Configuration		Index 03Ah
7.7.9.	DRAM Bank 3 Timing Parameter Register	DRAM_T3	Configuration		Index 03Bh
8.	**PCI Controller**				
8.3.	Configuration Address Register	Config_ Address	IO	0CF8h	
8.4.	Configuration Data Register	Config_Data	IO	0CFC - CFFh	
8.5.	**North Bridge Configuration Registers**				
8.5.1.	North Bridge Vendor Identification Register	NB_V_ID	PCI Config		Index 00h
8.5.2.	North Bridge Device Identification Register	NB_D_ID	PCI Config		Index 02h
8.5.3.	North Bridge PCI Command Register	NB_Com	PCI Config		Index 04
8.5.4.	North Bridge PCI Status Register	NB_Stat	PCI Config		Index 06h
8.5.5.	North Bridge PCI Revision Id Register	NB_R_ID	PCI Config		Index 08h
8.5.6.	North Bridge Device Class Code Register	NB_C_Code	PCI Config		Index 09h
8.5.7.	North Bridge Header Type register	NB_Contol	PCI Config		Index Eh
8.5.9.	North Bridge Control Register	NB_Cont	PCI Config		Index 050h
8.5.9.	North Bridge PCI Error Status Register	NB_E_Stat	PCI Config		Index 054h
8.7.	**South Bridge PCI Function 0 Configuration Registers**			0CF8h	
8.7.1.	South BridgeVendor Identification Register	SB_V_ID0	PCI config F#0	0CFCh	Index 00h
8.7.2.	South Bridge Device Identification Register	SB_D_ID0	PCI Config F#0		Index 02h
8.7.3.	South Bridge PCI Command Register	SB_Com_0	PCI Config F#0		Index 04h
8.7.4.	South Bridge PCI Status Register	SB_Stat0	PCI Config F#0		Index 06h
8.7.5.	South Bridge PCI Revision Id Register	SB_R_ID0	PCI Config F#0		Index 08h
8.7.6.	South Bridge Device Class Code Register	SB_C_Code0	PCI Config F#0		Index 09h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
8.7.7.	South Bridge Header Type Register	SB_Head0	PCI Config F#0		Index 0Eh
8.7.8.	South Bridge Miscellaneous Register	SB_Misc0			Index 040h
8.8.	**South Bridge PCI Function 1 Configuraiton Registers**			0CF8h	
8.8.1.	South Bridge Vendor Identification Register	SB_V_ID1	PCI config F#1	0CFCh	Index 00h
8.8.2.	South Bridge Device Identification Register	SB_D_ ID1	PCI Config F#1		Index 02h
8.8.3.	South Bridge PCI Command Register	SB_Com1	PCI Config F#1		Index 04h
8.8.4.	South Bridge PCI Status Register	SB_Stat1	PCI Config F#1		Index 06h
8.8.5.	South Bridge Revision ID Register	SB_R_ID1	PCI Config F#1		Index 08h
8.8.6.	South Bridge Programming Interface Register	Prog_Int	PCI Config F#1		Index 09h
8.8.7.	South Bridge Sub-Class Code Register	Sub_Class	PCI Config F#1		Index 0Ah
8.8.8.	South Bridge Base-Class code Register	Base_Class	PCI Config F#1		Index 0Bh
8.8.9.	South Bridge Latency Timer control Register	Lat_T	PCI Config F#1		Index 0Dh
8.8.10.	South Bridge Header Type Register	Head_T	PCI Config F#1		Index 0Eh
8.8.11.	South Bridge IDE Base Address 0 Register	Base0	PCI Config F#1		Index 010h
8.8.12.	South Bridge IDE Base Address 1 Register	Base1	PCI Config F#1		Index 014h
8.8.13.	South Bridge IDE Base Address 2 Register	Base2	PCI Config F#1		Index 018h
8.8.14.	South Bridge IDE Base Address 3 Register	Base3	PCI Config F#1		Index 01ch
8.8.15.	South Bridge IDE Base Address 4 Register	Base4	PCI Config F#1		Index 020h
8.8.16.	South Bridge Primary IDE Timing Register	Prime_IDE_T	PCI Config F#1		Index 040h
8.8.17.	South Bridge Secondary IDE Timing Register	Sec_IDE_T	PCI Config F#1		Index 044h
8.8.18.	South Bridge Miscellaneous Register	SB_Misc1	PCI Config F#1		Index 048h
9.5.	**ISA standard Registers**				
9.5.1.	DMA 1 Channel 0 Base and Current Address	DMA1_CBA0	IO	0000h	
9.5.1.	DMA 1 Channel 0 Base and Current Count	DMA1_CBC0	IO	0001h	
9.5.1.	DMA 1 Channel 1 Base and Current Address	DMA1_CBA1	IO	0002h	

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.5.1.	DMA 1 Channel 1 Base and Current Count	DMA1_CBC1	IO	0003h	
9.5.1.	DMA 1 Channel 2 Base and Current Address	DMA1_CBA2	IO	0004h	
9.5.1.	DMA 1 Channel 2 Base and Current Count	DMA1_CBC2	IO	0005	
9.5.1.	DMA 1 Channel 3 Base and Current Address	DMA1_CBA3	IO	0006h	
9.5.1.	DMA 1 Channel 3 Base and Current Count	DMA1_CBC3	IO	0007h	
9.5.1.	DMA 1 Read Status / Write Command Register	DMA1_RSWC	IO	0008h	
9.5.1.	DMA 1 Request Register	DMA1_RR	IO	0009h	
9.5.1.	DMA 1 Read Command / Write Single Mask Register	DMA1_RCWSM	IO	000Ah	
9.5.1.	DMA 1 Mode Register	DMA1_Mode	IO	000Bh	
9.5.1.	DMA 1 Set / Clear Byte Pointer Flip - Flop	DMA1_SCBPFF	IO	000Ch	
9.5.1.	DMA 1 Read Temp Register / Master Clear	DMA1_RTMC	IO	000Dh	
9.5.1.	DMA 1 Clear Mask / Clear All Request	DMA1_CMCAR	IO	000Eh	
9.5.1.	DMA 1 Read / Write all Mask Register Bits	DMA1_RWMB	IO	000Fh	
9.5.2.	Interrupt Controller 1 Registers	IC_1	IO	0020h	
9.5.2.	Interrupt Controller 1 Mask Register	IC_1MR	IO	0021h	
9.5.3.	Interval Timer Register Counter 0 Count	IT_0	IO	0040h	
9.5.3.	Interval Timer Register Counter 1 Count	IT_1	IO	0041h	
9.5.3.	Interval Timer Register Counter 2 Count	IT_2	IO	0042h	
9.5.3.	Command Mode Register	IT_3	IO	0043h	
9.5.4.	Port Bh Register	Port _B	IO	0061h	
9.5.5.	Port 60h Register	Port_60		0060h	
9.5.5.	Port 64h Register	Port_64		0064h	
9.5.6.	Port 70h Register	Port_70	IO	0070h	
9.5.7.	Interrupt Controller 2 Registers	IC_2R	IO	00A0h	
9.5.7.	Interrupt Controller 2 Mask	IC_2M	IO	00A1h	
9.5.8.	DMA Controller 2 Registers	DMA_Cont2	IO		
9.5.8.	DMA 2 Channel 0 Base and Current Address	DMA2_CBA0	IO	00C0h	
9.5.8.	DMA2 Channel 0 Base and Current Count	DMA2_CBC0	IO	00C2h	
9.5.8.	DMA 2 Channel 1 Base and Current Address	DMA2_CBA1	IO	00C4h	
9.5.8.	DMA 2 Channel 1 Base and Current Count	DMA2_CBC1	IO	00C6h	

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.5.8.	DMA 2 Channel 2 Base and Current Address	DMA2_CBA2	IO	00C8h	
9.5.8.	DMA 2 Channel 2 Base and Current Count	DMA2_CBC2	IO	00CAh	
9.5.8.	DMA 2 Channel 3 Base and Current Address	DMA2_CBA3	IO	00CCh	
9.5.8.	DMA 2 Channel 3 Base and Current Count	DMA2_CBC3	IO	00CEh	
9.5.8.	DMA 2 Read Status / Write Command Register	DMA2_RSWC	IO	00D0h	
9.5.8.	DMA 2 Request Register	DMA2_RR	IO	00D2h	
9.5.8.	DMA 2 Read Command / Write Single Mask Register	DMA2_RCWSM	IO	00D4h	
9.5.8.	DMA 2 Mode Register	DMA2_Mode	IO	00D6h	
9.5.8.	DMA 2 Set / Clear Byte Pointer Flip - Flop	DMA2_SCBPFF	IO	00D8h	
9.5.8.	DMA 2 Read Temporary / Master Clear	DMA2_RTMC	IO	00DAh	
9.5.8.	DMA 2 Clear Mask / Clear All Requests Register	DMA2_CMCAR	IO	00DCh	
9.5.8.	DMA 2 Read / Write all Mask Register Bits	DMA2_RWMRB	IO	00DEh	
9.5.9.	DMA Page Registers	DMA_Page	IO		
9.5.9.	DMA Page Registers Port 80h (reserved)	Port_80	IO	0080h	
9.5.9.	DMA Page Register Channel 2	DMA_PRC2	IO	0081h	
9.5.9.	DMA Page Register Channel 3	DMA_PRC3	IO	0082h	
9.5.9.	DMA Page Register Channel 1	DMA_PRC1	IO	0082h	
9.5.9.	DMA Page Register Port 84h (Reserved)	Port_84	IO	0084h	
9.5.9.	DMA Page Register Port 85h (Reserved)	Port_85	IO	0085h	
9.5.9.	DMA Page Register Port 86h (Reserved)	Port_86	IO	0086h	
9.5.9.	DMA Page Register Channel 0	DMA_PRC0	IO	0087h	
9.5.9.	DMA Page Register Port 87h	Port_87	IO	0088h	
9.5.9.	DMA Page Register Channel 6	DMA_PRC6	IO	0089h	
9.5.9.	DMA Page Register Channel 7	DMA_PRC7	IO	008Ah	
9.5.9.	DMA Page Register Channel 5	DMA_PRC5	IO	008Bh	
9.5.9.	DMA Page Register Port 8Bh (Reserved)	Port_8B	IO	008Ch	
9.5.9.	DMA Page Register Port 8Ch (Reserved)	Port_8C	IO	008Dh	
9.5.9.	DMA Page Register Port 8Dh (Reserved)	Port_8D	IO	008Eh	
9.5.9.	DMA Page Register Port 8Eh (Reserved)	Port_8E	IO	008Fh	

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

LIST OF REGISTERS

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.6.	**ISA Configuration Registers**			0022h	
9.6.1.	Miscellaneous Control Register 0	Misc_Cont0	Configuration	0023h	Index 050h
9.6.2.	Miscellaneous Control Register 1	Misc_Cont1	Configuration		Index 051h
9.6.3.	PIRQA Routing control Register 0	PAR_Cont0	Configuration		Index 052h
9.6.4.	PIRQB Routing control Register 0	PBR_Cont0	Configuration		Index 053h
9.6.5.	PIRQC Routing control Register 0	PCR_Cont0	Configuration		Index 054h
9.6.6.	PIRQD Routing control Register 0	PDR_Cont0	Configuration		Index 055h
9.6.7.	Interrupt Level Control Register 0	IRQ_Lev_C_0	Configuration		Index 056h
9.6.8.	Interrupt Level Control Register 1	IRQ_Lev_C_1	Configuration		Index 057h
9.6.9.	IPC Configuration Register	IPC_Conf	Configuration		Index 001h
9.6.10.	VMI IRQ Routing Control Register	VIR_Cont	Configuration		Index 058h
9.6.11.	ISA Syncronizer Bypass Register	ISA_Sync	Configuration		Index 059h
11.3.	**VGA registers**				
11.4.	General Registers				
11.4.1.	Motherboard Enable Register	MBEN		0094h	
11.4.2.	Add-in VGA Enable Register	ADDEN		046E8h	
11.4.3.	Video Subsystem Enable 1 Register	VSE1		0102h	
11.4.4.	Video Subsystem Enable 2 Register	VSE2		03C3h	
11.4.5.	Miscellaneous Output Register	MISC		03CC/ 03C2h	
11.4.6.	Input Status Register #0	INP0		03C2h	
11.4.7.	Input Status Register #1	INP1		03XAh	
11.5.	Sequencer Registers				
11.5.1.	Sequencer Index Register	SRX	IO	003C4h	
11.5.2.	Sequencer Reset Register	SR0	IO	003C5h	Index 000h
11.5.3.	Sequencer Clocking Mode Register	SR1	IO		Index 001h
11.5.4.	Sequencer Plane Mask Register	SR2	IO		Index 002h
11.5.5.	Sequencer Character Map Register	SR3	IO		Index 003h
11.5.6.	Sequencer Memory Mode Register	SR4	IO		Index 004h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.5.7.	Extended Register Lock/Unlock Register	SR6	IO		Index 006h
11.6.	Graphics Controller Registers				
11.6.1.	Graphics Controller Index Register	GRX	IO	003CEh	
11.6.2.	Graphics Set/Reset Register	GR0	IO	003CFh	Index 000h
11.6.3.	Graphics Enable Set/Reset Register	GR1	IO		Index 001h
11.6.4.	Graphics Color Compare Register	GR2	IO		Index 002h
11.6.5.	Raster Op/Rotate Count Register	GR3	IO		Index 003h
11.6.6.	Graphics Read Map Select Register	GR4	IO		Index 004h
11.6.7.	Graphics Mode Register	GR5	IO		Index 005h
11.6.8.	Graphics Miscellaneous Register	GR6	IO		Index 006h
11.6.9.	Graphics Color Don't Care Register	GR7	IO		Index 007h
11.6.10.	Graphics Bit Mask Register	GR8	IO		Index 008h
11.7.	**Attribute Controller Registers**				
11.7.1.	Attribute Controller Index Register	ARX	IO	03C0h	
11.7.2.	Attribute Palette Registers	AR0 - ARF	IO	03C1/03C0h	
11.7.3.	Attribute Ctrl Mode Register	AR10	IO	03C1/03C0h	
11.7.4.	Attribute Ctrl Overscan Color Register	AR11	IO	03C1/03C0h	
11.7.5.	Attribute Color Plane Enable Register	AR12	IO	03C1/03C0h	
11.7.6.	Attribute Horz Pixel Panning Register	AR13	IO	03C1/03C0h	
11.7.7.	Attribute Color Select Register	AR14	IO	03C1/03C0h	
11.8.	CRT Controller Registers				
11.8.1.	Index Register	CRX	see Note 1	03X4h	
11.8.2.	Horizontal Total Register	CR0	see Note 1	03X5h	Index 000h
11.8.3.	Horiz display End Register	CR1			Index 001h
11.8.4.	Horiz Blanking Start Register	CR2			Index 002h
11.8.5.	Horiz Blanking End Register	CR3			Index 003h
11.8.6.	Horiz Retrace Start Register	CR4			Index 004h
11.8.7.	Horizontal Retrace end Register	CR5			Index 005h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

LIST OF REGISTERS

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.8.8.	Vertical Total Register	CR6			Index 006h
11.8.9.	Overflow Register	CR7			Index 007h
11.8.10.	Screen A Preset Row Scan Register	CR8			Index 008h
11.8.11.	Character Cell Height Register	CR9			Index 009h
11.8.12.	Cursor Start Register	CRA			Index 00Ah
11.8.13.	Cursor End Register	CRB			Index 00Bh
11.8.14.	Start Address High Register	CRC			Index 00Ch
11.8.15.	Start Address Low Register	CRD			Index 00Dh
11.8.16.	Text Cursor Offset High Register	CRE			Index 00Eh
11.8.17.	Text Cursor Offset Low Register	CRF			Index 00Fh
11.8.18.	Vertical Retrace Start Register	CR10			Index 010h
11.8.19.	Vertical Retrace End Register	CR11			Index 011h
11.8.20.	Vertical Display End Register	CR12			Index 012h
11.8.21.	Offset Register	CR13			Index 013h
11.8.22.	Underline Location Register	CR14			Index 014h
11.8.23.	Vertical Blanking Start reg	CR15			Index 015h
11.8.24.	Vertical Blanking End Register	CR16			Index 016h
11.8.25.	Mode Register	CR17			Index 017h
11.8.26.	Line Compare Register	CR18			Index 018h
11.8.27.	Graphics Control Data	CR22			Index 019h
11.8.28.	Attribute Address Flipflop	CR24			Index 020h
11.8.29.	Attribute Index Readback	CR26			Index 021h
11.9.	**VGA Extended Registers**		see Note 1	03X4h	
11.9.1.	Repaint Control Register 0	CR19	see Note 1	03X5h	Index 019h
11.9.2.	Repaint Control Register 1	CR1A			Index 01Ah
11.9.3.	Repaint Control Register 2	CR1B			Index 01Bh
11.9.4.	Repaint Control Register 3	CR1C			Index 01Ch
11.9.5.	Page Register 0	CR1D			Index 01Dh
11.9.6.	Page Register 1	CR1E			Index 01Eh
11.9.7.	Graphics Extended Enable Register	CR1F			Index 01Fh
11.9.8.	Graphics Extended GBASE Register	CR20			Index 020h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.9.9.	Graphics Extended Aperture Register	CR21			Index 021h
11.9.10.	Repaint Control Register 4	CR25			Index 025h
11.9.11.	Repaint Control Register 5	CR27			Index 027h
11.9.12.	Palette Control Register	CR28			Index 028h
11.9.13.	Cursor Height Register	CR29			Index 029h
11.9.14.	Cursor Color 0 Register A	CR2A			Index 02Ah
11.9.15.	Cursor Color 0 Register B	CR2B			Index 02Bh
11.9.16.	Cursor Color 0 Register C	CR2C			Index 02Ch
11.9.17.	Cursor Color 1 Register A	CR2D			Index 02Dh
11.9.18.	Cursor Color 1 Register B	CR2E			Index 02Eh
11.9.19.	Cursor Color 1 Register C	CR2F			Index 02Fh
11.9.20.	Graphics Cursor Address Register 0	CR30			Index 030h
11.9.21.	Graphics Cursor Address Register 1	CR31			Index 031h
11.9.22.	Graphics Cursor Address Register 2	CR32			Index 032h
11.9.23.	Urgent Start Register	CR33			Index 033h
11.9.24.	Displayed Frame Y Offset 0 Register	CR34			Index 034h
11.9.25.	Displayed Frame Y Offset 1 Register	CR35			Index 035h
11.9.26.	Interlace Half Field Start Register	CR39			Index 039h
11.9.27.	Implementation Number Register	CR3A			Index 03Ah
11.9.28.	Graphics Version Register	CR3B			Index 03Bh
11.9.29.	DRAM Timing Parameter Register	CR3C			Index 03Ch
11.9.30.	DRAM Arbitration control Register 0	CR3D			Index 03Dh
11.9.31.	Miscellaneous Test Register	CR3E			Index 03Eh
11.9.32.	DDC Control Register	CR3F			Index 03Fh
11.9.33.	TV Interface Control Register	CR40			Index 040h
11.9.34.	TV Horizontal Active Video Start A Register	CR41			Index 041h
11.9.35.	TV Horizontal Active Video Start B Register	CR42			Index 042h
11.9.36.	TV Horizontal Sync End A Register	CR43			Index 043h
11.9.37.	TV Horizontal Sync End B Register	CR44			Index 044h
11.12.	**RAMDAC registers**				

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.12.1.	Palette Pixel Mask Register	Pixel_Mask		03C6h	
11.12.2.	Palette Read index Register	Read_Index		03C7h	
11.12.3.	Palette State Register	Palette_State		03C7h	
11.12.4.	Palette Write Index Register	Write_Index		03C8h	
11.12.5.	Palette Data Register	Palette_Data		03C9h	
11.13	**DCLK Control Registers**			0022h	
11.13.1.	DCLK Control Register 00	DCLK00	PCI Config	0023h	Index 042h
11.13.2.	DCLK Control Register 01	DCLK01	PCI Config		Index 043h
11.13.3.	DCLK Control Register 10	DCLK10	PCI Config		Index 044h
11.13.4.	DCLK Control Register 11	DCLK11	PCI Config		Index 045h
11.13.5.	DCLK Control Register 20	DCLK20	PCI Config		Index 046h
11.13.6.	DCLK Control Register 21	DCLK21	PCI Config		Index 047h
11.13.7.	DCLK Control Register 30	DCLK30	PCI Config		Index 048h
11.13.8.	DCLK Control Register 31	DCLK31	PCI Config		Index 049h
12.	**Graphics Engine**				
12.5.	**VGA Operand Sources**				
12.9.1.	Back Ground Color Register	Background		GBase +400000h	Index 0004h
12.9.2.	Cursor Coordinate Register	Cursor_XY			Index 011Ch
12.9.3.	Top of Data FIFO Register	Data_Port			Index 0804h
12.9.4.	Destination Operand Base Address Register	Dst_Base			Index 0018h
12.9.5.	Destination Pitch Register	Dst_Pitch			Index 0028h
12.9.6.	Destination Operand Coordinate Register	Dst_XY		GBase +410000h	
12.9.7.	Foreground Color Register	Foreground		GBase +400000h	Index 0034h
12.9.8.	Height Register	Height			Index 0048h
12.9.9.	Pattern Base Address Operand Register	Pattern			Index 0058h
12.9.10.	Pixel Depth Operand Register	Pixel_Depth			Index 007Ch
12.9.11.	Raster Operation Register	ROP			Index 008Ch

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
12.9.12.	Source Base Address Operand Register	Src_Base			Index 0098h
12.9.13.	Source Pitch Operand Register	Src_Pitch			Index 00ACh
12.9.14.	Source Coordinate Register	Src_XY			Index 00BDh
12.9.15.	Status Register	Status			Index 0908
12.9.16.	Width Register	Width			Index 00C8h
12.9.17.	Extra Use Register	Xtra			Index 00D4h
12.9.18.	SRC Transparency Compare Register	SRC_ Transparency			Index 0ECh
12.9.19.	DST Transparency Compare Register	DST_ Transparency			Index 0FCh
12.12	**Graphics Clock Registes**			0022h	
12.12.1	GCLK Control Register 0	GCLK00	PCI Config	0023h	Index 040h
12.12.2	GCLK Control Register 1	GCLK01	PCI Config		Index 041h
13.6.	**Video Input Port Registers**				
13.6.1.	Frame Buffer Address Readback Register	Fb1_Adr		GBase +600000h	Index 000h
13.6.2.	Video Input Port Configuration Register	Vin_Cfg			Index 004h
13.6.3.	Video Input Port Status Register	Vin_Stat			Index 008h
13.6.4.	Video Input Buffer Addr 0	Vin_Ad0			Index 00Ch
13.6.5.	Video Input Buffer Addr 1	Vin_Ad1			Index 010h
13.6.6.	Video Input Desination Pitch	Vin_Dp			Index 014h
13.6.7.	External Timing Generator 1	Vtg_Ext1			Index 028h
13.6.8.	External Timing Generator 2	Vtg_Ext2			Index 02Ch
13.6.9.	Horizontal Timing Generator	Vtg_HT			Index 030h
13.6.10.	Video Timing Generator	Vtg_VT			Index 034h
13.7.	**Video Accelerator Registers**				
13.1.1.	Source Specification Registers		see Note 2	GBase +480000h	
13.8.1.	Video Source Base Register	Video_Src_ Base			Index 00h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

LIST OF REGISTERS

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
13.8.2.	Video Source Pitch Register	Video_Src_ Pitch			Index 04h
13.8.3.	Video Source Dimension Register	Video_Src_ Dim			Index 08h
13.8.4.	CRTC Burst Length Register	CRTC_Burst_ Length			Index 0Ch
13.8.5.	Video Burst Length Register	Video_Burst_ Length			Index 010h
13.9.	Destination Specification Registers				
13.9.1.	Video Destination Register	Video_Dst_XY			Index 014h
13.9.2.	Video Destination Dimension Register	Vid_Dst_ Dim			Index 018h
13.10.	Filter Control Registers				
13.10.1.	Horizontal Scaling and Decimation Register	Horiz_Scl			Index 020h
13.10.2.	Vertical Control and Decimation Register	Vert_Scl			Index 028h
13.10.3.	Color Space Converter Specification Register	Clr_Con_Spec			Index 02Ch
13.11.	Video and Graphics mixing control Registers				
13.11.1.	Mix Mode Register	Mix_Mode			Index 030h
13.11.2.	Color Key Register	CLR_Key			Index 034h
13.11.3.	Chroma Key Low Register	CKL			Index 038h
13.11.4.	Chroma Key High Register	CKH			Index 03Ch
13.11.5.	Status Register	Filter_Stat			Index 040h
14.2.	**Power Management Controller Registers:**			0022h	
14.2.1.	Timer Register 0	Timer0	Configuration	0023h	Index 060h
14.2.2.	Timer Register 1	Timer1	Configuration		Index 061h
14.2.3.	Timer Register 2	Timer2	Configuration		Index 08Dh
14.2.4.	System Activity Enable Register 0	Sys_Activ_en0	Configuration		Index 062h
14.2.5.	System Activity Enable Register 1	Sys_Activ_en1	Configuration		Index 063h
14.2.6.	System Activity Enable Register 2	Sys_Activ_en2	Configuration		Index 064h
14.2.7.	House-Keeping Activity Enable Register 0	HK_Activ_en0	Configuration		Index 065h

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
14.2.8.	House-Keeping Activity Enable Register 1	HK_Activ_en1	Configuration		Index 066h
14.2.9.	Peripheral Inactivity Detection Register 0	Perif_Inact0	Configuration		Index 067h
14.2.10.	Peripheral Activity Detection Register 0	Perif_Act0	Configuration		Index 069h
14.2.11.	Peripheral Activity Detection Register 1	Perif_Act1	Configuration		Index 06Ah
14.2.12.	Address Range 0 Register 0	Add_ Rang0-0	Configuration		Index 06Bh
14.2.13.	Address Range 0 Register 1	Add_ Rang0-1	Configuration		Index 06Ch
14.2.14.	SMI Control Register 0	SMI_Cont0	Configuration		Index 071h
14.2.15.	SMI Status Register 0	SMI_Stat0	Configuration		Index 073h
14.2.16.	SMI Status Register 1	SMI_Stat1	Configuration		Index 074h
14.2.17.	Peripheral Inactivity Status Register 0	Perif_Stat0	Configuration		Index 075h
14.2.18.	Activity Status Register 0	Activ_Stat0	Configuration		Index 077h
14.2.19.	Activity Status Register 1	Activ_Stat1	Configuration		Index 078h
14.2.20.	Activity Status Register 2	Activ_Stat2	Configuration		Index 079h
14.2.21.	PMU State Register	PMU	Configuration		Index 07Ah
14.2.22.	General Purpose Register	GP	Configuration		Index 07Bh
14.2.23.	Clock Control Register 0	Clk_Cont0	Configuration		Index 07Ch
14.2.24.	Doze Timer Read Back Register	Doze	Configuration		Index 088h
14.2.25.	Standby Timer Read Back Register	Standby	Configuration		Index 089h
14.2.26.	Suspend Timer Read Back Register	Suspend	Configuration		Index 08Ah
14.2.27.	House-Keeping Timer Read Back Register	HK_Timer	Configuration		Index 08Bh
14.2.28.	Peripheral Timer Read Back Register	Perif_Timer	Configuration		Index 08Ch

Note 1: X can stand for B (Monochrome Display) or D (Color Display)
Note 2: G_Base can be found in the register CR20 address 3X5h index 20h
Note 3: For Strap Option Settings and Register Description see STPC Client Datasheet.

Table 6-2. CPU Registers located in the ST 486 Manual

Register Name	Mnemonic	Purpose	Address	Access type
Configuration Registers			0022h	
Configuration Control 1	CCR1	IO	0023h	C1h
Configuration Control 2	CCR2	IO		C2h
Configuration Contol 3	CCR3	IO		C3h
SMM Address Region	SMAR	IO		CDH
Device Indentification 0	DIR0	IO		FEh
Device Identification 1	DIR1	IO		FFh

7. DRAM CONTROLLER

7.1. INTRODUCTION

This chapter describes the mapping of the CPU memory and IO address spaces.

The STPC uses a Unified Memory Architecture; the system memory and the graphics buffers use the same memory space. This chapter provides information on the memory address map and the graphics memory usage, together with information on the arbitration logic which resolves accesses to the main memory. Details of memory shadowing and cachability by software control and the Memory Hole for ISA BIOS are also given. The actual interface to the external DRAM modules is presented. Also introduced in this chapter are the PCI configuration space mapping registers, further details are in the chapter relating to the PCI Bus Controller.

7.2. MEMORY CONTROLLER

The STPC handles the memory data (DATA) bus directly, controlling from 2 to 128 MBytes. This main memory is supported using 4 SIMM sockets (Banks 1 to 4) which can be populated with either single or double sided 36-bit (4 bit parity) or 32-bit data SIMMs. Parity is not supported. Four DRAM densities are supported: 1M (256KX4), 4M (512KX4), 16M (4MX4) and 64M (16MX4).

The internal Graphics Controller does not support 32 bit banks, therefore bank 0 SIMMs must be used in pairs, 64 bits wide. Banks 1,2 and 3 can be 32 or 64 bits wide.

Single sided SIMMs or double-sided SIMMs are supported in the following configurations :

256Kx32, 1Mx32, 2Mx32, 4Mx32, 8Mx32 256Kx36, 1Mx36, 2Mx36, 4Mx36, 8Mx36 (parity bits are not used)

The DRAM Controller supports Extended Data Out DRAM (EDO DRAM) as well as Fast Page Mode DRAM (FPM DRAM) - the default DRAM type. Fast Page Mode allows accesses to the same row address to be executed without a RAS# cycle. The column address is latched at the falling edge of CAS#. Read data is valid towards the end of the CAS# low pulse, then the data bus goes to high impedance after CAS# goes high.

EDO DRAMs keep driving read data after CAS# goes high. This allows the data valid time for setup and hold to be overlapped with CAS# precharge. If any of the SIMMs do not have EDO DRAM, then the memory controller will use Fast Page Mode timing. The SIMM type is programmed by software and can be detected on initialization through the memory data pins via a resistor network.

The STPC Memory Controller provides various programmable DRAM parameters to allow the DRAM interface to be optimized for different processor bus speeds and DRAM speed grades.

7.3. MEMORY ADDRESS MAP

7.3.1. 00000000h-0009FFFFh (640K)

Host access maps to the main memory and no ISA or PCI cycle will be initiated. PCI master cycles in this range maps to main memory provided they are not claimed by a PCI Slave. The STPC relies on subtractive decode before initiating an internal memory cycle. ISA master cycles in this range maps to main memory. The STPC will negate IOCHRDY if necessary.

The DMA master cycles in this range maps to main memory. The STPC will actively drive the SD bus during target reads and modify main memory for target write transfers.

This address segment is considered always cacheable in the L1 cache. PCI and ISA master cycles in this range, require the L1 cache.

7.3.2. 000A0000h-000BFFFFh (128K)

This 128K address segment contains the video frame buffer. Normally this address segment is mapped to the DOS frame buffer located in the main memory. However, if VGA is disabled or the VGA memory map mode is such that the VGA does not occupy the entire 128K address range, the host cycle is forwarded to the PCI bus and if not claimed by a PCI slave, it is further forwarded to the ISA bus.

The PCI master cycles in this range, if not claimed by a PCI slave, will be mapped to the main memory or will be forwarded to the ISA bus as per the VGA decode described above.

Similarly, the ISA or DMA master cycles will either map to the main memory or will be forwarded to the PCI. If no PCI slave claims the cycle, the STPC assumes existence of an ISA memory device at this address range.

This segment is never cacheable.

7.3.3. 000C0000h-000C3FFFh (16K)

This 16K address segment can be programmed via Shadow Control register 0 to either map to main memory or expansion busses. Further, reads and writes can have different mappings. If mapped to main memory, this segment will behave as the 0-640K segment.

If not mapped to main memory, a host cycle will first be translated to the PCI cycle and if unclaimed on the PCI bus, will be subtractively decoded and translated to an ISA cycle. A PCI master cycle, if unclaimed by a PCI slave will be forwarded to the ISA bus. An ISA or DMA master cycle, will be translated to the PCI bus and if unclaimed, an ISA memory device at this address range is responsible for the data.

If mapped to the main memory, the cacheability of this address range is controlled by Shadow Control register 3. If mapped to the ISA bus, the ROMCS# signal may optionally be asserted as controlled by Shadow Control register 3. This allows the system and video/peripheral BIOS to physically reside in a single ROM device.

7.3.4. 000C4000h-000C7FFFh (16K)

This range has the same characteristics as that of 000C0000h-000C3FFFh segment as described above. The shadow control for this address range is provided via Shadow Control register 0 and cacheability and ROM chip-select control via Shadow Control register 3.

7.3.5. 000C8000h-000CBFFFh (16K)

This range has the same characteristics as that of 000C0000h-000C3FFFh segment as described above, with the exception of the cacheability attribute. This address range is hardwired to be non-cacheable. Shadow control for this address range is provided via Shadow Control register 0 and ROM chip-select control via Shadow Control register 3.

7.3.6. 000CC000h-000CFFFFh (16K)

This range has the same characteristics as that of 000C8000h-000CBFFFh segment as described above. Shadow control for this address range is provided via Shadow Control register 0 and ROM chip-select control via Shadow Control register 3. This address range is hardwired to be non-cacheable.

7.3.7. 000D0000h-000DFFFFh (64K)

This range has the same characteristics as that of 000CC000h-000CFFFFh segment as described above. Shadow control for this address range is provided via Shadow Control register 1 and can be controlled at 16K resolution. ROM chip-select generation for the entire 64K range can be controlled via Shadow Control register 3. This address range is hardwired to be non-cacheable.

7.3.8. 000E0000h-000EFFFFh (64K)

This range has the same characteristics as that of 000CC000h-000CFFFFh segment as described above. Shadow control for this address range is provided via Shadow Control register 2 and can be controlled at 16K resolution. ROM chip-select generation for the entire 64K range can be controlled via Shadow Control register 3. This address range is hardwired to be non-cacheable.

7.3.9. 000F0000h - 000FFFFFh (64K)

This range has the same characteristics as that of 000C0000h-000C7FFFh segment as described above. Shadow control for this address range is provided via Shadow Control register 3. If not shadowed in the main memory, cycles in this address range which are forwarded to the ISA bus will always results in an ROMCS# assertion. The cacheability of this address segment is controlled via Shadow Control register 3.

7.3.10. 00100000H (1M) - TOP OF ADDRESSABLE DRAM MEMORY

This address segment is mapped to the main memory with the exception of one hole that can optionally be opened in this range via the Memory Hole registers. The address range defined for the hole is mapped to the expansion busses and is described later in this section. The addressable DRAM memory can be different from the populated memory due to the memory remapping and the frame buffer. This is described in more detail in a later section.

With the exception of the memory holes, this address range has the same characteristics as the 0-640K (compatible DOS memory) range.

7.3.11. TOP OF ADDRESSABLE DRAM MEMORY - FFFEFFFFH (4G-64K)

With the exception of memory space allocated to the Extended Graphics (described later), all cycles above the addressable DRAM memory are forwarded to the expansion busses.

Host access in this range initiates a PCI cycle and if unclaimed by a PCI slave, they are forwarded to ISA. Note that the ISA address space is only 16M. Higher addresses are aliased to this 16M space.

If a PCI master access in this range is not claimed by a PCI slave, it will be forwarded to the ISA bus.

An ISA or DMA master cycle is forwarded to the PCI bus and if not claimed by a PCI slave, an ISA memory device is responsible for the data.

7.3.12. FFFF0000 - FFFFFFFFh (4G-64K)

This address segment is an alias of the 64K segment located at F0000h-FFFFFh and has the same attributes except that this segment can never be shadowed into the DRAM memory.

This is also true for address E0000h, D0000h and C0000h provided I/O register Index 51h (see Section 9.6.2.) is set correctly.

7.3.13. EXTENDED GRAPHICS SEGMENT

A 16M segment of memory anywhere between Top of addressable DRAM memory and 256M can be optionally enabled via extended VGA Graphics Registers (GRA). This segment is located at 16M granularity. Refer to the Graphics section for more detailed description of the layout of this memory segment.

Host access to this region is absorbed by the STPC and are either consumed internally or initiate a frame buffer memory access.

PCI master access to this region, if not claimed by a PCI slave is absorbed by the STPC and treated the same way as a host access.

This address range by definition is not accessible to ISA and DMA masters since it must be located at a 16M granularity above the addressable DRAM memory. The ISA and DMA masters can access only up to 16M address range.

This address segment is always considered non-cacheable.

7.3.14. MEMORY HOLE

The Memory Hole register allows the creation of a hole in the memory space in 1-16M address range. This hole allows mapping expansion bus cards in the AT compatible address range when the addressable main memory size exceeds 16M. A host/PCI/ISA/DMA master cycle in this address range is handled in the same way as a cycle above the addressable memory range described above.

7.3.15. SMM MEMORY

The STPC uses the physical memory behind the CPU address range A0000h - B0000h for the SMM memory. The SMM base address register inside CPU needs to be programmed to A0000h. The initialization of the SMM memory is controlled by RAM System management register and redirects the CPU A0000h-B0000h address range to SMM memory. After the initialization, SMM memory can only be accessed when SMIACT# is active. The cacheability of this segment is hardwired to 0.

7.3.16. ADDRESSABLE DRAM MEMORY

Addressable DRAM memory is a function of the size of populated DRAM, the size of graphic memory, the size of memory hole, and the shadow control of D0000h-DFFFFh and E0000h-EFFFFh segments.

TOPM = The size of total physical DRAM is defined by DRAM Bank 3 Register.

TOGM = The size of graphic memory is defined by Graphic memory size register.

MHOLE_SIZE = The size of memory hole defined by Memory Hole Control register.

REMAP_SIZE = 128KB, if none of the 8 x 16KB-segments of D0000h-EFFFFh is enabled for shadow, or 0KB, if any of the 8 x 16KB-segments of D0000h-EFFFFh is enabled for shadow.

The addressable DRAM memory =

TOPM - TOGM + MHOLE_SIZE + REMAP_SIZE

7.3.17. CPU ADDRESS TO DRAM ADDRESS MAPPING

The STPC implements a single memory subsystem for both the system as well as the frame buffer memory. In other words, the size of the DRAM available to the system is reduced by the size of the DRAM allocated to the frame buffer.

The CPU's concept of a physical address is a logical address to the STPC and is remapped to a DRAM physical address. This section refers to the CPU's physical address as the "CPU address" and to the DRAM's physical address as the "DRAM address".

The lower range of the DRAM, starting from the DRAM address 00h, is allocated to frame buffer. The rest of the memory is used by the system. The CPU address is mapped to the DRAM address space above the frame buffer address space. Since the size of the frame buffer can vary and is controlled by the Graphics Memory Size Register (Index 36 of the STPC configuration registers).

STPC also defines a memory hole to allow the existence of memory devices on the PCI or ISA busses. The size of the CPU address space is increased by these memory holes, if they exist. CPU address space D0000h to EFFFFh is mapped to the add-in card BIOS area. If this ROM space is not shadowed, then the CPU address space is increased by another 128 KBytes (also see Section 7.6.1.).

For example:

Total populated DRAM = 4 MBytes

Frame buffer size = 256 KBytes

Memory hole size = 1 MByte

Memory hole starting address = 200000h

Shadow feature for D0000h to EFFFFh = disabled

The total CPU memory = 4 MBytes - 256 KBytes + 1 MByte + 128 KBytes = 4 MBytes plus 896 KBytes

Since the frame buffer is 256 KBytes, the system memory is reduced by 256 KBytes and becomes 3 MBytes plus 768 KBytes. Since a 1 MByte memory hole exists, the CPU address space is increased by 1 MByte and becomes 4 MBytes plus 768 KBytes. The CPU address between 3 MBytes plus 768 KBytes and 1 MBytes above this is map- ped to the memory hole.

Since the shadowing of the CPU address range D0000h to EFFFFh reserved for add-on card BIOS is not enabled, the CPU memory is increased by 128 KBytes to make use of this DRAM space that no device accesses. The total CPU memory then becomes 4 MBytes plus 896 KBytes.

7.4. IO ADDRESS MAP

The STPC implements a number of registers in IO address space.

These register occupy the map in the IO space in the Table 7-2 below.

7.4.1. PCI CONFIGURATION ADDRESS MAP

The STPC occupies Device number 0 slot on the PCI bus and implements a number of registers in PCI configuration address space. These registers occupy the following map (see Table 7-1):

Offset	Description
00h-01h	Vendor Identification register
02h-03h	Device Identification register
04h-05h	PCI Command register
06h-07h	PCI Status register
08h	PCI Revision ID register
40h	PCI Control register

Table 7-1. PCI configuration address space

IO address	Description	Notes
0000h-000Fh	8237 DMA controller 1 registers.	1
0020h-0021h	8259 Interrupt controller 1 registers.	
0022h	STPC specific configuration registers index port	
0023h	STPC specific configuration registers data port	
0040h-0043h	8254 Timer/Counter registers.	1
0060h-0064h	Keyboard shadow registers.	1
0070h-0071h	NMI Mask control registers.	1
0080h-008Fh	DMA Page registers.	
0094h	Mother-board VGA enable.	2
00A0h-00A1h	8259 Interrupt controller 2 registers.	1
061h	ISA standard Port B.	1
00C0h-00DFh	8237 DMA controller 2 registers.	1
0102h	VGA setup register.	
03B4h,03B5h,03BAh	VGA registers.	
03D4h,03D5h,03DAh		
03C0h-03CFh		
0CF8h	PCI configuration Address register.	
0CFCh-0CFFh	PCI configuration Data register.	
46E8h	VGA add-in mode enable register.	2

Note 1: This address range is partially decoded. Refer to the Register Description section for more details.

Note 2: This address is occupied only if the STPC is strapped to look like a mother-board VGA.

Table 7-2. IO map space

7.5. CACHE RELATED REGISTERS

7.5.1. CACHE ARCHITECTURE REGISTER 0

This register controls various attributes of the L2 cache.

Cash_Arc0 Access = 0022h/0023h Regoffset = 020h

7	6	5	4	3	2	1	0
CPU PAS	BAO	L1 WB	SRAM		L2 B	L2 WBC	L2 BC
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**CPU PAS**	**CPU pipelined access support.** (see table below:Table 7-3)
Bit 6	**BAO**	**Burst addressing order.** (see table below: Table 7-4)
Bit 5	**L1 WB**	**L1 write back indication.** (see table below: Table 7-5)
Bits 4-3	**SRAM**	**SRAM Type.** These bist control the type of SRAMs used to construct L2 cache. (see table below: Table 7-6)
Bit 2	**L2 B**	**Number of L2 banks.** When programmed to 2 banks, L2 interleaving is enabled (see table below: Table 7-7)
Bit 1	**L2 WBC**	**L2 write back control.** (see table below: Table 7-8)
Bit 0	**L2 BC**	**L2 cache enable**

Bit 7	CPU pipelined access
0	not supported
1	supported

Table 7-3. CPU pipelined access

Bit 6	Burst order
0	Intel
1	linear

Table 7-4. Burst order

Bit 5	L1 write back
0	Not supported
1	Supported

Table 7-5. L1 write back

Bit 4	Bit 3	L2 cache SRAM type
0	0	asynchronous SRAM
0	1	synchronous burst SRAM

Table 7-6. L2 cache SRAM type

Bit 4	Bit 3	L2 cache SRAM type
1	0	synchronous burst pipelined SRAM
1	1	reserved

Table 7-6. L2 cache SRAM type

Bit 2	L2 Banks
0	One bank
1	Two banks

Table 7-7. L2 Banks

Bit 1	L2 write back control
0	write through
1	write back

Table 7-8. L2 write back control

.

Bit 0	L2 cache
0	disabled
1	enabled

Table 7-9. L2 cache enable

7.5.2. CACHE ARCHITECTURE REGISTER 1

This register controls various attributes of L2 cache.

Cash_Arc1 Access = 0022h/0023h Regoffset = 021h

7	6	5	4	3	2	1	0
L2 CS			IO NA	S FIFO		R AWE	Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-5	**L2 CS**	**L2 cache size.** (see table below: Table 7-10)
Bit 4	**IO NA#Enable**	**IO NA#ENABLE.** (see table below: Table 7-11)
Bits 3-2	**S FIFO**	**Source FIFO low water mark.** These bits control the degree of concurrency between a L1 cache line fill and start of the next memory access. A cache line wide read buffer is implemented. Due to pipelining, it is possible that the buffer may be filled up ahead of drain. Then if the next access is also a read from memory, these bits determine when the next read will be kicked off relative to the drain of the current line from the read buffer. The optimal value is a function of the drain rate of the buffer which depends on the cache RAM type and the programmed burst parameters. A value of '0' for this field is the least optimal value but will always work. (see table below: Table 7-12)
Bit 1	**R AWE**	**Read around write enable.** (see table below: Table 7-13)
Bit 0	**Rsv**	**Reserved**

Bit 7	Bit 6	Bit 5	L2 Cache Size
0	0	0	64Kb
0	0	1	128Kb
0	1	0	256Kb
0	1	1	512Kb
1	0	0	1 MB
1	0	1	2 MB

Table 7-10. L2 Cache Size

Bit 4	NA# generation during IO cycles
0	generate NA#
1	Don't generate NA#

Table 7-11. NA# generation during IO cycles

Bit 3	Bit 2	Start next read...
0	0	only after completely finishing current fill
0	1	when 1 QWORD is still to be emptied
1	0	when 2 QWORDs are still to be emptied
1	1	when 3 QWORDs are still to be emptied

Table 7-12. Start next read...

Bit 1	Read around write enable
0	reads can not proceed around any posted writes
1	reads can go around a posted write if it is to a different address to the posted writes

Table 7-13. Read around write enable

7.5.3. CACHE ARCHITECTURE REGISTER 2

Cash_Arc2 Access = 0022h/0023h Regoffset = 022h

7	6	5	4	3	2	1	0
Rsv	SHDD	CWEPW	CDHAWE	BAWS		TAWS	
Default value after reset = FFh							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved**
Bit 6	**SHDD**	**Slow host data driver.** (see table below: Table 7-14)
Bit 5	**CWEPW**	**Cache write enable pulse width.** Applicable to asynchronous SRAMs only. Must be '0' for synchronous SRAMs.(see table below: Table 7-15)
Bit 4	**CDHAWE**	**Cache data hold after write enable.** Must be a '1' if 1.5 clocks wide write enable pulse width is selected via bit 5 above. (see table below:Table 7-16)
Bits 3-2	**BAWS**	**Burst access wait states.** (see table below: Table 7-17)
Bist 1-0	**TAWS**	**Tag access wait states.** (see table below: Table 7-18)

Bit 6	Host data bus driver
0	Slow, two clocks to drive HD bus
1	Fast, One clock to drive the HD bus

Table 7-14. Host data bus driver

Bit 5	Cache write enable pulse
0	1.5 clock wide
1	1 clock wide

Table 7-15. Cache write enable pulse

Bit 4	Cache data hold
0	data is kept valid for 1 extra clock after write enable
1	data removed in the same clock as write enable trailing edge

Table 7-16. Cache data hold

Bit 3	Bit 2	Burst access wait states
0	0	fastest
0	1	1 clock slower than fastest
1	0	2 clocks slower than fastest
1	1	3 clocks slower than fastest

Table 7-17. Burst access wait states

Bit 1	Bit 0	Tag access wait states
0	0	fastest
0	1	1 clock slower than fastest
1	0	2 clocks slower than fastest
1	1	3 clocks slower than fastest

Table 7-18. Tag access wait states

7.6. ADDRESS DECODE RELATED REGISTERS

The following registers are all 8-bit. They are accessed by setting the Configuration Index Port (22h) to the Configuration Index (C.I.) shown, and then reading or writing the appropriate values from the Configuration Register Data Port (23h).

7.6.1. MEMORY HOLE CONTROL REGISTER

This 8-bit register defines the enable, size, and starting address of memory hole. Any memory accesses to this memory hole are directed to PCI/ISA bus.

Mem_Hole Access = 0022h/0023h Regoffset = 024h

7	6	5	4	3	2	1	0
MHE	MHS			MHSA			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	MHE	**Memory Hole Enable.** This bit controls the enable of memory hole function.(see table below: Table 7-19).
Bits 6-4	MHS	**Memory Hole Size.** These bits control the size of memory hole (see table below: Table 7-20).
Bits 3-0	MHSA	**Memory Hole Start Address.** These bits control the bits 23-20 of the memory hole starting address. The memory hole starting address must be aligned to the hole size.

Bit 7	Memory Hole Enable
0	disabled
1	enabled

Table 7-19. Memory Hole Enable

Bit 6	Bit 5	Bit 4	Memory Hole Size
0	0	0	1 MB
0	0	1	2 MB
0	1	1	4 MB
1	1	1	8 MB
others			reserved

Table 7-20. Memory Hole Size

Programming notes

This memory hole is also non-cacheable.

7.6.2. SHADOW CONTROL REGISTER 0

This 8-bit register controls the read/write attributes of the memory located at C0000h-CFFFFh. Each 16k of the whole 64k is controlled by 2 bits, one for read and one for write.

Shadow_0 Access = 0022h/0023h Regoffset = 025h

7	6	5	4	3	2	1	0
RC1	WC1	RC2	WC2	RC3	WC3	RC4	WC4
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**RC1**	**Read Control CC000h-CFFFFh.** This bit controls the read attribute of the CC000h-CFFFFh memory. (see table below: Table 7-21)
Bit 6	**WC1**	**Write Control CC000h-CFFFFh.** This bit controls the write attribute of the CC000h-CFFFFh memory. (see table below: Table 7-22)
Bit 5	**RC2**	**Read Control C8000h-CBFFFh.** This bit controls the read attribute of the C8000h-CBFFFh memory. (see table below: Table 7-23)
Bit 4	**WC2**	**Write Control C8000h-CBFFFFh.** This bit controls the write attribute of the C8000h-CBFFFh memory. (see table below: Table 7-24)
Bit 3	**RC3**	**Read Control C4000h-C7FFFh.** This bit controls the read attribute of the C4000h-C7FFFh memory. (see table below: Table 7-25)
Bit 2	**WC3**	**Write Control C4000h-C7FFFFh.** This bit controls the write attribute of the C4000h-C7FFFh memory. (see table below: Table 7-26)
Bit 1	**RC4**	**Read Control C0000h-C3FFFh.** This bit controls the read attribute of the C0000h-C3FFFh memory. (see table below: Table 7-27)
Bit 0	**WC4**	**Write Control C0000h-C3FFFFh.** This bit controls the write attribute of the C0000h-C3FFFFh memory. (see table below: Table 7-28)

Bit 7	Read Control CC000h-CFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-21. Read Control CC000h-CFFFFh

Bit 6	Write Control CC000h-CFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-22. Write Control CC000h-CFFFFh

Bit 5	Read Control C8000h-CBFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-23. Read Control C8000h-CBFFFh

Bit 4	Write Control C8000h-CBFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-24. Write Control C8000h-CBFFFFh

Bit 3	Read Control C4000h-C7FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-25. Read Control C4000h-C7FFFh

Bit 2	Write Control C4000h-C7FFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-26. Write Control C4000h-C7FFFFh

Bit 1	Read Control C0000h-C3FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-27. Read Control C0000h-C3FFFh

Bit 0	Memory Hole Size
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-28. Write Control C0000h-C3FFFFh

Programming Notes

There is single cacheability bit for the 32k Video BIOS segment (C0000h-C7FFFh) located in Shadow Control register 2. C7FFFFh-CFFFFh segment has the cacheability bit hardwired to ‘1’ (enabled). If shadow is enabled for read/write cycles, read from and write to this area are directed to the system memory. Or else the cycles are forwarded to the expansion busses.

7.6.3. SHADOW CONTROL REGISTER 1

Similar to Shadow Control Register 0, this 8-bit register controls the read/write attributes of the memory located at D0000h-DFFFFh.

Shadow_1 Access = 0022h/0023h Regoffset = 026h

7	6	5	4	3	2	1	0
SRC	SWC	SWC	SWC	SRC	SWC	SRC	SWC
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**SRC**	**Shadow Read Control DC000h-DFFFFh.** This bit controls the read attribute of the DC000h-DFFFFh memory (see table below: Table 7-29).
Bit 6	**SWC**	**Shadow Write Control DC000h-DFFFFh.** This bit controls the write attribute of the DC000h-DFFFFh memory (see table below: Table 7-30).
Bit 5	**SWC**	**Shadow Write Control D8000h-DBFFFh.** This bit controls the read attribute of the D8000h-DBFFFh memory (see table below: Table 7-31).
Bit 4	**SWC**	**Shadow Write Control D8000h-DBFFFh.** This bit controls the write attribute of the D8000h-DBFFFh memory (see table below: Table 7-32).
Bit 3	**SRC**	**Shadow Read Control D4000h-D7FFFh.** This bit controls the read attribute of the D4000h-D7FFFh memory (see table below: Table 7-33).
Bit 2	**SWC**	**Shadow Write Control D4000h-D7FFFh.** This bit controls the write attribute of the D4000h-D7FFFh memory (see table below: Table 7-34).
Bit 1	**SRC**	**Shadow Read Control D0000h-D3FFFh.** This bit controls the read attribute of the D0000h-D3FFFh memory (see table below: Table 7-35).
Bit 0	**SWC**	**Shadow Write Control D0000h-DFFFFh.** This bit controls the write attribute of the D0000h-DFFFFh memory (see table below: Table 7-36).

Bit 7	Shadow Read Control DC000h-DFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-29. Shadow Read Control DC000h-DFFFFh

Bit 6	Shadow Write Control DC000h-DFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-30. Shadow Write Control DC000h-DFFFFh

Bit 5	Shadow Write Control D8000h-DBFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-31. Shadow Write Control D8000h-DBFFFh

Bit 4	Shadow Write Control D8000h-DBFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-32. Shadow Write Control D8000h-DBFFFh

Bit 3	Shadow Write Control D4000h-D7FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-33. Shadow Write Control D4000h-D7FFFh

Bit 2	Shadow Write Control D4000h-D7FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-34. Shadow Write Control D4000h-D7FFFh

Bit 1	Shadow Read Control D0000h-D3FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-35. Shadow Read Control D0000h-D3FFFh

Bit 0	Shadow Write Control D0000h-DFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-36. Shadow Write Control D0000h-DFFFFh

Programming Notes

This entire 64K segment has the cacheability bit hardwired to ‘0’ (disabled).

7.6.4. SHADOW CONTROL REGISTER 2

Similar to Shadow Control Register 0, this 8-bit register controls the read/write attributes of the memory located at E0000h-EFFFFh.

Shadow_2 Access = 0022h/0023h Regoffset = 027h

7	6	5	4	3	2	1	0
RC	WC	RC	WC	RC	WC	RC	WC
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**RC**	**Read Control EC000h-EFFFFh.** This bit controls the read attribute of the EC000h-EFFFFh memory (see table below: Table 7-37).
Bit 6	**WC**	**Write Control EC000h-EFFFFh.** This bit controls the write attribute of the EC000h-EFFFFh memory (see table below: Table 7-38).
Bit 5	**RC**	**Read Control E8000h-EBFFFh.** This bit controls the read attribute of the E8000h-EBFFFh memory (see table below: Table 7-39).
Bit 4	**WC**	**Write Control E8000h-EBFFFh.** This bit controls the write attribute of the E8000h-EBFFFh memory (see table below: Table 7-40).
Bit 3	**RC**	**Read Control E4000h-E7FFFh.** This bit controls the read attribute of the E4000h-E7FFFh memory (see table below: Table 7-41).
Bit 2	**WC**	**Write Control E4000h-E7FFFh.** This bit controls the write attribute of the E4000h-E7FFFh memory (see table below: Table 7-42).
Bit 1	**RC**	**Read Control E0000h-E3FFFh.** This bit controls the read attribute of the E0000h-E3FFFh memory (see table below: Table 7-43).
Bit 0	**WC**	**Write Control E0000h-EFFFFh.** This bit controls the write attribute of the E0000h-EFFFFh memory (see table below: Table 7-44).

Bit 7	Read Control EC000h-EFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-37. Read Control EC000h-EFFFFh

Bit 6	Write Control EC000h-EFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-38. Write Control EC000h-EFFFFh

Bit 5	Read Control EC000h-EFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-39. Read Control EC000h-EFFFFh

Bit 4	Write Control E8000h-EBFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-40. Write Control E8000h-EBFFFh

Bit 3	Read Control E4000h-E7FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-41. Read Control E4000h-E7FFFh

Bit 2	Write Control E4000h-E7FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-42. Write Control E4000h-E7FFFh

Bit 1	Read Control E0000h-E3FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-43. Read Control E0000h-E3FFFh

Bit 0	Read Control EC000h-EFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-44. Read Control E0000h-EFFFFh

Programming Notes

This entire 64K segment has the cacheability bit hardwired to ‘0’ (disabled).

7.6.5. SHADOW CONTROL REGISTER 3

This 8-bit register controls the cacheability attributes of C0000h-C7FFFh and F0000h-FFFFFh shadow segments.

Shadow_3 Access = 0022h/0023h Regoffset = 028h

7	6	5	4	3	2	1	0
SMRAM	CCF	CCC	Rsv			RCF	WCF
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**SMRAM**	**SMRAM Initialization Enable.** This bit controls whether CPU accesses in A0000h-BFFFFh address range are decoded as VGA frame buffer access or SMRAM access. The STPC allows for 128KBytes of SMRAM. Physically this memory is located in the system memory behind the higher address range. This area of the system memory is normally unused since this address range is normally mapped to frame buffer which has its own memory. When the CPU is operating in SMM, accesses in the range of A0000-BFFFFh goes to SMRAM instead of VGA frame buffer. The rest of the address map remains unchanged. The address range A0000h-BFFFFh is always non-cacheable (see table below: Table 7-45).
Bit 6	**CCF**	**Cache Control F0000h-FFFFFh.** This bit controls the cacheability of F0000h-FFFFFh block when the shadow function is enabled (see table below: Table 7-46).
Bit 5	**CCC**	**Cache Control C0000h-C7FFFh.** This bit controls the cacheability of C0000h-C7FFFh block when the shadow function is enabled (see table below: Table 7-47).
Bits 4-2	**Rsv**	**Reserved.**
Bit 1	**RCF**	**Read Control F0000h-FFFFFh.** This bit controls the read attribute of F0000h-FFFFFh memory (see table below: Table 7-48).
Bit 0	**WCF**	**Write Control F0000h-FFFFFh.** This bit controls the write attribute of F0000h-FFFFFh memory (see table below: Table 7-49).

Bit 7	SMRAM Initialization Enable
0	A0000h-BFFFFh is interpreted as VGA frame buffer access
1	A0000h-BFFFFh is interpreted as SMRAM access

Table 7-45. SMRAM Initialization Enable

Bit 6	Cache Control F0000h-FFFFFh
0	cacheability disabled
1	cacheability enabled

Table 7-46. Cache Control F0000h-FFFFFh

Bit 5	Cache Control C0000h-C7FFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-47. Cache Control C0000h-C7FFFh

Bit 1	Read Control F0000h-FFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-48. Read Control F0000h-FFFFFh

Bit 0	Write Control F0000h-FFFFFh
0	shadow disabled for read cycle
1	shadow enabled for read cycle

Table 7-49. Write Control F0000h-FFFFFh

Programming notes

The rest of the shadow RAM segments have the cacheability bits hardwired to ‘0’ (disabled). This register also provides control over the address range for which ROM chip-select (ROMCS#) will be asserted allowing various BIOSes (system, video, disk etc.) to be implemented in a single part. Bit 7 of this register also provides accessibility to the SMM mode RAM (SMRAM).

7.6.6. VGA DECODE REGISTER

This 8-bit register controls address decode for the internal VGA as follows:

VGA_Dec Access = 0022h/0023h Regoffset = 029h

7	6	5	4	3	2	1	0
Rsv		GC	PC	DC	PSE	I VGA D	ADE
Default value after reset = 03h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**
Bit 5	**GC**	**stop g-clock** (Graphics clock).
Bit 4	**PC**	**stop p-clock.** (PCI Clock)
Bit 3	**DC**	**stop d-clock** (Dot clock).
Bit 2	**PSE**	**Palette Snoop Enable.** (see table below: Table 7-50).
Bit 1	**I VGA D**	**Internal VGA Disable.** This bit if set to a ‘0’ will disable internal VGA. Otherwise if set to a ‘1’, it will enable the internal VGA.
Bit 0	**ADE**	**Add-in Decode Enable.** This bit if set to a ‘0’ will map the internal VGA to add-in card address space. Otherwise if set to a ‘1’ it will map the VGA to mother-board address space.

Bit 2	Palette Snoop Enable
0	Palette write cycles are propagated to PCI bus in addition to updating the internal palette.
1	Palette write cycles are terminated internally and are not propagated to PCI.

Table 7-50. Palette Snoop Enable

7.7. DRAM CONTROLLER REGISTERS

The STPC manages 4 SIMM sockets which can be populated with either single or double sided 36-bit (4 bit parity) or 32-bit data SIMMs. Four DRAM densities are supported: 1M (256KX4), 4M (1MX4), 16M (4MX4) and 64M (16MX4). Although the system DRAM data bus is 64-bit wide, 32-bit DRAM banks are also supported by not populating the upper DWORD SIMM module for that particular bank. However Bank 0 must always be populated to 64-bit when the integrated Graphics Controller is enabled.

Configuration registers 30-33 provide the top addresses for each bank. Any bank can be skipped by the top addresses of two consecutive banks having the same address.

7.7.1. DRAM BANK 0 REGISTER

This 8-bit register controls the top address of the DRAM bank 0. Register bit 7-0 corresponding to memory address bits 27-20.

DRAM_B0 Access = 0022h/0023h Regoffset = 030h

7	6	5	4	3	2	1	0
Default value after reset = 0x07h							

Bank 0 Top Address = SIMM0 size in MB + SIMM1 size in MB -1.

This register defaults to 07h.

Example 1:

SIMM0 = 4MB, SIMM1 = 4MB

Bank 0 Top Address = 2x4 - 1 = 7 = 07h

Bank 1, 2, 3 Top Address = 07h

DRAM__B0 Access = 0022h/0023h Regoffset = 030h

7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1

Example 2:

SIMM0 = 32MB (dbl. sided), SIMM1 = 32MB (dbl. sided)

Bank 0 Top Address = 2x16 - 1 = 31 = 1Fh

Bank 1 Top Address = 32 + 2x16 - 1 = 63 = 3Fh

Bank 2, 3 Top Address = 3Fh

DRAM__B0 Access = 0022h/0023h Regoffset = 030h

7	6	5	4	3	2	1	0
0	0	1	1	1	1	1	1

Example 3:

SIMM0 = 32MB (dbl. sided), SIMM1 = 32MB (dbl. sided), SIMM2 = 16MB, SIMM3 = 16MB

Bank 0 Top Address = 2x16 - 1 = 31 = 1Fh

Bank 1 Top Address = 32 + 2x16- 1 = 63 = 3Fh

Bank 2 Top Address = 64 + 2x16 - 1 = 95= 5Fh

Bank 3 Top Address = 5Fh

DRAM__B0 Access = 0022h/0023h Regoffset = 030h

7	6	5	4	3	2	1	0
0	1	0	1	1	1	1	1

7.7.2. DRAM BANK 1 REGISTER

This register controls the top address of DRAM bank 1.

DRAM_B1 Access = 0022h/0023h Regoffset = 031h

7	6	5	4	3	2	1	0
Default value after reset = 0x07h							

7.7.3. DRAM BANK 2 REGISTER

This register controls the top address of DRAM bank 2.

DRAM_B2 Access = 0022h/0023h Regoffset = 032h

7	6	5	4	3	2	1	0
Default value after reset = 0x07h							

7.7.4. DRAM BANK 3 REGISTER

This register controls the top address of DRAM bank 3.

DRAM_B3 Access = 0022h/0023h Regoffset = 033h

7	6	5	4	3	2	1	0
Default value after reset = 0x07h							

7.7.5. MEMORY BANK WIDTH REGISTER

Each memory bank can have one or two 32 bit SIMMs causing the memory width for that bank to be 32 or 64 bits.

Mem_Width Access = 0022h/0023h Regoffset = 034h

7	6	5	4	3	2	1	0
Uns				MB 3	MB 2	MB 1	MB 0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	**Uns**	**Unused**
Bit 3	**MB 3**	**Memory Bank 3 Width Code** (see table below: Table 7-51)
Bit 2	**MB 2**	**Memory Bank 2 Width Code** (see table below: Table 7-51)
Bit 1	**MB 1**	**Memory Bank 1 Width Code** (see table below: Table 7-51)
Bit 0	**MB 0**	**Memory Bank 0 Width Code** (see table below: Table 7-51)

Code	Memory Bank Width
0	64 bits (default)
1	32 bits

Table 7-51. Memory Bank Width

7.7.6. DRAM BANK 0 TIMING PARAMETER REGISTER

This register controls RAS# and CAS# timing for RAS bank 0.

DRAM_T0 Access = 0022h/0023h Regoffset = 035h

7	6	5	4	3	2	1	0
M RAS	Rsv	B0 RAS		B0 RAS/CAS		B0 CAS	
Default value after reset = 80h							

Bit Number	Mnemonic	Description
Bit 7	**M RAS**	**Main RAS Active.** This bit controls if RAS is kept active after the current DRAM access. It applies to all banks in main memory space (see table below: Table 7-52).
Bit 6	**Rsv**	**Reserved.**
Bits 5-4	**B0 RAS**	**Bank 0 RAS Precharge.** RAS precharge timing for Bank 0 (see table below: Table 7-53).
Bits 3-2	**B0 RAS/CAS**	**Bank 0 RAS/CAS Delay.** RAS to CAS delay timing for Bank 0 (see table below: Table 7-54).
Bits 1-0	**B0 CAS**	**Bank 0 CAS Pulse Width**. These bits control the CAS pulse width (see table below: Table 7-55).

Bit 7	RAS active
0	keep RAS# active
1	deassert RAS#

Table 7-52. RAS active

Bit 5	Bit 4	RAS# precharge time
0	1	Reserved
1	0	2 cycles
1	1	3 cycles
0	0	4 cycles

Table 7-53. Bank 0 RAS precharge time

Bit 3	Bit 2	RAS# to CAS# delay
0	1	Reserved
1	0	2 cycles
1	1	3 cycles
0	0	4 cycles

Table 7-54. Bank 0 RAS/CAS delay

Bit 1	Bit 0	CAS# low pulse width
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles
0	0	4 cycles

Table 7-55. Bank 0 CAS low pulse width

7.7.7. DRAM BANK 1 TIMING PARAMETER REGISTER

This register controls RAS# and CAS# timing for RAS bank 1.

DRAM_T1 Access = 0022h/0023h Regoffset = 038h

7	6	5	4	3	2	1	0
Rsv		B1 RAS		B1 RAS/CAS		B1 CAS	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**
Bits 5-4	**B1 RAS**	**Bank 1 RAS Precharge.** These bits control the RAS precharge timing (see table below: Table 7-56).
Bits 3-2	**B1 RAS/CAS**	**Bank 1 RAS/CAS Delay.** These bits control the RAS to CAS delay timing (see table below: Table 7-57).
Bits 1-0	**B1 CAS**	**Bank 1 CAS Pulse Width.** These bits control the CAS pulse width (see table below: Table 7-58).

Bit 5	Bit 4	RAS# precharge time
0	1	Reserved
1	0	2 cycles
1	1	3 cycles
0	0	4 cycles

Table 7-56. Bank 1 RAS precharge time

Bit 3	Bit 2	RAS# to CAS# delay
0	1	Reserved
1	0	2 cycles
1	1	3 cycles
0	0	4 cycles

Table 7-57. Bank 1 RAS/CAS delay

Bit 1	Bit 0	CAS# pulse width
0	1	1 cycles
1	0	2 cycles
1	1	3 cycles
0	0	4 cycles

Table 7-58. Bank 1 CAS pulse width

7.7.8. DRAM BANK 2 TIMING PARAMETER REGISTER

This register controls RAS# and CAS# timing for RAS bank 2.

DRAM_T2 Access = 0022h/0023h Regoffset = 03Ah

7	6	5	4	3	2	1	0
Rsv		B2 RAS		B2 RAS/CAS		B2 CAS	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**
Bits 5-4	**B2 RAS**	**Bank 2 RAS Precharge.** These bits control the RAS precharge timing (see table below: Table 7-59).
Bits 3-2	**B2 RAS/CAS**	**Bank 2 RAS/CAS Delay.** These bits control the RAS to CAS delay timing (see table below: Table 7-60).
Bits 1-0	**B2 CAS**	**Bank 2 CAS Pulse Width.** These bits control the CAS pulse width (see table below: Table 7-61).

Bit 5	Bit 4	Bank 2 RAS# precharge time
0	1	Reserved
1	0	RAS# precharge time 2 cycles
1	1	RAS# precharge time 3 cycles
0	0	RAS# precharge time 4 cycles

Table 7-59. Bank 2 RAS precharge time

Bit 3	Bit 2	Bank 2 RAS# to CAS# delay
0	1	Reserved
1	0	RAS# to CAS# delay 2 cycles
1	1	RAS# to CAS# delay 3 cycles
0	0	RAS# to CAS# delay 4 cycles

Table 7-60. Bank 2 RAS/CAS delay

Bit 1	Bit 0	Bank 2 CAS pulse width
0	1	CAS# low pulse width 1 cycles
1	0	CAS# low pulse width 2 cycles
1	1	CAS# low pulse width 3 cycles
0	0	CAS# low pulse width 4 cycles

Table 7-61. Bank 2 CAS pulse width

7.7.9. DRAM BANK 3 TIMING PARAMETER REGISTER

This register controls RAS# and CAS# timing for RAS bank 3.

DRAM_T3 Access = 0022h/0023h Regoffset = 03Bh

7	6	5	4	3	2	1	0
Rsv		B3 RAS		B3 RAS/CAS		B3 CAS	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**
Bits 5-4	**B3 RAS**	**Bank 3 RAS Precharge.** These bits control the RAS precharge timing (see table below: Table 7-62).
Bits 3-2	**B3 RAS/CAS**	**Bank 3 RAS/CAS Delay.** These bits control the RAS to CAS delay timi-ng (see table below: Table 7-63).
Bits 1-0	**B3 CAS**	**Bank 3 CAS Pulse Width.** These bits control the CAS pulse width (see table below: Table 7-64).

Bit 5	Bit 4	Bank 3 RAS precharge time
0	1	Reserved
1	0	RAS# precharge time 2 cycles
1	1	RAS# precharge time 3 cycles
0	0	RAS# precharge time 4 cycles

Table 7-62. Bank 3 RAS precharge time

Bit 3	Bit 2	Bank 3 RAS/CAS delay
0	1	Reserved
1	0	RAS# to CAS# delay 2 cycles
1	1	RAS# to CAS# delay 3 cycles
0	0	RAS# to CAS# delay 4 cycles

Table 7-63. Bank 3 RAS/CAS delay

Bit 1	Bit 0	Bank 3 CAS pulse width
0	1	CAS# low pulse width 1 cycles
1	0	CAS# low pulse width 2 cycles
1	1	CAS# low pulse width 3 cycles
0	0	CAS# low pulse width 4 cycles

Table 7-64. Bank 3 CAS Pulse Width

7.7.10. GRAPHICS MEMORY SIZE REGISTER

This register defines the size of DRAM used by the graphics for frame buffer.

Graph_Mem Access = 0022h/0023h Regoffset = 036h

7	6	5	4	3	2	1	0
G RAS	Rsv	T GM					
Default value after reset = 04h							

Bit Number	Mnemonic	Description
Bit 7	**G RAS**	**Graphics RAS Active.** This bit controls if RAS is kept active after the current framebuffer DRAM access (see table below: Table 7-65).
Bit 6	**Rsv**	**Reserved**
Bits 5-0	**T GM**	**Top of Graphics Memory.** This indicates frame buffer size in 128KB units. The range is 0 to 32 for 0 to 4MB framebuffer, so 6 bits are necessary.

Bit 7	Graphics RAS Active
0	keep RAS# active
1	deassert RAS#

Table 7-65. Graphics RAS Active

7.7.11. MEMORY TYPE REGISTER

Mem_Type Access = 0022h/0023h Regoffset = 037h

7	6	5	4	3	2	1	0
B3 TC		B2 TC		B1 TC		B0 TC	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**B3 TC**	Bank 3 type code (see table below: Table 7-66).
Bits 5-4	**B2 TC**	Bank 2 type code (see table below: Table 7-66).
Bits 3-2	**B1 TC**	Bank 1 type code (see table below: Table 7-66).
Bits 1-0	**B0 TC**	Bank 0 type code (see table below: Table 7-66).

Code	Memory Type
00	Fast Page Mode (default)
01	Extended Data Out
10	Reserved
11	Reserved

Table 7-66. Memory type

7.7.12. DRAM REFRESH REGISTER

The register refresh disable bit also contains a number of host clocks for the DRAM refresh interval.

DRAM_Ref Access = 0022h/0023h Regoffset = 039h

7	6	5	4	3	2	1	0
RE	RC						
Default value after reset = 30h							

Bit Number	Mnemonic	Description
Bit 7	**RE**	**Refresh Enable.** This bit must be programmed to ‘0’ for normal operation
Bits 6-0	**RC**	**Refresh Cycle.** (HCLK frequency in MHz * 15.6us) >> 4

Examples: (rounded down to nearest integer)

round_down((75MHz * 15.6us) >> 4) = 73 = 49h

round_down((66MHz * 15.6us) >> 4) = 65 = 41h

round_down((60MHz * 15.6us) >> 4) = 58 = 3Ah

round_down((50MHz * 15.6us) >> 4) = 48 = 30h

Programming notes

The refresh interval should be reset to the smallest likely run time value (typically 48 HCLKs) to provide warm up cycles for the DRAM.

A refresh request is generated whenever this register is written to without setting the refresh enable bit.

7.8. DRAM INTERFACE

The STPC provides MA, RAS#, CAS#, WE# and MD for DRAM control. From 2 to 128 MBytes of main memory are supported in 1 to 4 banks. Banks 1,2 and 3 can be 32 or 64 bits wide. Bank 0 must be 64 bits wide since the graphics controller does not support 32 bit banks.

The following SIMMs are supported:

256Kx32, 1Mx32, 2Mx32, 4Mx32, 8Mx32

256Kx36, 1Mx36, 2Mx36, 4Mx36, 8Mx36 (parity bits not used).

The following picture, Figure 7-1, shows the DRAM organization. The DRAM interface is organized into four banks(RAS#[3:0]). Each row is eight Bytes wide (CAS#[7:0]) when both SIMMs of the same row are populated. If both SIMMs of the same row were populated, the DRAMs need to be of same densities. Banks 1, 2 and 3 are allowed to have only one SIMM populated. When only one SIMM is populated, it needs to be the lower one(CAS#[3:0], CAS#[7:4] = Fh, MD[31:0], MD[63:32] = FFFFFFFFh). MA, and WE# goes to all SIMMs.

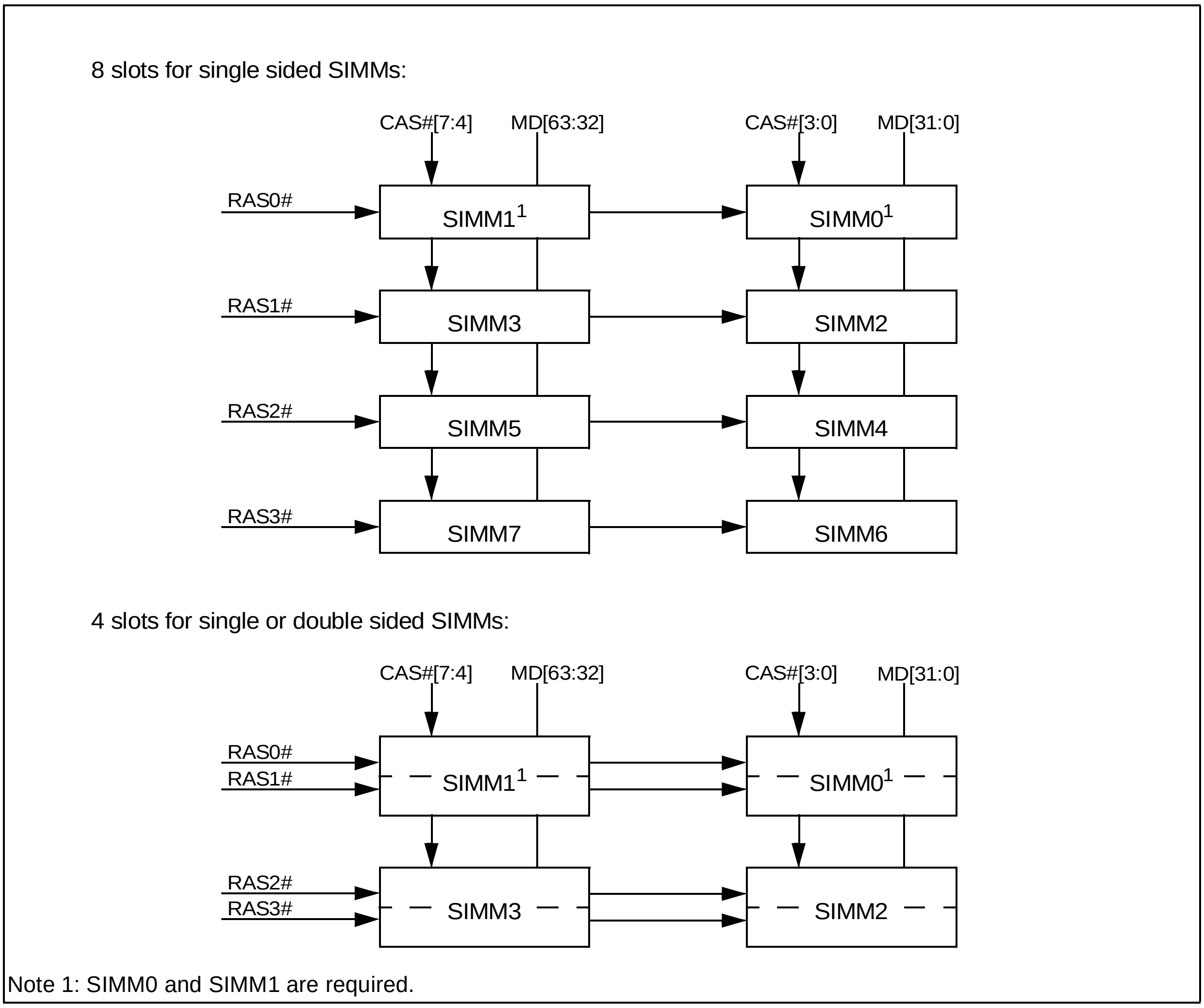


Figure 7-1. DRAM Organisation

The STPC DRAM interface consists of a host clock domain DRAM controller and a graphics clock domain DRAM controller. The arbitration between these two different clock domain controllers are described in Section 7.9. , DRAM arbitration. The host clock domain DRAM controller processes host initiated read and write cycles as well as DRAM refresh cycles. Graphics, CRTC and Video Scaler read and write cycles are processed by the graphics domain DRAM controller (see section 5.7).

7.8.1. FAST PAGE MODE (FPM) DRAM

This is the default DRAM type. Fast page mode allows accesses to the same row address to be executed without a RAS cycle. The column address is latched at the falling edge of CAS. Read data is valid toward the end of the CAS low pulse, then the data bus goes to high impedance after CAS goes high. The output enable is not necessary so it is tied low (active) on the standard 72 pin SIMMs.

7.8.2. EDO DRAM

The DRAM Controller supports Extended Data Out DRAM as well as Fast Page Mode DRAM. EDO DRAM keeps driving read data after CAS# goes high. This allows the data valid time for setup and hold to be overlapped with CAS# precharge.

The output enable is tied low (active) on the 72 pin EDO SIMMs. The DRAM internal output enable turns on when WE# and RAS# are low (active) at the falling edge of CAS#. The rising edge of CAS# has no effect on the output enable. A rising edge on RAS# or WE# is required to turn off the output enable.

The Page Access Mode pin 66 on the standard 72 pin DRAM SIMM indicates EDO DRAM when shorted to VSS or Fast Page Mode if left open. The SIMM type can be detected on initialization through the memory data pins via a resistor network.

7.8.3. Host Address to MA bus Mapping

Graphics memory resides at the beginning of Bank 0. Host memory begins at the top of graphics memory and extends to the top of populated DRAM.

The bank attributes can be retrieved from a lookup table to select the final DRAM row and column address mappings. (Table 7-69)

256KxN and 512KxN DRAMs have a 9 bit column address so A11 must be included in the page hit comparison. All other DRAM types have a fixed effective page size of 10 bits in this design.

Some 4MxN DRAMs have 2K refresh while others have 4K refresh. This corresponds to 11 row/11 column and 12row/10column address bits respectively. The Host Address most significant bit is mapped to both row and column MSB to make the memory controller insensitive to the difference between 2K and 4K refresh DRAMs.
The 2 Host Address least significant bits must correspond to the memory address least significant bits for burst memory.

7.8.4. DRAM MODULE PRESENCE AND TYPE DETECT

JEDEC standard second generation 72 pin DRAM SIMM Modules have the following presence detect pins:

pin 66	Page Mode Detected
GND	EDO
N/C	Fast Page

Table 7-67. DRAM Page Mode Detect

PD2 pin 69	PD3 pin 70	Speed Detected
		t_{RAC}
GND	GND	50ns
N/C	N/C	60ns
GND	N/C	70ns
N/C	GND	80ns

Table 7-68. DRAM Speed Detect

DRAM row address													
Bank Height	**Bank Width**	ROW address											
		MA11	**MA10**	**MA9**	**MA8**	**MA7**	**MA6**	**MA5**	**MA4**	**MA3**	**MA2**	**MA1**	**MA0**
256K	32-bit	X	X	X	A12	A11	A19	A18	A17	A16	A15	A14	A13
512K	32-bit	X	X	A20	A12	A11	A19	A18	A17	A16	A15	A14	A13
1M	32-bit	X	X	A12	A21	A20	A19	A18	A17	A16	A15	A14	A13
2M	32-bit	X	A22	A12	A21	A20	A19	A18	A17	A16	A15	A14	A13
4M (2k)	32-bit	X	A22	A12	A21	A20	A19	A18	A17	A16	A15	A14	A13
4M (4k)	32-bit	A23	A22	A12	A21	A20	A19	A18	A17	A16	A15	A14	A13
16M	32-bit	A24	A22	A12	A21	A20	A19	A18	A17	A16	A15	A14	A13
256K	64-bit	X	X	X	A12	A20	A19	A18	A17	A16	A15	A14	A13
1M	64-bit	X	X	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
2M	64-bit	X	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
4M (2K)	64-bit	X	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
4M (4K)	64-bit	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
16M	64-bit	A26	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
DRAM column address													
Bank Height	**Bank Width**	Column address											
		MA11	**MA10**	**MA9**	**MA8**	**MA7**	**MA6**	**MA5**	**MA4**	**MA3**	**MA2**	**MA1**	**MA0**
256K	32-bit	X	X	X	A10	A9	A8	A7	A6	A5	A4	A3	A2
512K	32-bit	X	X	X	A10	A9	A8	A7	A6	A5	A4	A3	A2
1M	32-bit	X	X	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
2M	32-bit	X	X	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
4M (2k)	32-bit	X	A23	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
4M (4k)	32-bit	X	X	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
16M	32-bit	A25	A23	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
256K	64-bit	X	X	X	A11	A10	A9	A8	A7	A6	A5	A4	A3
1M	64-bit	X	X	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
2M	64-bit	X	X	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
4M (2k)	64-bit	X	A24	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
4M (4k)	64-bit	X	X	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
16M	64-bit	A25	A24	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3

A = Corresponding remapped host address bit, MA = memory address,X = don’t care

DRAM row address

Table 7-69. Host Address to MA Bus Mapping

7.9. DRAM ARBITRATION

Following agents compete for the system DRAM memory:

CPU,

PCI masters,

ISA masters,

Graphics engine,

CRT controller,

Video Output,

Video Input Port,

Refresh controller.

A hierarchical arbitration scheme is used to optimize the DRAM bandwidth usage. The system arbiter arbitrates among CPU, PCI and ISA masters. Refer to system arbiter section of this specification for details of how this is done. The winner of this arbitration, the system master, competes with the remaining agents for DRAM. The DRAM arbiter employs a dynamic arbitration algorithm to optimize the DRAM utilization. The arbiter behavior changes depending on whether the scan is close to and during the display of the video window.

The following rules apply when the scan is not close to the video window.

Refresh request is the lowest priority and is serviced only if no other agent is actively requesting.

CRTC requests while current occupancy of the FIFO is above the low water mark are the next lowest priority requests and can be arbitrated out by GE, CRTC or video requests.

CRTC requests when the occupancy is below the low water mark (urgent requests) have the highest priority will win over all other agents.

Graphics engine requests lose to urgent CRTC and System master request. A System master request will terminate an ongoing Graphics service at the nearest CAS boundary while a CRTC request can terminate a on-going graphics service at the end of a sequence of read/write not exceeding 4 CAS cycles.

--The Video Output requests not close to the video window are prioritized just above the refresh.

When the scan is close to the video window and during the video window display, the arbiter behavior changes significantly. The goal of the arbiter here is to ensure that the CRTC and Video FIFO occupancy is above a programmable minimum number of Bytes. This is necessary because, some memory and screen configurations do not have sufficient bandwidth availability. For example, consider a memory system running at 40 MHz providing 40x8=320 MB/sec peak bandwidth with 1024x768x16-bitx75 Hz screen. At this resolution and refresh rate, the dot clock is 80MHz resulting in sustained drain rate of 80x2(16-bit pixel) x 2 (CRTC+ unscaled video) = 320 MB/sec. Since the drain rate is equal to the peak available bandwidth, it can not be sustained if all the pixels are to be fetched on demand. To overcome this, the arbiter ensures that a reservoir of CRTC and video pixels is available before the video window scan starts so that the difference of the fetch and drain rates can be made up for by dipping into this reservoir. This reservoir thus progressively shrinks as the video window is painted and approaches 0 Bytes by the end of the video window. To ensure that the reservoir is filled up, a programmable distance before the video window x position, the arbiter switches over to a different set of low water marks for determining the urgency of the CRTC and video requests. Once urgent, these requests win over other requesters thus ensuring that the reservoir is full. Further, to avoid thrashing between CRTC and Video requests, the arbiter employs a programmable burst length to arbitrate between the two. Once the CRTC service is started, it is not interrupted by video until the burst length number of cycles have occurred and vice-versa. Since the drain rates of video changes with the scaling factor, the CRTC and video have different burst length parameter. Once the video window repaints starts, the low water marks decrease linearly over the size of the window, to reflect the decreasing number of reservoir Bytes needed to make up for the difference in the fetch and drain rates. All other memory requesters are granted access, only if both CRTC and video FIFO occupancies are above their low water marks. The rules for granting the memory to the remaining agents are same as those listed above.

7.10 UPDATE HISTORY FOR DRAM CONTROLLER CHAPTER

Section	Change	Text
Table 7-69.	Replaced	DRAM Column address 16M 32-bit on Column address MA10 A24 replaced with A23

The following changes have been made to the DRAM Controller Chapter on 18/08/99.

Section	Change	Text
7.6.6.	Added	p-clock (PCI-Clock)

The following changes have been made to the Memory Chapter.

Section	Change	Text
7.2.	Replaced	" ~~Bank 0 SIMMs must be used in pairs, 64 bits wide, since the Graphics Controller does not support 32 bit banks. Banks 1,2 and 3 can be 32 or 64 bits wide~~." With "The internal Graphics Controller does not support 32 bit banks, therefore bank 0 SIMMs must be used in pairs, 64 bits wide. Banks 1,2 and 3 can be 32 or 64 bits wide."
7.2.	Added	" is programmed by software and"
7.4.	Replaced	"~~following map in the IO space~~" With "space in the Table 7-2 below"
7.4.	Added	Table title; "IO Map Space"
7.4.1.	Added	Table title "PCI configuration address space"
7.5.3.	Replaced	"~~Local Bus Disable~~" With "Reserved"
7.6.5.	Replaced	"~~VGA. It is presented here in the sequence of the Configuation Index. The function is~~" With "VGA"
7.6.6.	Added	"clock (Graphics"
7.6.6.	Added	"clock (Dot"
7.7.	Replaced	"~~Index~~" With "Configuration"
7.7.12.	Replaced	"~~Refresh disable bit. This register also contains the number of host clocks for the DRAM refresh interval~~." With "The register refresh disable bit also contains a number of host clocks for the DRAM refresh interval.
Table 7-69.	Replaced	"~~A24~~" With "X"
7.7.8.	Added	JEDEC standard second generation 72 pin DRAM SIMM Modules have the following presence detect pins
7.8.4.	Removed	"~~Table +~~ ~~S = short to Ground~~ ~~O = Open circuit~~ ~~The configuration codes are unique within the to 32MB density range. Software probing of memory locations is necessary to support a wider range of SIMM configurations.~~"

UPDATE HISTORY FOR DRAM CONTROLLER CHAPTER

8. PCI CONTROLLERS

8.1. INTRODUCTION

The PCI bus is the main data communication link to the STPC chip. Two PCI devices are present internally in the STPC, a “North Bridge” and a “South Bridge”. The STPC contains also a PCI arbiter which arbitrates between the 2 bridges and for up to three external PCI devices. Figure 8-1 below shows the layout of the PCI controllers within the STPC. Please refer to “PCI specification 2.1”, from PCI-SIG, to have more details on PCI bus standard.

The *North Bridge* translates appropriate host bus IO and Memory cycles to the PCI bus. It also supports generation of Configuration cycles on the PCI bus. The Configuration Address register, allows remapping host CPU’s IO cycles in the address range 0xCF8h-0xCFFh to configuration cycles on PCI bus.

The North Bridge, as a PCI bus agent (host bridge class) fully complies with PCI specification 2.1. The North Bridge also implements the PCI mandatory header registers in Type 0 PCI configuration space for easy porting of PCI-aware system BIOS. The North Bridge is assigned the Device Number 0xBh, which corresponds to IDSEL on AD11 signal. PCI configuration registers of the North Bridge are accessible by the Type 0 PCI configuration cycles generated at Device number 0xBh.

The South Bridge controller responds to PCI configuration read and write transactions. The South Bridge is assigned the Device Number 0xCh, which corresponds to IDSEL on AD12 signal. PCI configuration registers of the South Bridge are accessible by the Type 0 PCI configuration cycles generated by the North Bridge.

The South Bridge, as a PCI bus agent (expansion bridge class) fully complies with PCI specification 2.1. The South bridge implements two PCI functions:

Function 0, PCI to ISA bridge,

Function 1, IDE controller.

As per PCI specification, the South Bridge will respond to both function 0 and function 1 configuration cycles directed to configuration slot 12 (AD12).

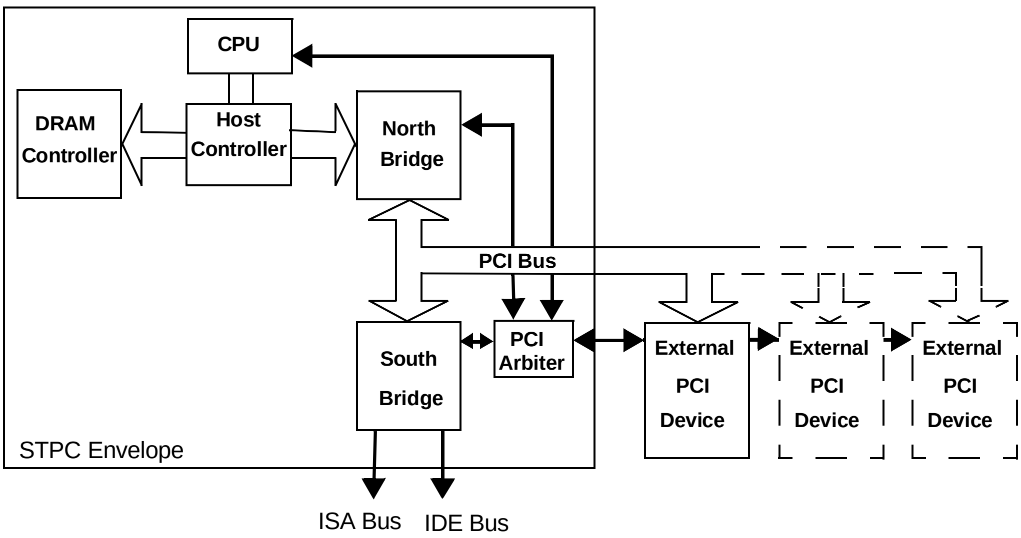


Figure 8-1. PCI Layout

8.1.1. PCI ADDRESS DECODE

The only positive decode done by the South Bridge is for the IDE controller PIO registers. The decode address ranges are dictated by the IDE configuration registers, see IDE section for details. ISA resources are accessed only via subtractive decode.

8.1.2. PCI ERROR HANDLING

Under control of South Bridge configuration registers, one or more of the following events can generate a 1 PCICLK long pulse on SERR#, which in turn can be made to generate an NMI to the CPU.

-ISA initiated transaction ending in target abort

8.1.3. PCI ARBITER

The PCI arbiter controls access to the PCI bus when several bus masters are present in the system. Whenever any other potential bus master needs to gain access to the bus it asserts its request. The arbiter then asserts a system hold condition, which eventually causes a hold signal to be asserted to the CPU. The CPU finishes what it is doing, tristates the internal bus and asserted a hold acknowledge. This eventually causes the assertion of a system hold acknowledge. Once the system hold acknowledge is asserted the arbiter asserts a grant to whichever requesting master is in the front of the line in the round-robin chain. When there are no requests pending or when the CPU is requesting the bus and it is in the front of the line, control of the bus is passed back to the CPU by the negation of the system hold condition.

8.2. METHOD FOR ACCESSING THE PCI CONFIGURATION REGISTERS

The PCI configuration registers are accessed, from the CPU, using two 32 bit registers mapped as IO at CF8h and CFCh.

Each read from and write to the PCI configuration registers must be done by:

- Writing the 32 bit address of the PCI config. register using type 0 format at IO CF8h.

- Reading or Writing 32 bit data at CFCh

All PCI configuration registers, inside the North and South bridges and all other external PCI devices, are seen from the CPU through those 2 x 32 bit registers.

An illustration of these registers is shown in Table 8-1& Table 8-2 below.

31	30 ----- 24	23 ------------ 16	15 -------------- 11	10 ------------------ 8	7 ------------------- 2	1 0
Enable	Reserved	Bus number	Device number	Function number	Register number	0

Table 8-1. Register CF8h

31 ---------------------------- 24	23 ---------------------------- 16	15 ----------------------------- 8	7 ------------------------------- 0
Byte 3	Byte 2	Byte 1	Byte 0

Table 8-2. Register CFCh

8.3. CONFIGURATION ADDRESS REGISTER

This is a 32-bit register accessible only via double-word IO read and write cycles.

Config_Address Access = 0xCF8h Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCI	Rsv							BN							
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DN					FN			RG						Rsv	
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bit 31	PCI	**PCI configuration register access enable.** When set to a ‘1’, host CPU IO cycles in address range CFCh-CFFh are converted to configuration cycles on the PCI bus. Otherwise if set to a ‘0’, IO cycles in this address range pass through as normal IO cycles on the PCI bus.
Bits 30-24	Rsv	**Reserved.** Must be written to ‘0’. Read back as ‘0’.
Bits 23-16	BN	**Bus Number.** This field selects a specific bus number in the system. Bus Number 0 is assigned to the PCI bus directly behind the North Bridge. This field is driven on bits 23-16 of the AD bus during the address phase.
Bits 15-11	DN	**Device Number.** This field selects a specific device on the bus. During a Type-0 configuration cycle, this field is decoded to assert the appropriate ID-SEL line. The North Bridge is assigned the Device Number 0xBh, which corresponds to IDSEL on AD11 signal. The South Bridge is assigned the Device Number 0xCh, which corresponds to IDSEL on AD12 signal.
Bits 10-8	FN	**Function Number.** During a PCI configuration cycle, this field is driven on bits 10-8 of the AD bus of the PCI during the address phase. Function 0: PCI to ISA bridge, Function 1, IDE controller.
Bits 7-2	RG	**Register Number.** During a PCI configuration cycle, this field is driven on bits 7-2 of the AD bus during the address phase.
Bits 1-0	Rsv	**Reserved**. Must be written to a ‘0’. Reads back as ‘0’.

8.4. CONFIGURATION DATA REGISTER

This is a 32-bit register accessible via 8-, 16- and 32-bit IO read and write cycles.

Config_Data Access = 0xCFCh Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

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

8.5. NORTH BRIDGE CONFIGURATION REGISTERS

The STPC North Bridge configuration registers are accessed using the values below :

Bus = 0h

Device = 0Bh (IDSEL internally connected to PCI address line 11)

Function = 0h (Host Bridge / PCI)

For example: Writting 80005800h at CF8h will access Vendor ID reg. index.

31 ----------	---------- 16	15 ----------	---------- 0	
Device ID : 0564hh		Vendor ID : 100Eh		00h
Status : 0280h		Command : 0007h		04h
Base class code: 00h	Sub class code: 00h	Program. Inter. Reg. : 00h	Revision : 00h	08h
	Header Type: 00h			0Ch
				...
Control Register : 00000000h				50h
Error Status Register : 00000000h				54h

Table 8-3. North Bridge Reset Values

8.5.1. NORTH BRIDGE VENDOR IDENTIFICATION REGISTER

This is a 16-bit read-only register implemented at configuration space offset 0h and 1h. It contains the Vendor Identifier assigned to the STPC.

NB_V_ID Access = 0xCF8h/0xCFCh Regoffset = 0x0h

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

Bits 15-0: These bits are hardwired to 100Eh.

Writes to this register have no effect.

8.5.2. NORTH BRIDGE DEVICE IDENTIFICATION REGISTER

This is a 16-bit read only register implemented at configuration space offset 2h and 3h. It contains the Device Identifier assigned to the North Bridge PCI Controller.

NB_D_ID Access = 0xCF8h/0xCFCh Regoffset = 0x2h

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

Bits 15-0 These bits are hardwired to 0564h

Writes to this register have no effect.

8.5.3. NORTH BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

NB_Com Access = 0xCF8h/0xCFCh Regoffset = 0x4h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv							S	AD	P	VGA	MW	ES	BS	ME	IO E
Default value after reset = 0007h															

Bit Number	Mnemonic	Description
Bits 15-9	**Rsv**	**Reserved**. These bits are hardwired to ‘0’. Writes have no effect on them.
Bit 8	**S**	**SERR# enable.** If this bit is set to a ‘1’, the North Bridge may assert SERR# upon detecting a target abort in response to an North Bridge initiated PCI transaction, upon being forced to end a non-configuration space transaction with a master abort, or if a parity error on the PCI bus is detected. If this bit is set to ‘0’, the North Bridge will not assert SERR#.
Bit 7	**AD**	**Address/Data stepping enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 6	**P**	**PERR# response.** Must always be set to ’0’.
Bit 5	**VGA**	**VGA Palette Snoop enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 4	**MW**	**Master Write and Invalidate Enable.** This bit is hardwired to a ‘0’. Writes to it have have no effect.
Bit 3	**ES**	**Enable Special cycles.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 2	**BS**	**Bus Master enabled.** This bit is hardwired to a ‘1’. Writes to it have no effect.
Bit 1	**ME**	**Memory Enable.** This bit is hardwired to a ‘1’. Writes to it have no effect.
Bit 0	**IO E**	**IO Enable.** This bit is hardwired to a ‘1’. Writes to it have no effect.

8.5.4. NORTH BRIDGE PCI STATUS REGISTER

This is the 16-bit PCI Status register.

NB_Stat Access = 0xCF8h/0xCFCh Regoffset = 0x6h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP	SS	SMA	RTA	STA	DT		DPED	FBBC	Rsv						
Default value after reset = 0280h															

Bit Number	Mnemonic	Description
Bit 15	**DP**	**Detected parity error.** This bit is set when a PCI parity error is detected. It may be cleared by software by writing a ‘1’ back to this bit.
Bit 14	**SS**	**Signaled SERR#.** This bit is set to a ‘1’ when SERR# is asserted by the North Bridge. Writing a ‘1’ to this bit will clear it.
Bit 13	**SMA**	**Signaled Master Abort.** This bit is set to a 1 when the North Bridge terminates a PCI transaction with a master abort. Writing a ‘1’ to this bit will clear it.
Bit 12	**RTA**	**Received Target Abort.** This bit is set to a ‘1’ when PCI transaction initiated by the North Bridge is terminated with a target abort. Writing a ‘1’ to this bit will clear it.
Bit 11	**STA**	**Signaled Target Abort.** This bit is hardwired to ‘0’.
Bits 10-9	**DT**	**DEVSEL Timing.** These bits are hardwired for medium timing to ‘01’. Writes have no effect.
Bit 8	**DPED**	**Data Parity Error Detected.** This bit is set to ‘1’ when a PCI data parity error is detected. Writing a ‘1’ will clear it.
Bit 7	**FBBC**	**Fast Back-to-Back Capable.** Hardwired to ‘1’. Indicates that the North Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return ‘1’, writes have no effect.
Bits 6-0	**Rsv**	**Reserved.** These bits are hardwired to ‘0’s.

8.5.5. NORTH BRIDGE PCI REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

NB_R_ID Access = 0xCF8h/0xCFCh Regoffset = 0x8h

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	**Rsv**	**Reserved.** These bits are hardwired to 00h.

8.5.6. NORTH BRIDGE DEVICE CLASS CODE REGISTER

This is a 24 bit read only register implemented at configuration space offset 9h, Ah, Bh.

NB_C_Code Access = 0xCF8h/0xCFCh Regoffset = 0x9h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCC								SCC							
Default value after reset = 00h								Default value after reset = 00h							

15	14	13	12	11	10	9	8
PIR							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 31-24	**BCC**	**Base Class Code.** These bits are hardwired to 00h.
Bits 23-16	**SCC**	**Sub Class Code.** These bits are hardwired to 00h.
Bits 15-8	**PIR**	**Programming Interface Register.** These bits are hardwired to 00h.

8.5.7. NORTH BRIDGE HEADER TYPE REGISTER

This is a 8 bit read only register hardwired to 00h

.

NB_Contol Access = 0xCF8h/0xCFCh Regoffset = 0xEh

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

8.5.8. NORTH BRIDGE CONTROL REGISTER

NB_Cont Access = 0xCF8h/0xCFCh Regoffset = 0x50h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv									PCI1	PCI2	PCI3	Rsv			
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv											P			SP	S
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-23	**Rsv**	**Reserved.** Hardwired to ‘0’.
Bit 22	**PCI1**	**PCI 2.0 Enable.** If this bit is set to ‘1’, North Bridge will be compatible with PCI 2.0 standard. If this bit is set to ‘0’, North Bridge is compatible with PCI 2.1 standard.
Bit 21	**PCI2**	**PCI to Host Read Prefetch Enable.** If this bit is set to ‘1’, all QWORD aligned burst reads from a PCI master addressed to the North Bridge system memory will use prefetch. If set to ‘0’, memory read cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst read attempts will be disconnected on the PCI bus.
Bit 20	**PCI3**	**PCI to Host Write Posting Enable.** If this bit is set to ‘1’, all burst writes from a PCI master addressed to the North Bridge system memory will be posted. If it is set to ‘0’, all memory write cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst write attempts will be disconnected on the PCI bus.
Bits 19-5	**Rsv**	**Reserved.** Hardwired to ‘0’.
Bit 4	**P**	**PERR_ on read data parity error enable.**
Bit 3	**P**	**PERR_ on write data parity error enable.**
Bit 2	**P**	**PERR_ on address parity error enable.**
Bit 1	**SP**	**SERR_ on PERR_ enable.**
Bit 0	**S**	**SERR_ on received target abort.**

8.5.9. NORTH BRIDGE PCI ERROR STATUS REGISTER

NB_E_Stat Access = 0xCF8h/0xCFCh Regoffset = 0x54h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv											RDP	WDP	AP	PES	RTAE
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-5	**Rsv**	**Reserved.** Hardwired to ‘0’.
Bit 4	**RDP**	**Read Data Parity Error Status.** This bit is set when a PCI read data parity error is detected. Writing a ‘1’ will clear it.
Bit 3	**WDP**	**Write Data Parity Error Status.** This bit is set when a PCI write data parity error is detected. Writing a ‘1’ will clear it.
Bit 2	**AP**	**Address Parity Error Status.** This bit is set when a PCI address parity error is detected. Writing a ‘1’ will clear it.
Bit 1	**PES**	**Parity Error Status.** System errors as a result of a parity error status. This bit is set to ‘1’ when SERR# was asserted as a result of parity error. Writing a ‘1’ will clear it.
Bit 0	**RTAE**	**Received Target Abort Error.** System errors as a result of a received target abort. This bit is set to ‘1’ when SERR# was asserted as a result of recieving a target abort. Writing a ‘1’ will clear it.

8.6. THE SOUTH BRIDGE

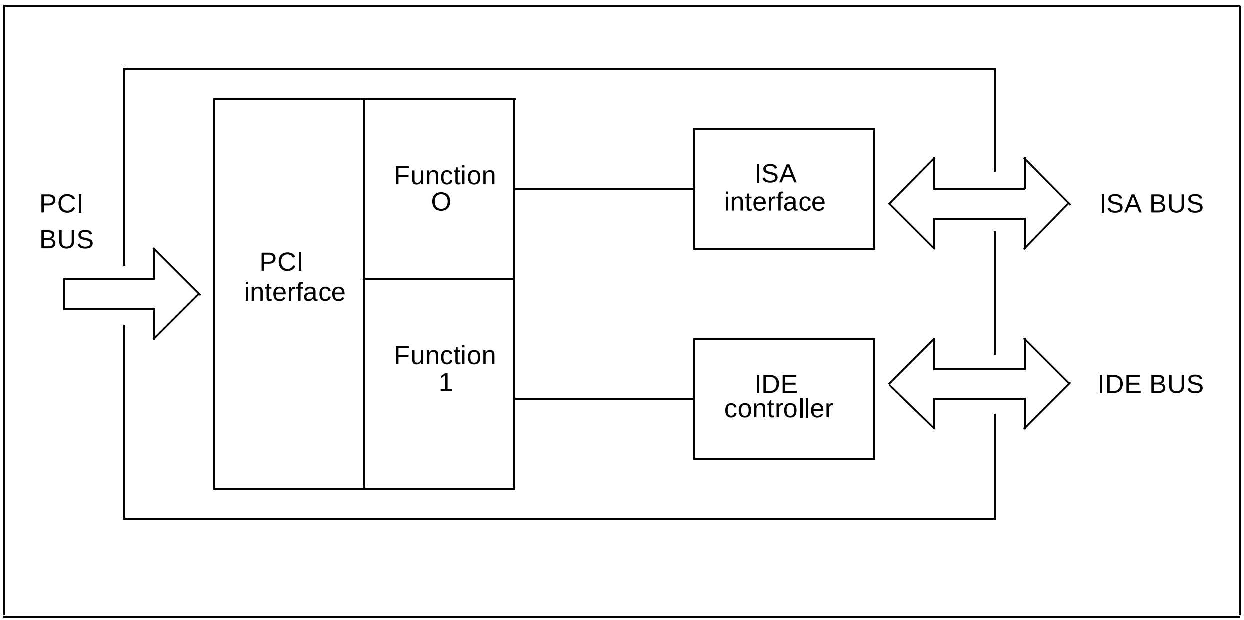


Figure 8-2. South bridge layout

The STPC South Bridge configuration registers are accessed using the values below :

Figure 8-2 Illustrates the South Bridge layout and the functions associated are listed below:

Bus = 0

Device = Ch (IDSEL internally connected to PCI address line 12)

Function = 0 (ISA bridge)

- Responds to IO / memory / config

- Translates Master ISA to PCI

- Translates PCI to Slave ISA

Function = 1 (IDE controller)

- Responds to IO / config

For example: Writting 80006110h at CF8h will access Function 1 (IDE) Command reg. index.

8.7. SOUTH BRIDGE PCI FUNCTION 0 CONFIGURATION REGISTERS

31	16	15	0	
Device ID: 55CCh		Vendor ID: 100Eh		00h
Status: 0280h		Command: 000Fh		04h
Base class code: 06h	Sub class code: 01h	Program. Inter. Reg. : 00h	Revision ID: 00h	08h
	Header: 80h			0Ch
				...
				...
			Miscellaneous reg : 00h	40h

Table 8-4. Function 0 (ISA bridge) Configuration Space Register Reset Values

This section describes Function 0 (F#0) configuration registers, including the PCI to ISA bridge control. The registers and reset values are illustrated in Table 8-4.

8.7.1. SOUTH BRIDGE VENDOR IDENTIFICATION REGISTER

This is a 16-bit read-only register implemented at configuration space offset 0h and 1h. It contains the Vendor Identifier assigned to STPC.

SB_V_ID0 Access = 0xCF8h/0xCFCh Regoffset = 0x0h

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

Bits 15-0 These bits are hardwired to 100Eh.

Writes to this register have no effect.

8.7.2. SOUTH BRIDGE DEVICE IDENTIFICATION REGISTER

This is a 16-bit read only register implemented at configuration space offset 02h and 03h. It contains the Device Identifier assigned to the South Bridge.

SB_D_ID0 Access = 0xCF8h/0xCFCh Regoffset = 0x2h

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

Bits 15-0 These bits are hardwired to 55CCh

Writes to this register have no effect.

8.7.3. SOUTH BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

SB_Com_0							Access = 0xCF8h/0xCFCh								Regoffset = 0x4h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv							S	AD	P	VGA	MW	ESC	BM	ME	IO E
Default value after reset = 000Fh															

Bit Number	Mnemonic	Description
Bits 15-9	**Rsv**	**Reserved**. These bits are hardwired to ‘0’s. Writes have no effect on them.
Bit 8	**S**	**SERR# enable.** If this bit is set to a ‘1’, the South Bridge may assert SERR# upon detecting a target abort in response to a South Bridge initiated PCI transaction on behalf of an ISA master. If this bit is set to ‘0’, the South Bridge will not assert SERR#.
Bit 7	**AD**	**Address/Data stepping enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 6	**P**	**PERR# response.** Setting this bit to ‘1’ enables parity error detection.
Bit 5	**VGA**	**VGA Palette Snoop enable.** This bits is hardwired to a ‘0’. Writes to it have no effect.
Bit 4	**MW**	**Master Write and Invalidate Enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 3	**ESC**	**Enable Special Cycles.** This bit is hardwired to a ‘1’. The South Bridge writes to it have no effect. The South Bridge responds to halt and shutdown cycles.
Bit 2	**BM**	**Bus Master enabled.** This bit is hardwired to a ‘1’. Writes to it have no effect.
Bit 1	**ME**	**Mem Enable.** This bit is hardwired to a ‘1’. Writes to it have no effect.
Bit 0	**IO E**	**IO Enable.** This bit is hardwired to a ‘1’. Writes to it have no effect.

8.7.4. SOUTH BRIDGE PCI STATUS REGISTER

This is the 16-bit PCI Status register.

SB_Stat0 Access = 0xCF8h/0xCFCh Regoffset = 0x6h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv	SS	SMA	RTA	STA	DT		DPED	FBBC	Rsv						
Default value after reset = 0280h															

Bit Number	Mnemonic	Description
Bit 15	**Rsv**	**Reserved**. This bit is hardwired to '0'.
Bit 14	**SS**	**Signaled SERR#.** This bit is set to a '1' when SERR# is asserted by the South Bridge on behalf of an ISA master cycle. Writing a '1' to this bit will clear it. SERR# is asserted in response to a target abort during an ISA master cycle on PCI bus and if bit-8 of the F#0 PCI command register is set to a '1' to enable SERR# signaling.
Bit 13	**SMA**	**Signaled Master Abort.** This bit is hardwired to a '0'.
Bit 12	**RTA**	**Received Target Abort.** This bit is set to a '1' when the PCI transaction is initiated by the South Bridge on behalf of an ISA master is terminated with a target abort. Writing a '1' to this bit will clear it.
Bit 11	**STA**	**Signaled Target Abort.** This bit is set to a '1' when the South Bridge terminates a PCI transaction with a target abort. Writing a '1' to this bit will clear it. The South Bridge will generate target abort if a A1-0 of a PCI IO cycle does not match the Byte enables.
Bits 10-9	**DT**	**DEVSEL Timing.** These bits are hardwired for medium timing to '01'. Writes have no effect.
Bit 8	**DPED**	**Data Parity Error Detected.** This bit is hardwired to '0'.
Bit 7	**FBBC**	**Fast Back-to-Back Capable.** Hardwired to '1'. Indicates that the South Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return '1', writes have no effect.
Bits 6-0	**Rsv**	**Reserved.** These bits are hardwired to '0'.

8.7.5. SOUTH BRIDGE PCI REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

SB_R_ID0 Access = 0xCF8h/0xCFCh Regoffset = 0x8h

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Bits 7-0 These bits are hardwired to 00h.

8.7.6. SOUTH BRIDGE DEVICE CLASS CODE REGISTER

This is a 24 bit read only register implemented at configuration space offset 09h, 0Ah, 0Bh.

SB_C_Code0 Access = 0xCF8h/0xCFCh Regoffset = 0x9h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BCC								SCC							
Default value after reset = 06h								Default value after reset = 01h							

15	14	13	12	11	10	9	8
PIR							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 31-24	**BCC**	**Base Class Code.** These bits are hardwired to 06h (Bridge Device).
Bits 23-16	**SCC**	**Sub Class Code.** These bits are hardwired to 01h (ISA Bridge).
Bits 15-8	**PIR**	**Programming Interface Register.** These bits are hardwired to 00h.

8.7.7. SOUTH BRIDGE HEADER TYPE REGISTER

This register is hardwired to 80h indicating that the South Bridge is a multi-function PCI device.

SB_Head0 Access = 0xCF8h/0xCFCh Regoffset = 0xEh

7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0

8.7.8. SOUTH BRIDGE MISCELLANEOUS REGISTER

SB_Misc0 Access = 0xCF8h/0xCFCh Regoffset = 040h

7	6	5	4	3	2	1	0
Rsv							PCI
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-1	**Rsv**	**Reserved.** Hardwired to 00h.
Bit 0	**PCI**	**PCI 2.0 Enable.** If this bit is set to ‘1’, South Bridge will be compatible with PCI 2.0 standard. If this bit is set to ‘0’, South Bridge is compatible with PCI 2.1 standard.

8.8. SOUTH BRIDGE PCI FUNCTION 1 CONFIGURATION REGISTERS

This section describes the Function 1 (F#1) configuration registers. The registers and reset values are illustrated in Table 8-5.

31	16	15	0	
Device: 55CCh		Vendor ID: 100Eh		00h
Status: 0280h		Command: 0000h		04h
Base class code: 01h	Sub class code: 01h	Program. Inter. Reg. : 8Ah	Revision ID: 00h	08h
	Header: 80h	Reserved: 00h		0Ch
IO Base address 0 register: 00000001h				10h
IO Base address 1 register: 00000001h				14h
IO Base address 2 register: 00000001h				18h
IO Base address 3 register: 00000001h				1Ch
Reserved				20h
				... ...
Primary IDE Timing register: 97609760h				40h
Secondary IDE Timing register: 97609760h				44h
			Miscellaneous reg : 00h	48h

Table 8-5. Function 1 (IDE Bridge) PCI Configuration Space Reister reset values

8.8.1. SOUTH BRIDGE VENDOR IDENTIFICATION REGISTER

This is a 16-bit read-only register implemented at configuration space offset 00h and 01h. It contains the Vendor Identifier assigned to STPC.

Bits 15-0 These bits are hardwired to 100Eh.

SB_V_ID1 Access = 0xCF8h/0xCFCh Regoffset = 0x0h

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

Writes to this register have no effect.

8.8.2. SOUTH BRIDGE DEVICE IDENTIFICATION REGISTER

This is a 16-bit read only register implemented at configuration space offset 02h and 03h. It contains the Device Identifier assigned to the South Bridge.

Bits 15-0 These bits are hardwired to 55CCh

SB_D_ID1 Access = 0xCF8h/0xCFCh Regoffset = 0x2h

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

Writes to this register have no effect.

8.8.3. SOUTH BRIDGE PCI COMMAND REGISTER

This is the 16-bit PCI command register.

SB_Com1 Access = 0xCF8h/0xCFCh Regoffset = 0x4h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv							SE	A	P	VGA	MWIE	ESC	Rsv	M E	IO E
Default value after reset = 0000h															

Bit Number	Mnemonic	Description
Bits 15-9	**Rsv**	**Reserved.** These bits are hardwired to ‘0’s. Writes have no effect on them.
Bit 8	**SE**	**SERR# Enable.** If this bit is set to a ‘1’, the South Bridge may assert SERR# upon detecting a master or target abort in response to a the South Bridge initiated PCI transaction on behalf of IDE master. If this bit is set to ‘0’, the South Bridge will not assert SERR#.
Bit 7	**A**	**Address/Data stepping enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 6	**P**	**PERR# response.** Setting this bit to ‘1’ enables parity error detection.
Bit 5	**VGA**	**VGA Palette Snoop enable.** This bits is hardwired to a ‘0’. Writes to it have no effect.
Bit 4	**MWIE**	**Master Write and Invalidate Enable.** This bit is hardwired to a ‘0’. Writes to it have have no effect.
Bit 3	**ESC**	**Enable Special cycles.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 2	**Rsv**	**Reserved.**
Bit 1	**M E**	**Mem Enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.
Bit 0	**IO E**	**IO Enable.** Setting this bit to a ‘1’ enables access to the IDE IO registers.

8.8.4. SOUTH BRIDGE PCI STATUS REGISTER

SB_Stat1 Access = 0xCF8h/0xCFCh Regoffset = 0x6h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv	SS	SMA	RTA	Rsv	DT		DPED	FBBC	Rsv						
Default value after reset = 0280h															

Bit Number	Mnemonic	Description
Bit 15	**Rsv**	**Reserved**. This bit is hardwired to ‘0’.
Bit 14	**SS**	**Signaled SERR#.** This bit is set to a ‘1’.
Bit 13	**SMA**	**Signaled Master Abort.** This bit is set to a ‘1’ when the West Bridge terminates a PCI transaction initiated on behalf of the IDE master with a master abort. The West Bridge master aborts an IDE master cycle if no target responds to this cycle.
Bit 12	**RTA**	**Received Target Abort.** This bit is set to a ‘1’ when a PCI transaction initiated by the West Bridge on behalf of the IDE master is terminated with a target abort. Writing a ‘1’ to this bit will clear it.
Bit 11	**Rsv**	**Reserved**. This bit is hardwired to ‘0’.
Bits 10-9	**DT**	**DEVSEL Timing.** These bits are hardwired for medium timing to ‘01’. Writes to these bits have no effect.
Bit 8	**DPED**	**Data Parity Error Detected.** This bit is hardwired to ‘0’.
Bit 7	**FBBC**	**Fast Back-to-Back Capable.** This bit is hardwired to ‘1’.
Bits 6-0	**Rsv**	**Reserved**. These bits are hardwired to ‘0’.

8.8.5. SOUTH BRIDGE REVISION ID REGISTER

This is the 8-bit read only PCI revision identification register.

SB_R_ID1 Access = 0xCF8h/0xCFCh Regoffset = 0x8h

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	**Rsv**	**Reserved.** These bits are hardwired to 00h in this stepping of the chip.

8.8.6. SOUTH BRIDGE PROGRAMMING INTERFACE REGISTER

Prog_Int Access = 0xCF8h/0xCFCh Regoffset = 0x09h

7	6	5	4	3	2	1	0
Rsv	Rsv						
Default value after reset = 8Ah							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** This bit is hardwired to ‘1’. Writes to have no effect on this bit.
Bits 6-4	**Rsv**	**Reserved.** These bits are hardwired to ‘0’. These bits are hardwired to ‘0’.
Bit 3		This bit is hardwired to ‘1’ indicating that the secondary channel is programmable to be either in legacy or native mode.
Bit 2		This bit selects the operating mode of the secondary channel. (see table below: Table 8-6)
Bit 1		This bit is hardwired to ‘1’ indicating that the primary channel is programmable to be either in legacy or native mode.
Bit 0		This bit selects the operating mode of the primary channel. (see table below: Table 8-7)

Bit 2	
0	Channel is in legacy mode. In legacy mode the secondary IDE channel occupies IO addresses 170h-177h and 376h.
1	Channel is in native mode. The address range occupied by the secondary IDE controller in native mode is specified by base address registers 2 and 3.

Table 8-6. Operating mode of the secondary channel

Bit 0	
0	Channel is in legacy mode. In legacy mode the Primary IDE channel occupies IO addresses 1F0h-1F7h and 3F6h.
1	Channel is in native mode. The address range occupied by the Primary IDE controller in native mode is specified by base address registers ‘0’ and ‘1’.

Table 8-7. Operating mode of the primary channel

8.8.7. SOUTH BRIDGE SUB-CLASS CODE REGISTER

This register is hardwired to 01h indicating that this is an IDE controller device.

Sub_Class Access = 0xCF8h/0xCFCh Regoffset = 0xAh

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

8.8.8. SOUTH BRIDGE BASE-CLASS CODE REGISTER

This register is hardwired to 01h indicating that Function 1 is a mass storage device.

Base_Class Access = 0xCF8h/0xCFCh Regoffset = 0xBh

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1

8.8.9. SOUTH BRIDGE LATENCY TIMER CONTROL REGISTER

Lat_T Access = 0xCF8h/0xCFCh Regoffset = 0xDh

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	**Rsv**	**Reserved.** These bits are hardwired to ‘0’.

8.8.10. SOUTH BRIDGE HEADER TYPE REGISTER

This register is hardwired to 80h indicating that the South Bridge is a PCI multi-function device.

Head_T Access = 0xCF8h/0xCFCh Regoffset = 0xEh

7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0

8.8.11. SOUTH BRIDGE IDE BASE ADDRESS 0 REGISTER

This 32-bit register contains the base IO address for accessing the primary IDE channel’s command registers. The base address is meaningful only when the Primary channel is programmed for native mode operation. If programmed for legacy mode operation, the primary channel’s command registers are decoded at 1F0h IO address.

Base0 Access = 0xCF8h/0xCFCh Regoffset = 0x10h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA														Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-3	**BA**	**Base Address.** This field specifies the 8-Byte IO address range where the primary channel command registers are located.
Bit 2		Hardwired to ‘0’ to indicate that this base address occupies 4-Bytes in IO space.
Bit 1	**Rsv**	**Reserved**. Hardwired to ‘0’.
Bit 0	**MSI**	**Memory Space Indicator.** This bit is hardwired to ‘1’ to indicate IO space.

8.8.12. SOUTH BRIDGE IDE BASE ADDRESS 1 REGISTER

This 32-bit register contains the base IO address for accessing the primary IDE channel's Control registers. The base address is meaningful only when the Primary channel is programmed for native mode operation. If programmed for legacy mode operation, the primary channel's control register are decoded at 3F6h.

Base1 Access = 0xCF8h/0xCFCh Regoffset = 0x14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA														Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-2	**BA**	**Base Address.** This field specifies the 4-Byte IO address range where the primary channel command registers are located.
Bit 1	**Rsv**	**Reserved**. Hardwired to '0'.
Bit 0	**MSI**	**Memory Space Indicator.** This bit is hardwired to '1' to indicate IO space.

8.8.13. SOUTH BRIDGE IDE BASE ADDRESS 2 REGISTER

This 32-bit register contains the base IO address for accessing the secondary IDE channel's command registers. The base address is meaningful only when the secondary channel is programmed for native mode operation. If programmed for legacy mode operation, the secondary channel's command registers are decoded at 170h IO address.

Base2 Access = 0xCF8h/0xCFCh Regoffset = 0x18h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA														Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-3	**BA**	**Base Address.** This field specifies the 8-Byte IO address range where the secondary channel command registers are located.
Bit 2		Hardwired to '0' to indicate that this base address occupies 4-Bytes in IO space.
Bit 1	**Rsv**	**Reserved**. Hardwired to '0'.
Bit 0	**MSI**	**Memory Space Indicator.** This bit is hardwired to '1' to indicate IO space.

8.8.14. SOUTH BRIDGE IDE BASE ADDRESS 3 REGISTER

This 32-bit register contains the base IO address for accessing the secondary IDE channel's Control registers. The base address is meaningful only when the secondary channel is programmed for native mode operation. If programmed for legacy mode operation, the secondary channel's control register is decoded at 376h.

Base3 Access = 0xCF8h/0xCFCh Regoffset = 0x1Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA														Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-2	**BA**	**Base Address.** This field specifies the 4-Byte IO address range where the secondary channel command registers are located.
Bit 1	**Rsv**	**Reserved**. Hardwired to '0'.
Bit 0	**MSI**	**Memory Space Indicator.** This bit is hardwired to '1' to indicate IO space.

8.8.15. SOUTH BRIDGE IDE BASE ADDRESS 4 REGISTER

This 32-bit register contains the base IO address for accessing the bus master control and status register.

Base4 Access = 0xCF8h/0xCFCh Regoffset = 0x20h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
Default value after reset = 00000001h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												HW		Rsv	MSI
Default value after reset = 00000001h															

Bit Number	Mnemonic	Description
Bits 31-4	**BA**	**Base Address.** This field specifies the 16-bytes IO address range where the Bus master control and status registers are located.
Bits 3-2	**HW**	Hardwired to ‘0’ to indicate that this base address occupies 16-bytes in IO space.
Bit 1	**Rsv**	**Reserved**. Hardwired to ’0’.
Bit 0	**MSI**	**Memory space indicator.** This bit is hardwired to ‘1’ to indicate IO space.

8.8.16. SOUTH BRIDGE PRIMARY IDE TIMING REGISTER

This 32-bit register contains all the timing information for the Read and Write signals when the primary port is active.

Bits 31-16 control the timing of the slave device on the primary port and bits 15-0 control the timing of the master device.

Prime_IDE_T Access = 0xCF8h/0xCFCh Regoffset = 0x40h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDSMS		IDRT		IDAT		IDE PIO R			IDE PIO AT			ISE	EWP	ERP	EPA
Default value after reset = 7F607F60h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSMS		DRT		DAT		PIO RT			PIO AT			OSE	EW	MERP	EP
Default value after reset = 7F607F60h															

Bit Number	Mnemonic	Description
Bits 31-30	**IDSMS**	**IDE DMA Speed Mode Select.** These bits, along with bits 29-26, determine the width of the read and write signals during DMA transfers. Refer to the Table 8-8 below to determine the number of clocks for active and recovery times for the read/write signals. The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.
Bits 29-28	**IDRT**	**IDE DMA Recovery Time.** These bits, along with bits 31-30, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to Table 8-9 below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.
Bits 27-26	**IDAT**	**IDE DMA Active Time.** These bits, along with bits 31-30, determine the duration of the active time of the read/write signals during DMA transfers. Refer to Table 8-10 below to determine the number of clocks for active times of the read/write signals. Default is 01h.
Bits 25-23	**IDE PIO R**	**IDE PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers. The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks. Refer to the Table 8-11 to determine the number of clocks for recovery times of the read/write signals.

Bit Number	Mnemonic	Description
Bits 22-20	IDE PIO AT	**IDE PIO Active Time.** These bits determine the duration of the active time of the read/write signals during PIO transfers. The transfers to non-data registers of the IDE device will always have an active time of 10 clocks. Refer to the Table 8-12 to determine the number of clocks for active times of the read/write signals: Note that the address setup time is implied. After initial startup latency, the address setup time is as follows in Table 8-13
Bit 19	ISE	**IOCHRDY Sampling Enable.** This bit when set to 1, enables IOCHRDY sampling of the IDE device, ie, the active read/write signals will be stretched (when IOCHRDY is low), to extend the cycle, if this bit is set. When set to 0, IOCHRDY is ignored and the read/write signals are deasserted after the specified number of PCI clocks.
Bit 18	EWP	**Enable Write Posting.** This bit when set to 1, enables posting of data into the FIFO during PIO data write transfers. Note that the non-data writes are never posted.
Bit 17	ERP	**Enable Read Prefetch.** This bit when set to 1 enables data to be prefetched into the data FIFO during PIO read commands. If set to 0, prefetch is completely disabled.
Bit 16	EPA	**Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when A0h is written into the command register - Register offset 07h for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices. Prefetch for the IDE controller is given in Table 8-14.
Bits 15-14	DSMS	**DMA Speed Mode Select.** These bits, along with bits 13-10, determine the width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals. The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.
Bits13-12	DRT	**DMA Recovery Time.** These bits, along with bits 15-14, determine the duration of the recovery(inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.
Bits 11-10	DAT	**DMA Active Time.** These bits, along with bits 15-14, determine the duration of the active time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.
Bits 9-7	PIO RT	**PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers. The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks. Refer to Table 8-18 below to determine the number of clocks for recovery times of the read/write signals.

Bit Number	Mnemonic	Description
Bits 6-4	PIO AT	**PIO Active Time.** These bits determine the duration of the active time of the read/write signals during PIO transfers. The transfers to non-data registers of the IDE device will always have an active time of 10 clocks. Refer to Table 8-19 below to determine the number of clocks for active times of the read/write signals. Note that the address setup time is implied. After initial startup latency, the address setup time is given in Table 8-20.
Bit 3	OSE	**IOCHRDY Sampling Enable.** This bit when set to 1, enables IOCHRDY sampling of the IDE device, ie, the active read/write signals will be stretched (when IOCHRDY is low), to extend the cycle, if this bit is set. When set to 0, IOCHRDY is ignored and the read/write signals are deasserted after the specified number of PCI clocks.
Bit 2	EW	**Enable Write Posting.** This bit when set to 1, enables posting of data into the FIFO during PIO data write transfers. Note that the non-data writes are never posted.
Bit 1	MERP	**Enable Read Prefetch.** This bit when set to 1 enables data to be prefetched into the data FIFO during PIO read commands. If set to 0, prefetch is completely disabled.
Bit 0	EP	**Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when A0h is written into the command register - Register offset 07h for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices. Prefetch for the IDE controller is given in Table 8-21.

Bit 31	Bit 30	DMA Speed Mode
0	0	Fast mode
0	1	Medium fast mode
1	0	Medium slow (Default)
1	1	Slow mode

Table 8-8. DMA Speed Mode Select

		Bit 29-28	Bit 29-28	Bit 29-28	Bit 29-28
Bit 31	Bit 30	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	10	11	12
1	1	14	15	16	20

Table 8-9. IDE DMA Recovery Time Settings

	27-26			
31-30	00	01	10	11
00	1	2	3	4
01	5	6	7	8
10	9	$\boxed{10}$	11	12
11	14	15	16	20

Table 8-10. IDE DMA Active Time Settings

Bit 25	Bit 24	Bit 23	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-11. Recovery R/W Signal Time

Bit 22	Bit 21	Bit 20	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-12. Active R/W Signal Time

Bit 22	Bit 21	Address setup time
0	0	1 clock
0	1	2 clocks
1	X	3 clocks

Table 8-13. Address Setup Time

Bit 17	Bit 16	Prefetch
0	X	Completely disabled
1	0	Enabled for non-ATAPI commands only
1	1	Enabled for all commands

Table 8-14. Prefetch Encoding

Bit 15	Bit 14	DMA Speed Mode
0	0	Fast mode
0	1	Medium fast mode
1	0	Medium slow
1	1	Slow mode (Default)

Table 8-15. DMA Speed Mode Select

		Bit 13-12	Bit 13-12	Bit 13-12	Bit 13-12
Bit 15	Bit 14	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	$\boxed{10}$	11	12
1	1	14	15	16	20

Table 8-16. DMA Recovery Time Settings

		Bit 11-10	Bit 11-10	Bit 11-10	Bit 11-10
Bit 15	Bit 14	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	$\boxed{10}$	11	12
1	1	14	15	16	20

Table 8-17. DMA Active Time Settings

Bit 9	Bit 8	Bit 7	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10 (Default)
1	1	1	12

Table 8-18. PIO R/W Signal Recovery Time

Bit 6	Bit 5	Bit 4	PCI Clocks
0	0	0	2
0	0	1	3
0	1	0	4
0	1	1	5
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-19. PIO R/W Signal Active Time

Bit 6	Bit 5	Address setup time
0	0	1 clock
0	1	2 clocks
1	X	3 clocks

Table 8-20. Address Setup Time Encoding

Bit 1	Bit 0	Prefetch
0	X	Completely disabled
1	0	Enabled for non-ATAPI commands only
1	1	Enabled for all commands

Table 8-21. Prefetch IDE Controller Encoding

8.8.17. SOUTH BRIDGE SECONDARY IDE TIMING REGISTER

This 32-bit register contains all the timing information for the Read and Write signals when the secondary port is active.

Bits 31-16 control the timing of the slave device on the secondary port and bits 15-0 control the timing of the master device on the secondary port.

Sec_IDE_T Access = 0xCF8h/0xCFCh Regoffset = 0x44h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SDSMS		SDRT		SDAT		SPIO RT			SPIO AT			SIO	SEWP	SERP	SEPA
Default value after reset = 7F607F60h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSMS		DRT		DAT		PIO RT			PIO AT			IO	EWP	ERP	EP
Default value after reset = 7F607F60h															

Bit Number	Mnemonic	Description
Bits 31-30	**SDSMS**	**DMA Speed Mode Select.** These bits, along with bits 29-26, determine width of the read and write signals during DMA transfers. Refer to Table 8-22 below to determine the number of clocks for active and recovery times for the read/write signals. The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.
Bits 29-28	**SDRT**	**DMA Recovery Time.** These bits, along with bits 31-30, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to Table 8-23 below to determine the number of clocks for recovery times of the read/write signals.Default is 01h
Bits 27-26	**SDAT**	**DMA Active Time.** These bits, along with bits 31-30, determine the duration of the active time of the read/write signals during DMA transfers. Refer to Table 8-24 below to determine the number of clocks for active times of the read/write signals. Default is 01h.
Bits 25-23	**SPIO RT**	**PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers. The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks. Refer to Table 8-25 below to determine the number of clocks for recovery times of the read/write signals.

Bit Number	Mnemonic	Description
Bits 22-20	SPIO AT	**PIO Active Time.** These bits determine the duration of the active time of the read/write signals during PIO transfers. The transfers to non-data registers of the IDE device will always have an active time of 10 clocks. Refer to the table below to determine the number of clocks for active times of the read/write signals:Table 8-26. Note that the address setup time is implied. After initial startup latency, the address setup time is given in Table 8-27.
Bit 19	SIO	**IOCHRDY Sampling Enable.** This bit when set to 1, enables IOCHRDY sampling of the IDE device, ie, the active read/write signals will be stretched (when IOCHRDY is low), to extend the cycle, if this bit is set. When set to 0, IOCHRDY is ignored and the read/write signals are deasserted after the specified number of PCI clocks.
Bit 18	SEWP	**Enable Write Posting.** This bit when set to 1, enables posting of data into the FIFO during PIO data write transfers. Note that the non-data writes are ever posted.
Bit 17	SERP	**Enable Read Prefetch.** This bit when set to 1 enables data to be prefetched into the data FIFO during PIO read commands. If set to 0, prefetch is completely disabled.
Bit 16	SEPA	**Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when A0h is written into the command register - Register offset 07h for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices. Prefetch for the IDE controller is given in Table 8-28.
Bits 15-14	DSMS	**DMA Speed Mode Select.** These bits, along with bits 13-10, determine the width of the read and write signals during DMA transfers. Refer to Table 8-29 below to determine the number of clocks for active and recovery times for the read/write signals. The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.
Bits 13-12	DRT	**DMA Recovery Time.** These bits, along with bits 15-14, determine the duration of the recovery(inactive) time of the read/write signals during DMA transfers. Refer to Table 8-30 below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.
Bits 11-10	DAT	DMA Active Time. These bits, along with bits 15-14, determine the duration of the active time of the read/write signals during DMA transfers. Refer to Table 8-31 below to determine the number of clocks for active times of the read/write signals. Default is 01h.
Bits 9-7	PIO RT	**PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers. The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks. Refer to Table 8-32 below to determine the number of clocks for recovery times of the read/write signals:

Bit Number	Mnemonic	Description
Bits 6-4	PIO AT	**PIO Active Time.** These bits determine the duration of the active time of the read/write signals during PIO transfers. The transfers to non-data registers of the IDE device will always have a active time of 10 clocks. Refer to Table 8-33 below to determine the number of clocks for active times of the read/write signals: Note that the address setup time is implied. After initial startup latency, the address setup time is given in Table 8-34.
Bit 3	IO	**IOCHRDY Sampling Enable.** This bit when set to 1, enables IOCHRDY sampling of the IDE device, ie, the active read/write signals will be stretched (when IOCHRDY is low), to extend the cycle, if this bit is set. When set to 0, IOCHRDY is ignored and the read/write signals are deasserted after the specified number of PCI clocks.
Bit 2	EWP	**Enable Write Posting.** This bit when set to 1, enables posting of data into the FIFO during PIO data write transfers. Note that the non-data writes are never posted.
Bit 1	ERP	**Enable Read Prefetch.** This bit when set to 1 enables data to be prefetched into the data FIFO during PIO read commands. If set to 0, prefetch is completely disabled.
Bit 0	EP	**Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when A0h is written into the command register - Register offset 07h for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices. Prefetch for the IDE controller is given in Table 8-35.

Bit 31	Bit 30	Speed Mode
0	0	Fast mode
0	1	Medium fast mode
1	0	Medium slow
1	1	Slow mode (Default)

Table 8-22. DMA Speed Mode Select.

		Bit 29-28	Bit 29-28	Bit 29-28	Bit 29-28
Bit 31	Bit 30	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	10	11	12
1	1	14	15	16	20

Table 8-23. DMA Recovery Time.

		Bit 27-26	Bit 27-26	Bit 27-26	Bit 27-26
Bit 31	**Bit 30**	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	10	11	12
1	1	14	15	16	20

Table 8-24. DMA Active Time.

Bit 25	Bit 24	Bit 23	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-25. PIO R/W Recovery Time

Bit 22	Bit 21	Bit 20	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-26. PIO Active Time Encoding

Bit 22	Bit 21	Address setup time
0	0	1 clock
0	1	2 clocks
1	X	3 clocks

Table 8-27. Address Setup Time

Bit 17	Bit 16	Prefetch
0	X	Completely disabled
1	0	Enabled for non-ATAPI commands only
1	1	Enabled for all commands

Table 8-28. IDE Controller Prefetch Encoding

Bit 15	Bit 14	Description
0	0	Fast mode
0	1	Medium fast mode
1	0	Medium slow
1	1	Slow mode (Default)

Table 8-29. DMA Speed Mode Select.

		Bit 13-12	Bit 13-12	Bit 13-12	Bit 13-12
Bit 15	Bit 14	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	10	11	12
1	1	14	15	16	20

Table 8-30. DMA Recovery Time.

		Bit 11-10	Bit 11-10	Bit 11-10	Bit 11-10
Bit 15	Bit 14	00	01	10	11
0	0	1	2	3	4
0	1	5	6	7	8
1	0	9	10	11	12
1	1	14	15	16	20

Table 8-31. DMA Active Time.

Bit 9	Bit 8	Bit 7	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-32. PIO R/W Signal Recovery Time

Bit 6	Bit 5	Bit 4	PCI Clocks
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	12

Table 8-33. PIO R/W Signal Active Time

Bit 3	Bit 2	Address setup time
0	0	1 clock
0	1	2 clocks
1	X	3 clocks

Table 8-34. Address Setup Time Encoding

Bit 1	Bit 0	Prefetch
0	X	Completely disabled
1	0	Enabled for non-ATAPI commands only
1	1	Enabled for all commands

Table 8-35. IDE Controller Prefetch Encoding

8.8.18. SOUTH BRIDGE MISCELLANEOUS REGISTER

This register contains miscellaneous informations.

SB_Misc1 Access = 0xCF8h/0xCFCh Regoffset = 0x48h

7	6	5	4	3	2	1	0
SR	Rsv	Rsv	Rsv	Rsv	Rsv	SID	PID
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**SR**	**Soft Reset.** When set to 1, the IDE controller is reset. It does not affect the timing control register. The FIFOs and the internal state machines are cleared.
Bits 6-2	**Rsv**	**Reserved.**
Bit 1	**SID**	**Secondary Interrupt Detect.** This bit is set when the secondary interrupt is active. It is cleared by writing a 1 to this bit in the register.
Bit 0	**PID**	**Primary Interrupt Detect.** This bit is set when the primary interrupt is active. It is cleared by writing a 1 to this bit in the register.

8.9 UPDATE HISTORY FOR PCI CONTROLLER CHAPTER

The following changes have been made to the PCI Controller Chapter 04/02/2000

Section	Change	Text
8.5.8.	Replaced	Bit 22 has been inverted to make 1 PCI 2.0 compatibility enabled

The following changes have been made to the PCI Controller Chapter 04/02/2000

Section	Change	Text
8.8.16.	Replaced	**Reinstated all Bus Master register definitions**
8.8.17.	Replaces	**Reinstated all Bus Master register definitions**

The following changes have been made to the PCI Controller Chapter 28/01/2000 /

Section	Change	Text
8.8.16.	Replaced	**PIO Active Time** bit 6-4 have been re defined
8.8.16.	Replaced	Default value from 97609760h to 7F607F60h
8.8.17.		Default value from 97609760h to 7F607F60h

The following changes have been made to the PCI Controller Chapter 13/01/2000

Section	Change	Text
8.5.3.	Replaced	~~“**PERR# response.** Setting this bit to ‘1’ enables parity error detection.”~~ with “**PERR# response.** Must always be set to ’0’.”

The following change has been made to the PCI Controller Chapter on 10/08/99

Bits 3-2 have been sewt to reserved in South Bridge Miscelaneous Register Index 48H.

The following changes have been made to the PCI Controller Chapter from Release 1.2 to Release 1.3.

East Bridge reference has been changed for North Bridge Reference & West Bridge reference has been changed for West Bridge reference.

The following changes have been made to the PCI Controller Chapter from Revision 1.0 to Release 1.2.

UPDATE HISTORY FOR PCI CONTROLLER CHAPTER

Section	Change	Text
9.1	Added	“Please refer to “PCI specification 2.1”, from PCI-SIG, to have more details on PCI bus standard.”
9.1	Added	“*The West Bridge* controller responds to PCI configuration read and write transactions as a PCI bus agent (expansion bridge class). The West bridge implements two PCI functions: Function 0, PCI to ISA bridge, Function 1, IDE controller.
9.2	Added	PCI Target Interface: PCI bus mastering devices can communicate with West/East Bridge resources via the PCI target interface. The target interface also contains the West/East Bridge PCI configuration register set, access to these registers is handled locally to the target interface.
9.3	Added	PCI Address Decode: The only positive decode done by the West Bridge is for the IDE controller PIO registers. The decode address ranges are dictated by the IDE configuration registers, see IDE section for details. ISA resources are accessed only via subtractive decode.
9.4	Added	PCI Error Handling: Under control of West Bridge configuration registers, one or more of the following events can generate a 1 PCICLK long pulse on SERR#, which in turn can be made to generate an NMI to the CPU. -ISA initiated transaction ending in target abort -IDE initiated transaction ending in target abort
9.4	Added	PCI Arbitor: The PCI arbitor controls access to the PCI bus when several bus masters are present in the system. Whenever any other potential bus master needs to gain access to the bus it asserts its request. The arbitor then asserts a system hold condition, which eventually causes a hold signal to be asserted to the CPU. The CPU finishes up what it is doing, tristates the internal bus and asserted a hold acknowledge. This eventually causes the assertion of a system hold acknowledge. Once the system hold acknowledge is asserted the arbitor asserts a grant to whichever requesting master is in the front of the line in the round-robin chain. When there are no requests pending or when the CPU is requesting the bus and it is in the front of the line, control of the bus is passed back to the CPU by the negation of the system hold condition.
9.4	Removed	*The West Bridge* controller responds to PCI configuration read and write transactions as a PCI bus agent (expansion bridge class). The West bridge implements two PCI functions: Function 0, PCI to ISA bridge and Function 1, Bus master IDE controller.
9.7	Changed	“~~All PCI config registers are seen through those 2 x 32 bit registers , inside the East and West bridges and all other external PCI devices~~.“ With “All PCI configuration registers, inside the East and West bridges and all other external PCI devices, are seen from the CPU through those 2 x 32 bit registers.”
9.8	Removed	“~~EAST BRIDGE PCI RELATED REGISTERS~~”
9.8.1	Changed	“~~CF8h (Config_Address)~~” With “CF8h”
9.8.2	Changed	“ ~~CFCh (Config_Data)~~” With “CFCh”
9.8.2	Changed	~~“This is a 32-bit register accessible via 8-, 16- and 32-bit read and writecycles.”~~ With “This is a 32-bit register accessible via 8-, 16- and 32-bit IO read and write-cycles.”

Section	Change	Text
9.9	Changed	"~~Bus = 0~~ ~~Device = BL (IDSEL internally connected to AD11)~~ ~~Function = 0~~" With "using the values below : Bus = 0h Device = Bh (IDSEL internally connected to PCI address line 11) Function = 0h (Host Bridge / PCI)"
9.9.1	Chaned	"~~ST Microelectronics~~" With "the STPC"
9.9.6	Changed	"~~Bits 15-8 (9h) These bits are hardwired to 00h~~." With "Bits 15-8 (09h) **Programming Interface Register.** These bits are hardwired to 00h."
9.9.7	Changed	"~~Bit 31-23 *Reserved.* Hardwired to '0'~~." With "Bit 31-23 *Reserved* Hardwired to '0'."
9.9.7	Changed	"~~Bit 21 **PCI to Host Read Prefetch Enable.** If this bit is set to '1', all QWORD aligned burst reads from a PCI master addressed to the East Bridge system memory will use prefetch. If set to '0', memory read cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst read attempts will be disconnected on the PCI bus.~~ " With "Bit 22 **PCI 2.0 Enable.** If this bit is set to '1', East Bridge memory will be compatible with PCI 2.0 standard. If this bit is set to '0', compatible with PCI 2.1 standard.
9.9.7	Changed	"~~Bit 20 **PCI to Host Write Posting Enable.** If this bit is set to '1', all burst writes from a PCI master addressed to the East Bridge system memory will be posted. If it is set to '0', all memory write cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst write attempts will be disconnected on the PCI bus.~~" With "Bit 21 **PCI to Host Read Prefetch Enable.** If this bit is set to '1', all QWORD aligned burst reads from a PCI master addressed to the East Bridge system memory will use prefetch. If set to '0', memory read cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst read attempts will be disconnected on the PCI bus."
9.10	Changed	"~~The STPC's West Bridge (W.B.) controller responds to PCI configuration read and write transactions. The West bridge, as a PCI bus agent (expansion bridge class), fully complies with the PCI specification. The West Bridge PCI interface itself is . PCI Device Number 0Ch, which corresponds to IDSEL on AD12 signal. PCI configuration registers are accessible by the Type 0 PCI configuration cycles. Fig 5.2 Illustrates the West Bridge layout~~." With "The STPC's West Bridge controller responds to PCI configuration read and write transactions. The West bridge, as a PCI bus agent (expansion bridge class), fully complies with the PCI specificationconfiguration registers are accessible by the Type 0 PCI configuration cycles. Figure 8-2 Illustrates the West Bridge layout.. The functions associated with the West Bridge are listed below:"

UPDATE HISTORY FOR PCI CONTROLLER CHAPTER

Section	Change	Text
9.12	Changed	"~~Device = Ch (IDSEL internlly connected to AD12)~~ ~~Function = 0(ISA bridge)~~ ~~Function = 1 (IDE)~~ ~~F0:- Responds to IO / memory / config~~ ~~Translates Master ISA to PCI~~ ~~Translates PCI to Slave ISA~~ ~~F1;- Responds to IO / config~~" With "Device = Ch (IDSEL internally connected to PCI address line 12) Function = 0(ISA bridge) Responds to IO / memory / config - Translates Master ISA to PCI - Translates PCI to Slave ISA Function = 1 (IDE controller) Responds to IO / config
9.10	Removed	~~West Bridge PCI Interface:~~ ~~There is error handling logic that takes care of detecting and logging bus anomolies.~~
9.10	Removed	~~PCI Target Interface:~~ ~~PCI bus mastering devices can communicate with West Bridge resources (IDE PIO registers or ISA based devices) via the PCI target interface. The target interface also contains the West Bridge PCI configuration register set, access to these registers is handled locally to the target interface.~~
9.10	Removed	~~PCI Address Decode:~~ ~~The only positive decode done by the West Bridge is for the IDE controller PIO registers. The decode address ranges are dictated by the IDE configuration registers, see IDE section for details. ISA resources are accessed only via subtractive decode.~~
9.10	Removed	~~PCI Error Handling:~~ ~~Under control of West Bridge configuration registers, one or more of the following events can generate a 1 PCICLK long pulse on SERR#, which in turn can be made to generate an NMI to the CPU.~~ ~~- ISA initiated transaction ending in target abort~~ ~~- IDE initiated transaction ending in target abort~~

Section	Change	Text
9.10	Removed	~~PCI Arbitor:~~ ~~The PCI arbitor controls access to the PCI bus when several bus masters are present in the system. Whenever any other potential bus master needs to gain access to the bus it asserts its request. The arbitor then asserts a system hold condition, which eventually causes a hold signal to be asserted to the CPU. The CPU finishes up what it is doing, tristates the internal bus and asserted a hold acknowledge. This eventually causes the assertion of a system hold acknowledge. Once the system hold acknowledge is asserted the arbitor asserts a grant to whichever requesting master is in the front of the line in the round-robin chain. When there are no requests pending or when the CPU is requesting the bus and it is in the front of the line, control of the bus is passed back to the CPU by the negation of the system hold condition.~~
9.11.3	Changed	"~~Bit 7-4 *Reserved.* These bits are hardwired to a '0'. Writes to it have no effect.~~" With "Bit 7 **Address/Data stepping enable.** This bit is hardwired to a '0'. Writes to it have no effect.
9.11.3	Addes	Bit 6 **PERR# response.** Setting this bit to '1' enables parity error detection. Bit 5 **VGA Palette Snoop enable.** This bits is hardwired to a '0'. Writes to it have no effect. Bit 4 **Master Write and Invalidate Enable.** This bit is hardwired to a '0'. Writes to it have have no effect.
9.11.3	Changed	"~~**PCI Config. F#0 Offset 09h-0Bh (WB_Class_Code0)**~~" With "PCI Config. F#0 Offset 09h-0Bh"
9.11.3	Changed	"~~Bits 15-8 (09h) *Reserved.* Hardwired to 00h~~." With "Bits 15-8 (09h) **Programming Interface Register.** These bits are hardwired to 00h.
9.11.8	Added	"West Bridge Miscellaneous register PCI Config. F#1 0 Offset 40h"
9.12	Replaced	"~~W.B. PCI FUNCTION 1~~ ~~CONFIGURATION REGISTERS~~ ~~This section describes the Function 1 (F#1) configuration registers~~." With "West Bridge PCI FUNCTION 1 CONFIGURATION REGISTERS This section describes the Function 1 (F#1) configuration registers. The registers and reset values are illistrated in Table 8-5."
9.12.1	Replaced	"~~W. B. Vendor Identification register~~ ~~register PCI Config. F#1 Offset 00h-01h (WB_Vend_ID1)~~ ~~This is a 16-bit read-only register implemented at configuration space offset 00h and 01h. It contains the Vendor Identifier assigned to ST Microelectronics.~~" With "West Bridge Vendor Identification register PCI Config. F#1 Offset 00h-01h This is a 16-bit read-only register implemented at configuration space offset 00h and 01h. It contains the Vendor Identifier assigned to STPC."
9.12.3	Replaced	"~~Bridge initiated PCI transaction on behalf of IDE master~~" With "Bridge"
9.12.3	Replaced	"~~Bit 7-3 *Reserved.* These bits are hardwired to a '0'. Writes to it have no effect.~~" With "Bit 7 **Address/Data stepping enable.** This bit is hardwired to a '0'. Writes to it have no effect.
9.12.3	Removed	Bit 2 Reserved
9.12.3	Added	Bit 6 **PERR# response.** Setting this bit to '1' enables parity error detection.

UPDATE HISTORY FOR PCI CONTROLLER CHAPTER

Section	Change	Text
9.12.3	Replaced	“~~Bit 1 **Mem Enable** This bit is hardwired to a ‘0’. Writes to it have no effect.~~” With “Bit 1 5 **VGA Palette Snoop enable.** This bit bits is hardwired to a ‘0’. Writes to it have no effect.”
9.12.3	Added	“Bit 4 **Master Write and Invalidate Enable.** This bit is hardwired to a ‘0’. Writes to it have have no effect.”
9.12.3	Added	“Bit 3 **Enable Special cycles.** This bit is hardwired to a ‘0’. Writes to it have no effect.”
9.12.3	Added	“Bit 1 **Mem Enable.** This bit is hard wired hardwired to a ‘0’. Writes to it have no effect.”
9.12.3	Added	“Bit 0 **IO Enable.** Setting this bit to a ‘1’ enables access to the IDE IO registers.”
9.12.4	Replaced	“~~Bit 14 **Signaled SERR#.** This bit is set to a ‘1’ when SERR# is asserted by the West Bridge on behalf of an IDE master cycle. Writing a ‘1’ to this bit will clear it. SERR# is asserted in response to master or target abort during an IDE master cycle on the PCI bus and if bit 8 of the F#1 PCI command register is set to a ‘1~~’.” With “Bit 14 **Signaled SERR#.** This bit is set to a ‘1’.”
9.12.4	Removed	~~“when the West Bridge terminates a PCI transaction initiated on behalf of the IDE master with a master abort. The West Bridge master aborts an IDE master cycle if no target responds to this cycle.”~~
9.12.4	Removed	~~“when a PCI transaction initiated by the West Bridge on behalf of the IDE master is terminated with a target abort. Writing a ‘1’ to this bit will clear it’.”~~
9.12.4	Replaced	~~“0 = Channel is in legacy mode. In legacy mode the secondary IDE channel occupies IO addresses 170h-177h and 376h. “~~ With “0 = Channel is in legacy mode. In legacy mode the secondary IDE channel occupies IO addresses 170h-177h and 376h.”
9.12.6	Removed	~~“indicating that the device supports master IDE”~~”
9.12.15	Replaced	~~“This 32-bit register contains the base IO address for accessing the bus master control and status register.~~ ~~Bits 31-4 **Base Address.** This field specifies the 16-bytes IO address range where the Bus master control and status registers are located.~~ ~~Bits 3-2 Hardwired to ‘0’ to indicate that this base address occupies 16-bytes in IO space.~~ ~~Bit 1 *Reserved.* Hardwired to ‘0’.~~ ~~Bit 0 **Memory space indicator.** This bit is hardwired to ‘1’ to indicate IO space.~~ ~~This register defaults to 00000001h at reset.~~” With “Bits 31-0 *Reserved.*

Section	Change	Text
9.12.16	Replaced	"~~Bits 31-30~~ ~~**IDE DMA Speed Mode Select.**~~ ~~These bits, along with bits 29-26, determine the width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.~~ ~~00 - Fast mode~~ ~~01 - Medium fast mode~~ ~~10 - Medium slow (Default)~~ ~~11 - Slow mode~~ ~~The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.~~" With "Bits 31-30 *Reserved.* Bits 29-28 *Reserved.* Bits 27-26 *Reserved."*
9.12.16	Replaced	"~~Bits 29-28~~ ~~**IDE DMA Recovery Time.**~~ ~~These bits, along with bits 31-30, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h~~." With "Bits 25-23 **IDE PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers.
9.12.16	Added	The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks.
9.12.16	Replaced	"Bits 27-26 **IDE DMA Active Time.** These bits, along with bits 31-30, determine the duration of the active time of the read/write during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h" With "Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h
9.12.16	Replaced	Table starting "27 - 26" with table starting "Bits 25 - 23"
9.12.16	Replaced	"~~Bits 22-23~~ ~~**IDE PIO Recovery Time.**~~ ~~These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers..~~ ~~The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks.~~ ~~Refer to the table below to determine the number of clocks for recovery times of the read/write signals:"~~ With "Bits 22- 20 **IDE PIO Active Time.** These bits determine the duration of the active time of the read/write signals during PIO transfers. The transfers to non-data registers of the IDE device will always have a active time of 10 clocks. Refer to the table below to determine the number of clocks for active times of the read/write signals:
9.12.16	Removed	The transfers to non-data registers of the IDE device will always have a active time of 10 clocks. Refer to the table below to determine the number of clocks for active times of the read/write signals:

Section	Change	Text
9.12.16	Replaced	"~~Bits 15-14 **DMA Speed Mode Select.** These bits, along with bits 13-10, determine the width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.~~ ~~00 - Fast mode~~ ~~01 - Medium fast mode~~ ~~10 - Medium slow~~ ~~11 - Slow mode (Default)~~ ~~The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.~~" With "Bits 15-14 *Reserved.* Bits 13-12 *Reserved.* Bits 11-10 *Reserved.*"
9.12.16	Replaced	"~~Bits 13-12 **DMA Recovery Time.** These bits, along with bits 15-14, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.~~" With "Bits 9-7 **PIO Recovery Time.** These bits, along with bits 15-14, determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers.
9.12.16	Added	The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks.
9.12.16	Removed	Bits 11-10 **DMA Active Time.** These bits, along with bits 15-14, determine the duration of the active time of the read/write signals during DMA transfers.
9.12.16	Replaced	"~~active~~" With "recovery"
9.12.16	Removed	Default is 01h
9.12.16	Removed	Bits 9-7 **PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers. The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks. Refer to the table below to determine the number of clocks for recovery times of the read/write signals:

Section	Change	Text
9.12.17	Replaced	~~“Bits 31-30 **DMA Speed Mode Select.** These bits, along with bits 29-26, determine width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.~~ ~~00 - Fast mode~~ ~~01 - Medium fast mode~~ ~~10 - Medium slow~~ ~~11 - Slow mode (Default)~~ ~~The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.”~~ With “Bits 31-30 *Reserved.*”
9.12.17	Replaced	~~“Bits 29-28 **DMA Recovery Time.** These bits, along with bits 31-30, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.”~~ With “Bits 29-28 *Reserved.*”
9.12.17	Replaced	~~“Bits 27-26 **DMA Active Time.** These bits, along with bits 31-30, determine the duration of the active time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.~~” With “Bits 27-26 *Reserved.*”
9.12.17	Replaced	~~“Bits 15-14 **DMA Speed Mode Select.** These bits, along with bits 13-10, determine the width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.~~ ~~00 - Fast mode~~ ~~01 - Medium fast mode~~ ~~10 - Medium slow~~ ~~11 - Slow mode (Default)~~ ~~The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.~~” With “Bits 15-14. *Reserved*

Section	Change	Text
9.12.17	Replaced	“~~Bits 13-12~~ **~~DMA Recovery Time.~~** ~~These bits, along with bits 15-14, determine the duration of the recovery(inactive) time of the read/write signals during DMA transfers.~~ ~~Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01~~h.” With “Bits 13-12. *Reserved*
9.12.18	Replaced	“~~Bits 11-10~~ **~~DMA Active Time.~~** ~~These bits, along with bits 15-14, determine the duration of the active time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.~~” With “Bits 11-10 *Reserved.*”

9. ISA INTERFACE

9.1. INTRODUCTION

The North Bridge acts as a bridge between the host CPU bus and the PCI bus. Reads and writes which are initiated by the CPU are subtractively decoded. Reads and writes that target North Bridge internal registers or main memory are routed to those targets, and all other reads and writes are sent to the PCI bus. The cycles for interrupt acknowledge, shutdown, stop grant and halt are also sent to the PCI bus.

The North Bridge also routes PCI reads and writes to main memory and its controler registers. The South Bridge acts as a bridge between the PCI bus and the ISA bus. ISA bus cycles may be initiated by PCI bus cycles, or by an ISA bus card. Additionally, refresh cycles are run periodically by the ISA controller.

The South Bridge will claim all PCI cycles which were initiated outside the South Bridge and not claimed by any other PCI slave. Reads and writes to PCI configuration registers, or to the IDE controller are routed appropriately by the South Bridge's PCI controller. All other PCI operations, including reads and writes to the South Bridge internal registers, are sent to the ISA controller. With the exception of some writes to the keyboard controller. Under certain conditions, a read or write cycle sent to the ISA bus controller creates one or more ISA bus cycles.

Some ISA pins are time shared between the ISA bus and the IDE bus, so the ISA controller must arbitrate for sole ownership of the ISA bus before starting a cycle. Because of the speed difference between the ISA bus and the PCI bus, and the requirement that PCI cycles be less than a certain number of clocks, PCI cycles which go to the ISA bus will require retries on the PCI bus.

The cycles for interrupt acknowledge, shutdown, stop grant and halt are also sent to the ISA bus controller. These cycles do not create ISA bus cycles, but they use the same state machines for timing and arbitration as reads and writes. Interrupt acknowledge is covered in this section, shutdown, stop grant and halt are covered in Section 9.1.1. (Special Cycles below).

ISA bus cycles which are initiated by an ISA bus card are either DMA cycles, in which case the address is supplied by the DMA controller, or ISA bus master cycles, in which case the address is supplied by the card itself.

9.1.1. SPECIAL CYCLES

Certain PCI special cycles are detected and forwarded to the ISA bus. Special cycles in which data bits 15-0 contain either 0000h or 0001h on the PCI bus, shutdown and halt respectively, are snooped and passed onto the ISA bus. Byte enables and address bits 0 and 4 are passed from the PCI to the ISA as well to support decode of the special cycle by the ISA.

9.2. ISAPCI / ISA CYCLES

9.2.1. PCI TO ISA READ AND WRITE

The PCI transfers data up to four Bytes at a time, with Byte enables for each Byte. The South Bridge's PCI controller transfers these four Bytes and four Byte enables to the ISA controller. The ISA controller in turn runs one to four ISA cycles. For eight bit targets, the enabled Bytes are read or written in order, least significant Byte (lowest address) first.

For sixteen bit targets, enabled Bytes are again read or written in order, but a sixteen bit transfer is used when an even Byte is enabled and the following odd Byte is also enabled.

Eight bit ISA operations are by default four and a half ISACLK cycles, starting on a falling edge of ISACLK and ending on a rising edge. Sixteen bit cycles are by default two and a half ISACLK cycles, also starting on a falling edge of ISACLK and ending on a rising clock. An additional clock cycle may be added by setting bit 5 in Index Register 50h. Cycles can also be extended by pulling IOCHRDY low.

9.2.2. PCI TO INTERNAL REGISTER READ AND WRITE

All South Bridge internal registers are 8 bits. If an IO read or write targets an internal register, the target is assumed to be 8 bits wide (that is IOCS16# is ignored). Timing for reads and writes to internal registers is the same as eight bit cycles on the ISA bus (see previous section).

If a write targets an internal register of the South Bridge, the data is written to the register and also to the ISA bus. If a read targets an internal register, the internal register is read, the South Bridge drives the ISA data bus with the contents of the register, and an ISA read cycle is done.

Registers that are called index registers in this document are indirectly addressed through a register at IO address 22h. There are two copies of this register, one on the North Bridge and one on the South Bridge.

Writes to IO address 22h go to both copies of the register. Reads from IO address 22h normally come from the North Bridge copy of the register, and do not generate a read cycle on the PCI bus. For test purposes, this behavior can be changed by setting bit zero of index register 21h. In this case, a read from IO address 22h reads the South Bridge copy of the register, using a PCI read cycle.

After selecting an index register by writing to IO address 22h, that index register is read from or written to at IO address 23h. Some index registers are implemented in the North Bridge alone, some in the South Bridge alone, and some are duplicated and implemented in both.

For index registers that are implemented in the North Bridge alone, writes to IO address 23h write to the register, and reads of IO address 23h read from the register, and no PCI cycles are generated.

For index registers that are implemented in the South Bridge alone, writes to IO address 23h write to the register, and reads of IO address 23h read from the register. In both cases, the data must go over the PCI bus.

For index registers that are implemented in both the North Bridge and the South Bridge, writes to IO address 23h write to both copies of the register, requiring a PCI write cycle. Reads to IO address 23h reads from the North Bridge copy of the register, and generate no PCI cycles. For test purposes, this behavior can be changed by setting bit zero of index register 21h. In this case, the South Bridge copy of the register is read, using a PCI read cycle.

9.2.3. INTERRUPT ACKNOWLEDGE CYCLE

When an interrupt is requested, the interrupt controller in the South Bridge asserts the CPU's interrupt input. When the CPU services the interrupt, it must first get the interrupt vector from the interrupt controller. The interrupt vector is used to find the interrupt service routine. Also, since each interrupt request input of the interrupt controller has its own interrupt vector, the vector tells where the interrupt request came from.

To get the interrupt vector, the CPU generates two interrupt acknowledge cycles. Both of these cycles read data from the interrupt controller. The data returned by the first is ignored, while the data for the second contains the interrupt vector in bits 0-7. The North Bridge handles both of the cycles identically, converting them to PCI interrupt acknowledge cycles.

Outside of the interrupt controller, the South Bridge handles both cycles identically. The ISA controller converts the PCI cycles into interrupt acknowledge cycles for the interrupt controllers. The INTA# input of the interrupt controller is asserted for four and a half ISA bus clocks, starting on a falling edge of that clock, and during this time data is transferred from the interrupt controller to the ISA controller. This can be extended to five and a half clocks by setting bit 5 in Index Register 50h.

9.2.4. ISA REFRESH CYCLE

The ISA bus controller also creates ISA bus refresh cycles. The frequency of refresh cycles is controlled by programming counter 1 of the interval timer (see Section 9.5.3.).

9.2.5. ISA TO PCI READ AND WRITE

ISA initiated cycles are converted to PCI cycles by the ISA controller. The South Bridge pulls IOCHRDY low to extend these cycles until the PCI cycle has completed.

9.2.6. ISA TO PCI BUFFERED READS

ISA reads of host memory can be buffered. This is disabled by default, and can be enabled by setting bit 6 in Index Register 50h. When this bit is set, ISA bus initiated reads of host memory addresses always get their data from a four Byte buffer in the ISA controller which is filled on demand. This can reduce the amount of traffic for a block memory read by up to a factor of four.

The buffer is filled or refilled, under the conditions listed below, after the start of an ISA initiated read of a host memory address has been detected by the South Bridge. The South Bridge generates a PCI read of four Bytes, with the low two bits of the address set to zero, and the rest of the address set to be the same as the address on the ISA bus address. The requested data will be driven by the South Bridge onto the ISA bus to finish the ISA read cycle.

9.2.7. ISA TO PCI POSTED WRITES

ISA writes to host memory can be posted. This is disabled by default, and can be enabled by setting bit 7 in Index Register 50h. When this bit is set, ISA bus initiated writes to host memory addresses go to a four Byte buffer in the ISA controller. No PCI write is generated until the buffer is written to host memory.

The buffer is written to host memory when the buffer gets full, or there is a host memory write to a location not in the buffer, or a host memory write would overwrite data already in the buffer, or there is a ISA cycle which is not a host memory write, or the current ISA master gives up ownership of the bus.

If writing the buffer to host memory is triggered by an ISA bus cycle, that cycle is held up by pulling IOCHRDY low until the buffer has been written to host memory.

Note that it is possible for the South Bridge to generate writes with discontiguous Byte enables if posted writes are enabled.

9.2.8. ISA TO REGISTER READ AND WRITE

ISA initiated cycles which target South Bridge internal registers will first be tried on the PCI bus. If they are not claimed by a PCI target, then the register will be read or written. Reads and writes to IPC registers will cause the South Bridge to pull IOCHRDY low for at least the number of cycles programmed into Index Register 01. Reads and writes to the South Bridge registers which are not IPC registers are normally disabled. These can be enabled by setting bit 7 of Index Register 51h. Writes to these registers require a longer than standard recovery time of two ISACLK periods.

9.3. XBUS READ AND WRITE

The XBUS is an 8 bit subset of the ISA bus that connects low speed devices on the mother board to the CPU. In particular, the Real Time Clock (RTC), the Keyboard Controller, and the BIOS ROM will usually be connected via the XBUS. For the STPC, the XBUS shares address, data and command lines with the ISA bus. No buffers or transceivers are required to connect the XBUS to the ISA bus. The timing for XBUS cycle is the same as that for eight bit ISA cycles, see above.

9.3.1. REAL TIME CLOCK READ AND WRITE

The Real Time Clock (RTC) is connected to the XBUS. However the RTC is not connected to the command lines of the XBUS. Instead, three input pins of the RTC (CS#, RW#, DS) are controlled directly by the STPC. The MOT pin of the RTC must be tied low. The registers in the RTC are accessed indirectly, by first writing the register number to IO port 70h, and then reading or writing the register at IO port 71h.

The RTC input CS# is connected to the logical OR of the outputs RMRTCCS# and ISAOE#. CS# is the

chip select for the RTC, and it will be driven low (active) on any IO read or write to port 70h or port 71h, and also will be driven low by reads or writes to ROM address space.
The RTC input RW# is connected to the logical OR of the RTCRW# and ISAOE# outputs. RW# is write pulse for the RTC, and it will be asserted (low) during any IO write to port 71h.

The RTC input DS is connected to the logical OR of the South Bridge outputs RTCDS and ISAOE#. DS is the read pulse for the RTC, and it will be asserted (low) during any IO read of port 71h.
The RTC interrupt output IRQ# is directly connected to the IRQ8B input. There is an internal inverter between the pin IRQ8B and the interrupt controller to maintain compatibility with the PC-AT without requiring additional external glue logic.

9.3.2. KEYBOARD CONTROLLER READ AND WRITE

The keyboard controller is connected to the XBUS. The chip select input of the keyboard controller is connected to the logical OR of the KBCS# and ISAOE# outputs.

Reads and writes to IO addresses 60h, 62h, 64h, 66h, 68h, 6Ah, 6Ch, and 6Eh are taken by the South Bridge to be reads and writes to the keyboard controller. Writes to the keyboard controller may be intercepted by South Bridge for keyboard controller emulation (see Section 9.5.5.). In this case, neither IOW# or KBCS# will be asserted. For writes to the keyboard controller that are not intercepted, both IOW# and KBCS# will be asserted (low) during the write. Similarly, for any reads from the keyboard controller, both IOR# and KBCS# will be asserted (low) during the read.

9.3.3. BIOS ROM READ AND WRITE

The BIOS ROM is connected to the XBUS. The chip select for the ROM is connected to the logical OR of the RMRTCCS# and ISAOE# outputs.

9.4. FAST CPU RESET AND FAST GATE A20

In the original PC/AT system, Gate A20 and CPU reset are controlled by writing to the keyboard controller. This is to force the address bit 20 to low or to reset the CPU, or to switch between the real mode and protected mode. Since the keyboard operation is very slow and writing to the keyboard controler will affect the system performance if the program needs to switch the modes frequently.

The STPC supports keyboard emulation to speed up the Gate A20 and CPU reset. The A20M# output pin to CPU is high when writing data D1h to I/O port 64h then writing data xxxxxx1x binary (bit 1 = ‘1’) to I/O port 60h. The A20M# is low when writing data D1h to I/O port 64h then writing data xxxxxx0x binary (bit 1 = ‘0’) to I/O port 60h. The Fast Reset, also known as warm reset, is generated by writing data FEh to I/O port 64h or by writing data FEh to I/O port 64h then writing data xxxxxxx0 binary (bit 0 = ‘0’) to I/O port 60h.

These keyboard write cycles are intercepted and will not be sent to keyboard controller by keeping KBCS# and IOW# high during the I/O operation. This function is software transparent and no BIOS modification is required.

9.5. ISA STANDARD REGISTERS

The ISA standard registers correspond to the registers in the peripheral components integrated in the STPC as well as the miscellaneous ports implemented on a ISA motherboard. These registers reside in IO space.

The functions controlled by the ISA registers include the DMA and interrupt control, BIOS and keyboard interface.

9.5.1. DMA 1 CONTROLLER REGISTERS

DMA 1 controls 8 bit DMA transfers.

There are 16 DMA 1 registers. They are as shown in Table 9-1.

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 000x 0000	xxxx xxxx	DMA 1 Channel 0 Base and Current Address	DMA1_CBA0
XXXX XX00 000x 0001	xxxx xxxx	DMA 1 Channel 0 Base and Current Count	DMA1_CBC0
XXXX XX00 000x 0010	xxxx xxxx	DMA 1 Channel 1 Base and Current Address	DMA1_CBA1
XXXX XX00 000x 0011	xxxx xxxx	DMA 1 Channel 1 Base and Current Count	DMA1_CBC1
XXXX XX00 000x 0100	xxxx xxxx	DMA 1 Channel 2 Base and Current Address	DMA1_CBA2
XXXX XX00 000x 0101	xxxx xxxx	DMA 1 Channel 2 Base and Current Count	DMA1_CBC2
XXXX XX00 000x 0110	xxxx xxxx	DMA 1 Channel 3 Base and Current Address	DMA1_CBA3
XXXX XX00 000x 0111	xxxx xxxx	DMA 1 Channel 3 Base and Current Count	DMA1_CBC3
XXXX XX00 000x 1000	xxxx 0000	DMA 1 Read Status/Write Command register	DMA1_RSWC
XXXX XX00 000x 1001	1111 xxxx	DMA 1 Request register	DMA1_RR
XXXX XX00 000x 1010	0000 0000	DMA 1 Read Command/Write Single Mask register	DMA1_RCWSM
XXXX XX00 000x 1011	0000 0000	DMA 1 Mode register	DMA1_Mode
XXXX XX00 000x 1100	1111 1111	DMA 1 Set/Clear Byte pointer flip-flop	DMA1_SCBPFF
XXXX XX00 000x 1101	0000 0000	DMA 1 Read Temp register/Master Clear	DMA1_RTMC
XXXX XX00 000x 1110	1111 1111	DMA 1 Clear Mask/Clear all request	DMA1_CMCAR
XXXX XX00 000x 1111	1111 1111	DMA 1 Read/Write all Mask register bits	DMA1_RWMB

Table 9-1 DMA1 registers

Note that the not all bits of the address are used.

Programming notes

Channel 0 corresponds to pin DRQ0B, channel 1 to DRQ1B, channel 2 to DRQ2B, and channel 3 corresponds to DRQ3B.

9.5.2. INTERRUPT CONTROLLER 1 REGISTERS

There are two Interrupt controller 1 registers. They are as shown in Table 9-2.

Interrupt controller 1 is the master interrupt controller.

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 001x xxx0	0000 0000	Interrupt Controller 1 register	IC_1
XXXX XX00 001x xxx1	1111 1111	Interrupt Controller 1 Mask register	IC_1MR

Table 9-2 Interrupt Controller 1 registers

Note that not all bits of the address are used.

Programming notes

Interrupt controller 1 input IR0 is connected pin IRQ0, IR1 to IRQ1, IR2 to interrupt out from interrupt controller 2, IR3 to IRQ3, IR4 to IRQ4, IR5 to IRQ5, IR6 to IRQ6, and IR7 to IRQ7.

9.5.3. INTERVAL TIMER REGISTERS

The Interval contains 3 independent counters. Counter 0 is used to generate timer interrupts, counter 1 is used to generate ISA bus refresh, and counter 2 is used to create the speaker tone.

There are 4 Interval Timer registers. They are as shown in Table 9-3.

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 010x xx00	xxxx xxxx	Interval Timer Register Counter 0 Count	IT_0
XXXX XX00 010x xx01	xxxx xxxx	Interval Timer Register Counter 1 Count	IT_1
XXXX XX00 010x xx10	xxxx xxxx	Interval Timer Register Counter 2 Count	IT_2
XXXX XX00 010x xx11	1111 1111	Command Mode register	IT_3

Table 9-3 Interval Timer Registers

Note that not all bits of the address are decoded.

Programming notes

All three counters are clocked by 1.193 MHz nominal frequency (OSC/12). Counter 0 and counter 1 gates are always on, counter 2 gate is controlled by writing to Port Bh (see Section 9.5.4.).

9.5.4. PORT Bh REGISTER

This is the ISA compatible 8-bit Port Bh register located at XXXX XX00 0110 xxx1 IO address (bits 15-0). It has the following meaning:

Port_B Access = 0061h Regoffset =

7	6	5	4	3	2	1	0
Rsv	IOCHK	T/C 2S	ISA RC	ISO IOCHK	Rsv	SE	T/C 2 G
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** This bit is read-only, the state should be ignored by the programmer.
Bit 6	**IOCHK**	**IOCHK# .** This bit is set to a '1' when IOCHK# signal of the ISA bus is asserted. Once set, this bit is cleared by setting bit 3 of this register to a '1'. Bit 3 should be reset to a '0' to enable recording the next IOCHK#. IOCHK# generates NMI to the host CPU if NMI is enabled. This bit is read only. On the IBM PC-AT, the IOCHK# signal is connected to the set input of a 74 type F/F and the bit 3 output is connected to the clear input. The clock is tied high and the output of the Flip/Flop latch (F/F) is fed into this bit without any synchronization.
Bit 5	**T/C 2S**	**T/C 2 State.** This bit reflects the output of Timer/Counter 2 without any synchronization. This bit is read only.
Bits 4-3	**ISA RC**	**ISA Refresh Check.** This bit toggles on every rising edge of the REFRESH# signal of the ISA bus. This bit is read only. On the IBM PC-AT, the REFRESH# signal is connected to the clock input of a positive edge triggered Toggle F/F (74ALS74 with Q# connected to D). The output of the F/F is connected to this bit without any synchronization.
Bit 3	**ISO IOCHK**	**ISA IOCHK# Enable.** This bit is connected to the asynchronous clear input of the F/F which records the IOCHK#. It must be set to a '1' to clear the F/F and then set to a '0' to enable further IOCHK# assertions. This bit is read/write and cleared to a '0' by ISA reset.
Bit 2	**Rsv**	**Reserved.** Although this bit is read/write, it iscleared to a '0' by an ISA reset.
Bit 1	**SE**	**Speaker Enable.** This bit is ANDed with the Interval Timer counter 2 OUT signal to drive the Speaker output signal. This bit is read/write and cleared to a '0' by ISA reset.
Bit 0	**T/C 2G**	**T/C 2 Gate.** This bit is connected to the gate input of the Interval Timer counter 2. This bit is read/write and cleared to a '0' by ISA reset.

9.5.5. PORT 60h AND 64h REGISTERS

These registers shadow the Input buffer port of the keyboard controller.

They are located at 0000 0000 0110 x0x0 binary and 0000 0000 0110 x1x0 binary IO addresses respectively. The STPC uses these ports to generate HA20M# and Fast CPU reset.

HA20M# is generated in the following manner. Whenever the STPC detects a write to Port 60h following a data write of D1h to Port 64h, bit 1 of the data Byte being written at Port 60h is driven on the HA20M# internal connection of the STPC CPU core. Neither write cycles are forwarded to the keyboard controller.

Fast host CPU only reset is generated by two methods:

(1) whenever the STPC detects a write to Port 64h with data FEh.

(2) Whenever the STPC detects a write to Port 60h following a D1h data write to Port 64h, bit 0 of the data Byte being written at Port 60h is ‘0’.

The CPU reset is at least 16 host clocks. The write cycle is not forwarded to the keyboard controller.

9.5.6. PORT 70h REGISTER

This 8-bit write-only register contains the NMI enable bit and is located at 0000 0000 0111 0xx1 IO address.

Port_70 Access = 0070h Regoffset =

7	6	5	4	3	2	1	0
NMI E	Rsv						
Default value after reset = 80h							

Bit Number	Mnemonic	Description
Bit 7	**NMI E**	**NMI Enable.** NMI is asserted on encountering IOCHK# on the ISA bus (Port_ B) or SERR# on the PCI bus if this bit is set to a ‘0’. Setting this bit to a ‘1’ disables NMI generation.
Bit 6-0	**Rsv**	**Reserved.** must be written to ‘0’s. Read back is undefined.

Programming notes

Writing to this address also sets the address register in the Real Time Clock (RTC, not part of the STPC, it is normally connected via the ISA interface).

9.5.7. INTERRUPT CONTROLLER 2 REGISTERS

Interrupt controller 2 is the slave interrupt controller.

Interrupt controller 2 occupies two register locations. They are as shown in Table 9-4.

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 101x xxx0	0000 0000	Interrupt Controller 2 register	IC_2R
XXXX XX00 101x xxx1	1111 1111	Interrupt Controller 2 Mask register	IC_2M

Table 9-4. Interrupt Controller 2 Registers

Note that not all address bits are decoded.

Programming notes

Interrupt controller 2 input IR1 is connected to IRQ9, IR2 to IRQ10, IR3 to IRQ11, IR4 to IRQ12, IR6 to IRQ14, IR7 to IRQ15. IR0 driven by IRQ8 inverted. IR5 is driven by an internally generated floating point error interrupt request

9.5.8. DMA CONTROLLER 2 REGISTERS

There are 16 DMA_2 registers. They are as shown in Table .

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 1100 000x	xxxx xxxx	DMA 2 Channel 0 Base and Current Address	DMA2_CBAO
XXXX XX00 1100 001x	xxxx xxxx	DMA 2 Channel 0 Base and Current Count	DMA2_CBCO
XXXX XX00 1100 010x	xxxx xxxx	DMA 2 Channel 1 Base and Current Address	DMA2_CBA1
XXXX XX00 1100 011x	xxxx xxxx	DMA 2 Channel 1 Base and Current	DMA2_CBC1
XXXX XX00 1100 100x	xxxx xxxx	DMA 2 Channel 2 Base and Current Address	DMA2_CBA2
XXXX XX00 1100 101x	xxxx xxxx	DMA 2 Channel 2 Base and Current	DMA2_CBC2
XXXX XX00 1100 110x	xxxx xxxx	DMA 2 Channel 3 Base and Current Address	DMA2_CBA3
XXXX XX00 1100 111x	xxxx xxxx	DMA 2 Channel 3 Base and Current Count	DMA2_CBC3
XXXX XX00 1101 000x	1111 xxxx	DMA 2 Read Status/Write Command register	DMA2_RSWC
XXXX XX00 1101 001x	0000 0000	DMA 2 Request register	DMA2_RR
XXXX XX00 1101 010x	0000 0000	DMA 2 Read Command/Write Single Mask register	DMA2_RCWSM
XXXX XX00 1101 011x	0000 0000	DMA 2 Mode register	DMA2_Mode
XXXX XX00 1101 100x	1111 1111	DMA 2 Set/Clear Byte pointer flip-flop	DMA2_SCBPFF
XXXX XX00 1101 101x	0000 0000	DMA 2 Read Temporary/Master Clear	DMA2_RTMC
XXXX XX00 1101 110x	1111 1111	DMA 2 Clear Mask/Clear all requests register	DMA2_CMCAR
XXXX XX00 1101 111x	1111 1111	DMA 2 Read/Write all Mask register bits	DMA2_RWMRB

Table 9-5. DMA Controller 2 Registers

Note that the not all bits of the address are used.

9.5.9. DMA PAGE REGISTERS

The DMA Page registers defines address bits [16-23] for DMA transfers controlled by DMA 1 or DMA 2. Bits [0-15] are generated by the DMA controller, bits [16-23] come from the appropriate page register, and bits 31-24 are all zeroes.

There are 16 DMA page registers. They are as shown in Table 9-6.

IO address bits 15-0	Reset Value	Register Name	Mnemonic
XXXX XX00 1000 0000	xxxx xxxx	DMA Page Register Port 80h (Reserved)	Port_80
XXXX XX00 1000 0001	xxxx xxxx	DMA Page Register Channel 2	DMA_PRC2
XXXX XX00 1000 0010	xxxx xxxx	DMA Page Register Channel 3	DMA_PRC3
XXXX XX00 1000 0011	xxxx xxxx	DMA Page Register Channel 1	DMA_PRC1
XXXX XX00 1000 0100	xxxx xxxx	DMA Page Register Port 84h (Reserved)	Port_84
XXXX XX00 1000 0101	xxxx xxxx	DMA Page Register Port 85h (Reserved)	Port_85
XXXX XX00 1000 0110	xxxx xxxx	DMA Page Register Port 86h (Reserved)	Port_86
XXXX XX00 1000 0111	xxxx xxxx	DMA Page Register Channel 0	DMA_PRC0
XXXX XX00 1000 1000	xxxx xxxx	DMA Page Register Port 87h (Reserved)	Port_87
XXXX XX00 1000 1001	xxxx xxxx	DMA Page Register Channel 6	DMA_PRC6
XXXX XX00 1000 1010	xxxx xxxx	DMA Page Register Channel 7	DMA_PRC7
XXXX XX00 1000 1011	xxxx xxxx	DMA Page Register Channel 5	DMA_PRC5
XXXX XX00 1000 1100	xxxx xxxx	DMA Page Register Port 8Bh (Reserved)	Port_8B
XXXX XX00 1000 1101	xxxx xxxx	DMA Page Register Port 8Ch (Reserved)	Port_8C
XXXX XX00 1000 1110	xxxx xxxx	DMA Page Register Port 8Dh (Reserved)	Port_8D
XXXX XX00 1000 1111	xxxx xxxx	DMA Page Register Port 8Eh (Reserved)	Port_8E

Table 9-6. DMA Page registers

9.6. ISA CONFIGURATION REGISTERS

9.6.1. MISCELLANEOUS CONTROL REGISTER 0

Misc_Cont0 Access = 0022h/0023h Regoffset = 050h

7	6	5	4	3	2	1	0
ISA WPE	ISA RBE	ISA WIC	ISA CFS	KRE	CPU D		
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**ISA WPE**	ISA Write Post Enable. If ‘1’, posted writes to host memory by ISA DMA or ISA bus master are enabled.
Bit 6	**ISA RBE**	**ISA Read Buffer Enable.** If ‘1’, buffered reads of host memory by ISA DMA or ISA bus master are enabled.
Bit 5	**ISA WIC**	**ISA Wait Insert Control.** This bit controls if extra wait state is inserted for slower ISA devices. (see table below: Table 9-7.).
Bit 4	**ISA CFS**	**ISA Clock Frequency Select.** This bit selects the ISA clock frequency (see table below: Table 9-8.).
Bit 3	**KRE**	**Keyboard Reset Enable.** This bit if set to a ‘1’, keyboard emulation fast gate A20 and fast reset are disabled. The source of warm reset indication is from the keyboard controller and the CPU core will use the gate A20 indication from keyboard controller for its internal A20M# input.
Bits 2-0	**CPU D**	**CPU Deturbo.** These three bits define the ratio CPU is held. (see table below: Table 9-9.).

Bit 5	ISA Wait Insert Control
0	no extra wait state for ISA cycle
1	one extra wait state for ISA cycle

Table 9-7. ISA Wait Insert Control

Bit 4	ISA Clock Frequency Select
0	ISA clock is 14.31818MHz / 2
1	ISA clock is PCICLK /4

Table 9-8. ISA Clock Frequency Select

Bit 2	Bit 1	Bit 0	CPU Deturbo
0	0	0	deturbo is disabled.
0	0	1	CPU is held 1/2 of the time.
0	1	0	CPU is held 2/3 of the time.
0	1	1	CPU is held 3/4 of the time.
1	0	0	CPU is held 4/5 of the time.
1	0	1	CPU is held 5/6 of the time.
1	1	0	CPU is held 6/7 of the time.
1	1	1	CPU is held 7/8 of the time.

Table 9-9. CPU Deturbo

9.6.2. MISCELLANEOUS CONTROL REGISTER 1

Misc_Cont1			Access = 0022h/0023h				Regoffset = 051h
7	6	5	4	3	2	1	0
IPC W	CLK 24	HCLK D	Rsv	ROM	S E S	S D S	S C S
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**IPC W**	**IPC Write control.** This bit controls the ISA master writes to the IPC register (see table below: Table 9-10.).
Bit 6	**CLK 24**	**CLK24 Enable.** This bit controls the output of CLK24. (see table below: Table 9-11.).
Bit 5	**HCLK D**	**HCLK Disable.** This bit controls the generation of HCLK. (see table below: Table 9-12.)
Bit 4	**Rsv**	**Reserved**.
Bit 3	**ROM**	**ROM Write Protect Enable.** This bit, if set to a ‘1’, disables write cycles to ROM BIOS on extended bus. If set to ‘0’, write to extended bus ROM BIOS is allowed. Note: This bit can not disable the write to shadowed BIOS in DRAM since after shadow is enabled, all writes to BIOS should be forwarded to extended bus.
Bit 2	**S E S**	**Segment E Share.** This bit controls if E0000h-EFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment. (see table below: Table 9-13.).
Bit 1	**S D S**	**Segment D Share.** This bit controls if D0000h-DFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment. (see table below: Table 9-14.).
Bit 0	**S C S**	**Segment C Share.** This bit controls if C0000h-CFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment. (see table below: Table 9-15.).

Bit 7	IPC Write control
0	ISA master writes to IPC register disabled
1	ISA master writes to IPC register enabled

Table 9-10. IPC Write control

Bit 6	CLK24 Enable
0	CLK24 generated normally
1	Clock synthesizer for CLK24 is disable (CLK24 will not toggle)

Table 9-11. CLK24 Enable

Bit 5	HCLK Disable
0	HCLK generated normally
1	Clock synthesizer for HCLK is disabled (HCLK will not toggle)

Table 9-12. HCLK Disable

Bit 2	Segment E Share
0	sharing disabled
1	sharing enabled

Table 9-13. Segment E Share

Bit 1	Segment D Share
0	sharing disabled
1	sharing enabled

Table 9-14. Segment D Share

Bit 0	Segment C Share
0	sharing disabled
1	sharing enabled

Table 9-15. Segment C Share

9.6.3. PIRQA ROUTING CONTROL REGISTER 0

This 8-bit register controls the routing of PCI Interrupt A# to one of the interrupt inputs of the 8259 as follows:

PAR_cont0 Access = 0022h/0023h Regoffset = 052h

7	6	5	4	3	2	1	0
RE A	Rsv			RC A			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**RE A**	**Routing Enable A#.** If set to a ‘1’, this bit enables the routing of PCI interrupt, otherwise the PCI interrupt A# is unconnected.
Bits 6-4	**Rsv**	**Reserved**. Writes have no affect. Reads return undefined value.
Bits 3-0	**RC A**	**Routing Control A#.** These bits route the PCI interrupt A# as follows in Table 9-16.

Bit 3	Bit 2	Bit 1	Bit 0	Interrupt A# Route	Note
0	0	0	1	Reserved	1
0	0	0	1	Reserved	1
0	0	1	0	Reserved	1
0	0	1	1	IRQ3	
0	1	0	0	IRQ4	
0	1	0	1	IRQ5	
0	1	1	0	IRQ6	
0	1	1	1	IRQ7	
1	0	0	0	Reserved.	1
1	0	0	1	IRQ9	
1	0	1	0	IRQ10	
1	0	1	1	IRQ11	
1	1	0	0	IRQ12	
1	1	0	1	Reserved.	1
1	0	1	0	IRQ14	
1	0	1	1	IRQ15	

Note 1: Interrupt can not be routed to this level.

Table 9-16. Interrupt A# Route

9.6.4. PIRQB ROUTING CONTROL REGISTER 0

This 8-bit register controls the routing of PCI Interrupt B# to one of the interrupt inputs of the 8259 as follows:

PBR_Cont0			Access = 0022h/0023h				Regoffset = 053h
7	6	5	4	3	2	1	0
RE B	Rsv			RC B			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**RE B**	**Routing Enable B#.** If set to a ‘1’, this bit enables the routing of PCI interrupt, otherwise the PCI interrupt B# is unconnected.
Bits 6-4	**Rsv**	**Reserved.** Writes have no affect. Reads return undefined value.
Bits 3-0	**RC B**	**Routing Control B#.** These bits route the PCI interrupt B# as follows in Table 9-17.

Bit 3	Bit 2	Bit 1	Bit 0	Interrupt B# Route	Note
0	0	0	1	Reserved	1
0	0	0	1	Reserved	1
0	0	1	0	Reserved	1
0	0	1	1	IRQ3	
0	1	0	0	IRQ4	
0	1	0	1	IRQ5	
0	1	1	0	IRQ6	
0	1	1	1	IRQ7	
1	0	0	0	Reserved.	1
1	0	0	1	IRQ9	
1	0	1	0	IRQ10	
1	0	1	1	IRQ11	
1	1	0	0	IRQ12	
1	1	0	1	Reserved.	1
1	0	1	0	IRQ14	
1	0	1	1	IRQ15	

Note 1: Interrupt can not be routed to this level

Table 9-17. Interrupt B# Route

9.6.5. PIRQC ROUTING CONTROL REGISTER 0

This 8-bit register controls the routing of PCI Interrupt C# to one of the interrupt inputs of the 8259 as follows:

PCR_Cont0 Access = 0022h/0023h Regoffset = 054h

7	6	5	4	3	2	1	0
RE C	Rsv			RC C			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**RE C**	**Routing Enable C#.** If set to a ‘1’, this bit enables the routing of PCI interrupt, otherwise the PCI interrupt C# is unconnected.
Bits 6-4	**Rsv**	**Reserved.** Writes have no affect. Reads return undefined value.
Bits 3-0	**RC C**	**Routing Control C#.** These bits route the PCI interrupt C# as follows Table 9-18.

Bit 3	Bit 2	Bit 1	Bit 0	Interrupt C# Route	Note
0	0	0	1	Reserved	1
0	0	0	1	Reserved	1
0	0	1	0	Reserved	1
0	0	1	1	IRQ3	
0	1	0	0	IRQ4	
0	1	0	1	IRQ5	
0	1	1	0	IRQ6	
0	1	1	1	IRQ7	
1	0	0	0	Reserved.	1
1	0	0	1	IRQ9	
1	0	1	0	IRQ10	
1	0	1	1	IRQ11	
1	1	0	0	IRQ12	
1	1	0	1	Reserved.	1
1	0	1	0	IRQ14	
1	0	1	1	IRQ15	

Note 1: Interrupt can not be routed to this level.

Table 9-18. Interrupt C# Route

9.6.6. PIRQD ROUTING CONTROL REGISTER 0

This 8-bit register controls the routing of PCI Interrupt D# to one of the interrupt inputs of the 8259 as follows:

PDR_Cont0 Access = 0022h/0023h Regoffset = 055h

7	6	5	4	3	2	1	0
RE D	Rsv			RC D			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**RE D**	**Routing Enable D#.** If set to a ‘1’, this bit enables the routing of PCI interrupt, otherwise the PCI interrupt D# is unconnected.
Bits 6-4	**Rsv**	**Reserved.** Writes have no affect. Reads return undefined value.
Bits 3-0	**RC D**	**Routing Control D#.** These bits route the PCI interrupt D# as follows Table 9-19.

Bit 3	Bit 2	Bit 1	Bit 0	Interrupt D# Route	Note
0	0	0	1	Reserved	1
0	0	0	1	Reserved	1
0	0	1	0	Reserved	1
0	0	1	1	IRQ3	
0	1	0	0	IRQ4	
0	1	0	1	IRQ5	
0	1	1	0	IRQ6	
0	1	1	1	IRQ7	
1	0	0	0	Reserved.	1
1	0	0	1	IRQ9	
1	0	1	0	IRQ10	
1	0	1	1	IRQ11	
1	1	0	0	IRQ12	
1	1	0	1	Reserved.	1
1	0	1	0	IRQ14	
1	0	1	1	IRQ15	

Note 1: Interrupt can not be routed to this level.

Table 9-19. Interrupt D# Route

9.6.7. INTERRUPT LEVEL CONTROL REGISTER 0

This 8-bit register allows interrupt requests to the lower 8259 to be either level or edge sensitive on an interrupt by interrupt basis overriding the global edge/level control bit of the 8259.

IRQ_Lev_C_0 Access = 0022h/0023h Regoffset = 056h

7	6	5	4	3	2	1	0
IRQ C					Rsv		
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-3	**IRQ C**	**IRQ Control IRQ[7-3].** If set to a ‘1’, the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible to ISA).
Bits 2-0	**Rsv**	**Reserved.** Writes have no affect. Reads return undefined value.

9.6.8. INTERRUPT LEVEL CONTROL REGISTER 1

This register allows interrupt requests to the upper 8259 to be either level or edge sensitive on an interrupt by interrupt basis overriding the global edge/level control bit of the 8259.

IRQ_Lev_C_1 Access = 0022h/0023h Regoffset = 057h

7	6	5	4	3	2	1	0
IRQ C		Rsv	IRQ C				IPC
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**IRQ C**	**IRQ Control IRQ[15-14].** If set to a ‘1’, the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible with ISA).
Bit 5	**Rsv**	**Reserved.** Writes have no affect and the reads return undefined value.
Bits 4-1	**IRQ C**	**IRQ Control IRQ[12-9].** If set to a ‘1’, the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible to ISA).
Bit 0	**IPC**	This bit controls the outone value from IPC. When low will force the outone to low, else the outone is driven by IPC.

9.6.9. IPC CONFIGURATION REGISTER

This 8-bit register controls the timing of the DMA controllers, and also the number of wait states for writes to registers in the IPC.

IPC_Conf Access = 0022h/0023h Regoffset = 001h

7	6	5	4	3	2	1	0
IPC WS		DMA		DMA		DMA M	DMA C
Default value after reset = C0h							

Bit Number	Mnemonic	Description
Bits 7-6	**IPC WS**	**IPC Wait States.** These bits specify the number of ISACLK wait states for read or write to IPC register1 (see table below: Table 9-20.).
Bits 5-4	**DMA**	**DMA 16-Bit Wait States.** These bits specify the number of wait states in 16-bit DMA cycles (see table below: Table 9-21.).
Bits 3-2	**DMA**	**DMA 8-Bit Wait States.** These bits specify the number of wait states in 8 bit DMA cycle (see table below: Table 9-22.).
Bit 1	**DMA M**	**DMA MEMR# Timing.** If this bit is set to '1' the DMA controllers will assert MEMR# at the the same time as IOW#. If set to '0' (default), MEMR# will be asserted one clock after IOW#.
Bit 0	**DMA C**	**DMA Clock Select.** If this bit is set to '0' (default), the DMA controller clock will be ISACLK divided by two, otherwise the DMA controller clock will be ISACLK.

Bit 7	Bit 6	IPC Wait States
0	0	1
0	1	2
1	0	3
1	1	4 (Default)

Table 9-20. IPC Wait States

Bit 5	Bit 4	DMA 16-bit Wait States
0	0	1 (Default)
0	1	2
1	0	3
1	1	4

Table 9-21. DMA 16-bit Wait States

Bit 3	Bit 2	DMA 8-bit Wait States
0	0	1 (Default)
0	1	2
1	0	3
1	1	4

Table 9-22. DMA 8-bit Wait States

Programming notes

To read or write to this register, write 01 to index register 22h, and then read or write from data register 23h.

9.6.10. VMI IRQ ROUTING CONTROL REGISTER

This 8-bit register controls the routing of VMI Interrupt to one of the interrupt inputs of the 8259 as follows:

VIR_Cont Access = 0022h/0023h Regoffset = 058h

7	6	5	4	3	2	1	0
VMI RE	Rsv			VMI RC			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**VMI RE**	**VMI Routing Enable.** If set to a '1', this bit enables the routing of VMI interrupt, otherwise the VMI interrupt is unconnected.
Bits 6-4	**Rsv**	**Reserved**. Writes have no affect. Reads return undefined value.
Bits 3-0	**VMI RC**	**VMI Routing Control.** These bits route the VMI interrupt as follows in Table 9-23.

Bit 3	Bit 2	Bit 1	Bit 0	VMI Interrupt Route	Note
0	0	0	1	Reserved	1
0	0	0	1	Reserved	1
0	0	1	0	Reserved	1
0	0	1	1	IRQ3	
0	1	0	0	IRQ4	
0	1	0	1	IRQ5	
0	1	1	0	IRQ6	
0	1	1	1	IRQ7	
1	0	0	0	Reserved.	1
1	0	0	1	IRQ9	
1	0	1	0	IRQ10	
1	0	1	1	IRQ11	
1	1	0	0	IRQ12	
1	1	0	1	Reserved.	1
1	0	1	0	IRQ14	
1	0	1	1	IRQ15	

Note 1: Interrupt can not be routed to this level.

Table 9-23. VMI Interrupt Route

9.6.11. ISA SYNCRONIZER BYPASS REGISTER

This 8-bit register controls whether or not the signals between the PCI logic and the ISA logic are passed through synchronization logic.

ISA_Sync Access = 0022h/0023h Regoffset = 059h

7	6	5	4	3	2	1	0
Rsv							SE
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-1	**Rsv**	**Reserved.** Writes have no affect. Reads return undefined value.
Bit 0	**SE**	**Synchronisation Enable.** (see table below: Table 9-24.).

Bit 0	Synchronisation Enable
0	Enabled
1	Disabled

Table 9-24. Synchronisation Enable

Programming notes

This bit would normally be set only when the ISA clock is derived from PCI clock (that is index 50h, bit 4 is set to ‘1’). Setting this bit will result in a small improvement in ISA performance.

9.7 UPDATE HISTORY FOR ISA INTERFACE CHAPTER

Thefollowing changes have been carried out on the 06/08/99.

Table 6 has changed.

IO Address bits 15-0 have been changed from xxxx xx00 100x xxxx to xxxx xx00 1000 xxxx.

The following changes have been made to the ISA Interface Chapter from Release 1.2 to Release 1.3.

East Bridge reference has been changed for North Bridge Reference & West Bridge reference has been changed for West Bridge reference.

The following changes have been made to the ISA Interface Chapter from Revision 1.0 to Release 1.2.

Section	Change	Text
9.1.	Replaced	"~~internal~~" With "controler"
9.1.	Removed	"Every cycle initiated on the ISA bus is tried on the PCI bus. If the cycle is claimed by some PCI target, then data is read from or written to that target. If the PCI cycle is not claimed, and the cycle targets a West Bridge internal register, then that register is read from or written to. Otherwise, the target is expected to be on the ISA bus."
9.1.1.	Added	"Special cycles Certain PCI special cycles are detected and forwarded to the ISA bus. Special cycles in which data bits 15-0 contain either 0000h or 0001h on the PCI bus, shutdown and halt respectively, are snooped and passed onto the ISA bus. Byte enables and address bits 0 and 4 are passed from the PCI to the ISA as well to support decode of the special cycle by the ISA."
9.2.2.	Removed	"~~Whether an index registe is implemented in the East Bridge, West Bridge or both is indicated in the description of that register in this document~~."
9.2.6.	Removed	"~~The buffer will be refilled if the data requested by the current read is not in the buffer. Also, to avoid stale data, the buffer will be refilled for the first host memory read after an ISA bus master gets ownership of the bus, for the first host memory read after any ISA bus cycle which is not a host memory read, and for any ISA read of a byte in the buffer which has already been read since the buffer was last filled~~."
9.2.6.	Removed	"~~If a host memory read can be fulfilled without refilling the buffer, no PCI cycle is generated~~."
9.3.1.	Replaced	"~~four~~" With "three"
9.4.	Removed	"~~Detailed description of the 8237 DMA controller operation can be obtained from Intel Peripheral Components Data book~~."
9.5.	Replaced	Table 14, "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.2.	Replaced	Table 15, "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.3.	Replaced	Table 16, "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.4.	Replaced	"~~Bit 6 **ISA IOCHK# Enable.** This bit is set to a '1' when IOCHK# signal of the ISA bus is asserted. Once set, this bit is cleared by setting bit 3 of this register to a '1'. Bit 3 should be reset to a '0' to enable recording the next IOCHK#. IOCHK# generates NMI to the host CPU if NMI is enabled. This bit is read only~~." With "Bit 6 **IOCHK# .** This bit is set to a '1' when IOCHK# signal of the ISA bus is asserted. Once set, this bit is cleared by setting bit 3 of this register to a '1'. Bit 3 should be reset to a '0' to enable recording the next IOCHK#. IOCHK# generates NMI to the host CPU if NMI is enabled. This bit is read only.

UPDATE HISTORY FOR ISA INTERFACE CHAPTER

Section	Change	Text
9.5.4.	Replaced	"Bit 5 **ISA T/C 2 State.**" With "Bit 5 **T/C 2 State.**"
9.5.4.	Replaced	"Bit 1 **ISA Speaker Enable.**" With "Bit 1 **Speaker Enable.**"
9.5.6.	Replaced	Table 18, "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.6.	Remove	For a detailed description of the operation of the 8259 interrupt controller, refer to Intel Peripherals Components Data book.
9.5.7.	Replaced	Table 17, "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.7.	Removed	~~"Detailed description of the 8254 Timer/Counter operation can be obtained from Intel Peripheral Components Data book"~~
9.5.8.	Replaced	Table 19, "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.8.	Remove	Detailed description of the 8237 DMA controller operation can be obtained from Intel Peripheral Components Data book.
9.5.8.	Replace	Table 9-6. , "IO Address bits" "~~0000 00~~" With "XXXX XX"
9.5.8.	Added	(see Table 9-6)

10. IDE CONTROLLER

10.1. INTRODUCTION

The IDE controller provides 2 IDE channels, primary and secondary, for interfacing with up to 4 IDE drives. It supports PIO modes 0 to 4 plus DMA modes 0 to 2. The timings are individually programmable for all 4 IDE devices. Each channel has a 4 double word FIFO for data transfers which allows 4 levels of write posting or read prefetch. Accesses to the 8 bit non-data IDE registers bypass the FIFOs.

For each of the 4 drives there are 3 bits in the configuration registers which can selectively enable write posting, read prefetch and ATAPI read prefetch. If read prefetch is enabled, the IDE controller will prefetch data from the drive after the first read has been made. The prefetching will stop after 256 data reads (512 Bytes) which is the normal sector size. If the current command to the drive is ATAPI packet (A0h) or service (A2h) then the read prefetch will be disabled unless ATAPI read prefetch is set.

The 2 channels of the IDE controller can be individually programmed to operate in either legacy or native mode. In legacy mode the IDE interrupts are hardwired to INT 14 & 15. In native mode they both connect to PCI INTA. If legacy mode is selected INT 14 & 15 will not be available on the ISA bus even if IDE interrupts are disabled. In legacy mode the primary and secondary channels are hardwired to IO addresses 1F0h-1F7h and 170h-177h respectively and also 376h for both channels. In native mode the IO addresses are programmed by configuration registers. For informaion on PIO mode, please refer to the ATAPI Standard

The IDE controller provides DMA bus master transfer between IDE devices and system memory with scatter/gather capability. By performing the IDE data transfer as a bus master, the Bus Master Device offloads the CPU (no programmed IO for data transfer) and improves system performance in multitasking environments.

Before issuing the DMA command the system software must first create a Physical Region Descriptor (PRD) table in system memory. This table contains a list of pointers and byte counts for each entry. A register in the IDE controller is set to point to this table. The IDE DMA controller will read from system memory during DMA initialisation. Each entry in the PRD table is 8 bytes long and will have the format below:

10.2. PRD Table Entry

PRD1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EOT	Rsv														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOW															Rsv
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31	**OET**	This bit set to '1' if it is the last entry in the table
Bits 30-16	**Rsv**	**Reserved**
Bit 15-1	**NOWS**	**Number of 16 bit data packets**
Bit 0	**Rsv**	**Reserved;** This bit must be set to 0

PRD0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRPAS															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRPAS															Rsv
Default value after reset = undefined															

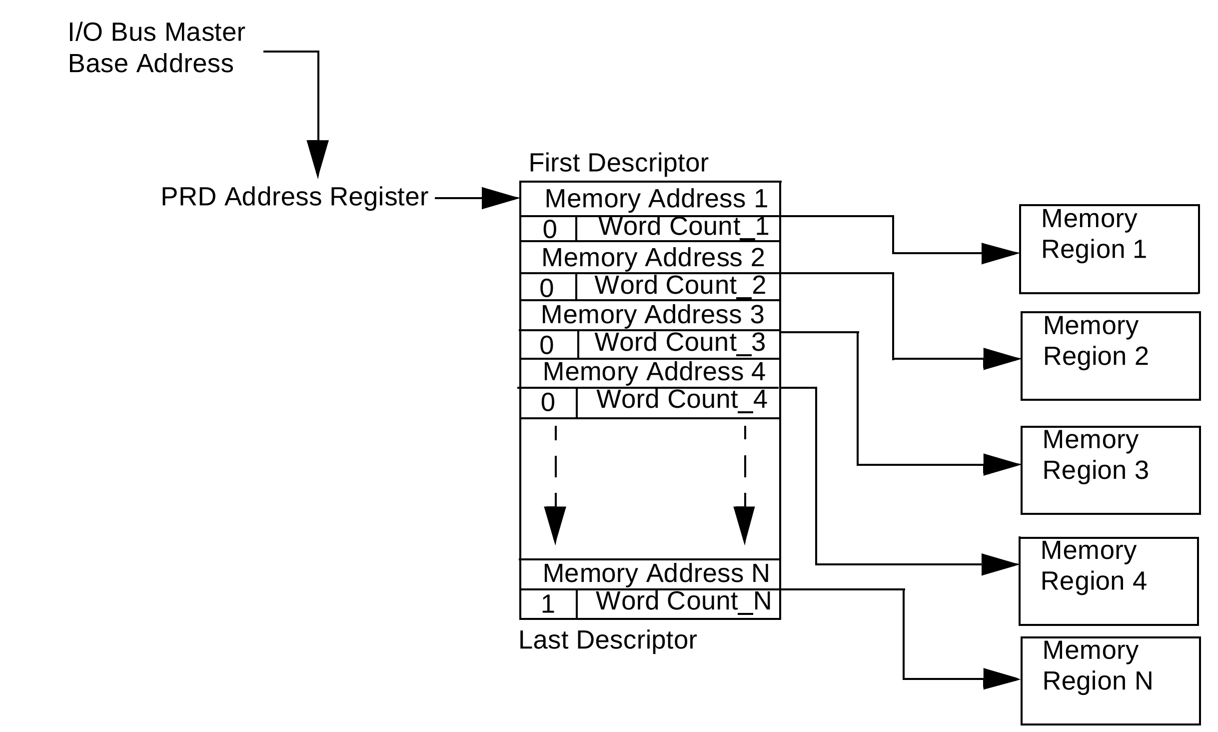
Bit Number	Mnemonic	Description
Bit 31-1	**MRPAS**	**Memory Physical Address of the first descriptor**
Bit0	**Rsv**	**Reserved;** This bit must be set to 0

The table must be aligned on a 4 byte boundary and should not cross a 64k boundary.

A memory region also should not cross a 64k boundaryboundary neither. An example of a PRD table is shown in Figure 10-1..

The primary and secondary channels each have an a PRD address pointer register.

Figure 10-1. PRD Table Entry Example



Note; For information on IO Bus Master Base Address, refer to Section 8.8.15. .

To save pins the IDE controller shares pins with the ISA interface. The On the IDE data bus, CS1 & CS3 signals are shared with the ISA address bus and keyboard controller/RTC pins. These signals are isolated by external transceiver devices. The ISAOE signal selects whether the pins are in IDE or ISA mode. The South Bridge arbitrates between the IDE controller and the ISA bus bridge to select which has control of the shared pins.

10.3. IDE Bus Master Registers

This document defines a register level programming interface for the internal busmaster ATA compatible (IDE) disk controller that directly moves data between IDE devices and main memory.

The system uses this programming interface will benefit from bundled software shipped with major OS's limiting the amount of software development required to provide a complete product.

The master mode programming interface is an extension of the standard IDE programming model. This means that devices can always be dealt with using the standard IDE programming model, with the master mode functionality used when the appropriate driver and devices are present. Master operation is designed to work with any IDE device that support DMA transfers on the IDE bus. Devices that only work in PIO mode can be used through the standard IDE programming model.

The programming interface defines a simple scatter/gather mechanism allowing large transfer blocks to be scattered to or gathered from memory. This cuts down on the number of interrupts to and interactions with the CPU. The interface defined here supports two IDE channels (primary and secondary).

10.3.1. Physical Region Descriptor Table

Before the controller starts a master transfer it is given a pointer to a Physical Region Descriptor Table. This table contains some a number of the Physical Region Descriptors (PRD) which defines the memory of areas that are involved in the data transfer. The descriptor table must be aligned on a 4 Byte boundary and the table cannot cross a 64KByte boundary in memory.

10.3.2. Physical Region Descriptor

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The data transfer will proceed until all regions described by the PRDs in the table have been transferred.

Each Physical Region Descriptor entry is 8 Bytes long.

- The first 4 bytes specify the byte address of a physical memory region.
- The next two bytes specify the count of the region in bytes (64K byte limit per region).

A value of zero in these two bytes indicates 64KByte. Bit 7 of the last byte indicates the end of the table; the Bus Master operation terminates when the last descriptor has been retired.

Note : The memory region specified by the descriptor is further restricted such that the region cannot straddle a 64K boundary. This means that the byte count can be limited to 64K, and the incrementer for the current address register need only extend from bit [1] to bit [15]. Also, the total sum of the descriptor byte counts must be equal to, or greater than the size of the disk transfer request. If greater than, then the driver must terminate the Bus Master transaction (by resetting bit zero of the command register to zero) when the drive issues an interrupt to signal transfer completion.

10.4. Bus Master IDE Register Description

The bus master IDE function uses 16 bytes of IO space. All bus master IDE IO space registers can be accessed as byte, word, word or Dword DWORD quantities. The description of the 16 bytes of IO registers followsfollows in table 7.2:

Offset	Register	R/W Status
00h	Bus Master IDE Command register Primary	R/W
01h	Device Specific	
02h	Bus Master IDE Status register Primary	RWC
03h	Device Specific	
04h-07h	Bus Master IDE PRD Table Address Primary	R/W
08h	Bus Master IDE Command register Secondary	R/W
09h	Device Specific	
0Ah	Bus Master IDE Status register Secondary	RWC
0Bh	Device Specific	
0Ch-0Fh	Bus Master IDE PRD Table Address Secondary	R/W

Table 10-1. Bus Master IDE Register Description

10.5. Bus Master IDE Command Register

10.5.1. IDE Command Register

This 8-bit Register is addressed at offset Base + 00h for the Primary IDE Channel and Base + 08h for the Secondary IDE Channel.

This register enables/disables Bus Master capability for the IDE function and provides direction control for the IDE DMA transfers. This register also provides the bits that software uses to indicate DMA capability of the IDE device.

IDE_COM

7	6	5	4	3	2	1	0
Rsv				RWCOM	Rsv		SSBM
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7-4	**Rsv**	**Reserved**
Bit 3	**RWCOM**	**Read Write Controle.** This bit sets the direction of the bus master transfer: when set to zero, PCI bus master reads are performed. When set to one, PCI bus master writes are performed: 0 = PCI bus master read 1 = PCI bus master write While a synchronous DMA transfer is in progress, this bit will be READ ONLY. The bit will return to read/write once the synchronous DMA transfer has been completed or halted. This bit must NOT be changed when the bus master function is actif
Bit 2-1	**Rsv**	**Reserved**
Bits 0	**SSBM**	**Stop/Start Bus Master.** Writing a ‘1’ to this bit enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from a zero to a one. The controller will transfer data between the IDE device and memory only when this bit is set. Master operation can be halted by writing a ‘0’ to this bit. All state information is lost when a ‘0’ is written; . Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., Bit 0= 1 in the Bus Master IDE Status Register for that IDE channel) and the drive has not yet finished its data transfer (bit 2=0 in the channel’s Bus Master IDE Status Register), the Bus Master command is said to be aborted and data transfered from the drive may be discarded before being written to the system memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the IDE Status register for that IDE channel being set, or both.

10.5.2. IDE Status Register

This 8-bit Register is addressed at offset Base + 02h for the Primary IDE Channel and Base + 0Ah for the Secondary IDE Channel.

This register provides status information about the IDE device and state of the IDE DMA transfer. Table describes IDE Interrupt Status and Bus Master IDE Active bit states after a DMA transfer has been started.

The IDE Status Register is illustrated in table 7.4 ;

IDE_COM

7	6	5	4	3	2	1	0
SO	D1DMA	D0DMA	Rsv		RWI	RWE	RWMI
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**SO**	**Simplex only.** This is hardwired to ’0’.
Bit 6	**D1DMA**	**Drive 1 DMA Capable.** This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.
Bit 5	**D0DMA**	**Drive 0 DMA Capable.** This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 0 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.
Bits 4-3	**Rsv**	**Reserved.** These bits return ‘0’ when read.
Bit 2	**RWI**	**Read/Write Interrupt.** This bit is set by the rising edge of the IDE interrupt line. This bit is cleared when a ‘1’ is written to it by software. Software can use this bit to determine if an IDE device has asserted its interrupt line. When this bit is read as a one, all data transfered from the drive is visible in system memory. For further details seeTable 10-2.
Bit 1	**RWE**	**Read/Write Error.** This bit is set when the controller encounters an error in transferring data to/from memory. The exact error condition is bus specific and can be determined in a bus specific manner. This bit is cleared when a ‘1’ is written to it by software.
Bit 0	**RWMI**	**Read/Write Bus Master IDE Active.** This bit is set to 1 when bit 0 in the Command register is set to 1. This bit is cleared (set to 0) when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the Command register. When this bit is read as a zero, all data transfered from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted. For further details seeTable 10-2.

Bit 2	Bit 0	Description
0	1	DMA transfer is in progress. No interrupt has been generated by the IDE device.
1	0	The IDE device generated an interrupt and the Physical Region Descriptors exhausted.This is normal completion where the size of the physical memory regions is equal to the IDE device transfer size.
1	1	The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case when the size of the physical memory regions is larger than the IDE device transfer size.
0	0	Error condition. If the IDE DMA Error bit is 1, there is a problem transferring data to/from

Table 10-2. Interrupt/Activity Status Combinations

10.5.3. Descriptor Table Pointer Register

This 32-bit Register is addressed at offset Base I/O+ 04h for the Primary IDE Channel and Base I/O+ 0Ch for the Secondary IDE Channel.

This register provides the base memory address of the Descriptor Table. The Descriptor Table must be DWord aligned and not cross a 4-Kbyte boundary in memory.

DT_Point

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADT															
Default value after reset = 00h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BADT														Rsv	
Default value after reset = 00h															

Bit Number	Mnemonic	Description
Bits 31-2	**BADT**	**Base address of Descriptor table.** This field corresponds to A[31-2].
Bits 1-0	**Rsv**	**Reserved**

The Descriptor Table must be Dword DWORD aligned. The Descriptor Table must not cross a 64K boundary in memory.

10.6. Operation

10.6.1. Standard Programming Sequence

To initiate a bus master transfer between memory and an Hard Disk device, the following steps are required:

1) Software prepares a PRD Table in system memory. Each PRD is 8 bytes long and consists of an address pointer to the starting address and the transfer count of the memory buffer to be transferred. In any given PRD Table, two consecutive PRDs are offset by 8-bytes and are aligned on a 4-byte boundary.

2) Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register . The direction of the data transfer is specified by setting the Read/Write Control bit. Clear the Interrupt bit and Error bit in the Status register.

3) Software issues the appropriate DMA transfer command to the disk device.

4) Engage the bus master function by writing a ‘1’ to the Start bit in the Bus Master IDE Command Register for the appropriate channel.

5) The controller transfers data to/from memory responding to DMA requests from the IDE device.

6) At the end of the transfer the IDE device signals an interrupt.

7) In response to the interrupt, software resets the Start/Stop bit in the command register. It then reads the controller status and then the drive status to determine if the transfer completed successfully.

10.7. Data Synchronization

When reading data from an IDE device, that data may be buffered by the IDE controller before using a master operation to move the data to memory. The IDE device driver in conjunction with the IDE controller is responsible for guaranteeing that any buffered data is moved into memory before the data is used.

The IDE device driver is required to do a read of the controller Status register after receiving the IDE interrupt. If the Status register returns with the Interrupt bit set then the driver knows that the IDE device generated the interrupt (important for shared interrupts) and that any buffered data has been flushed to memory. If the Interrupt bit is not set then the IDE device did not generate the interrupt and the state of the data buffers is unknown.

When the IDE controller detects a rising edge on the IDE device interrupt line (INTRQ) it is required to:

- Flush all buffered data
- Set the Interrupt bit in the controller Status register
- Guarantee that a read to the controller Status register does not complete until all buffered data has been written to memory.

Another way to view this requirement is that the first read to the controller Status register in response to the IDE device interrupt must return with the Interrupt bit set and with the guarantee that all buffered data has been written to memory.

10.7.1. Status Bit Interpretation

The table below gives a description of how to interpret the Interrupt and Active bits in the Controller status register after a DMA transfer has been started.

Interrupt bit	Active bit	Description:
0	1	DMA transfer is in progress. No interrupt has been generated by the IDE device.
1	0	The IDE device generated an interrupt. The controller exhausted the Physical Region Descriptors. This is the normal completion case where the size of the physical memory regions was equal to the IDE device transfer size.
1	1	The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case where the size of the physical memory regions was larger than the IDE device transfer size.
0	0	This bit combination signals an error condition. If the Error bit in the status register is set, then the controller has some problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is not set, then the PRD's specified a smaller size than the IDE transfer size.

10.8. Error Conditions

IDE devices are sector based mass storage devices. The drivers handle errors on a sector by sector basis; either a sector is transferred successfully or it is not. If the IDE DMA slave device never completes the transfer due to a hardware or software error, the Bus Master IDE command will eventually be stopped (by setting Command Start bit to zero) when the driver times out the disk transaction. Information in the IDE device registers will help isolate the cause of the problem.

If the controller encounters an error while doing the bus master transfers it will stop the transfer (ie. reset the Active bit in the Command register) and set the ERROR bit in the Status register. The controller does not generate an interrupt when this happens. The device driver can use device specific information (e.g.; PCI Configuration Space Status register) to determine what caused the error.

Whenever a requested transfer does not complete properly, information in the IDE device registers (Sector Count) can be used to determine how much of the transfer was completed and to construct a new PRD table to complete the requested operation. In most cases the existing PRD table can be used to complete the operation.

10.9. PCI Specifics

Bus master IDE controllers built to attach to a PCI bus must have the following characteristics:

1) The Class Code in PCI configuration space indicates IDE device and bit 7 of the Programming Interface register (offset 0x09) in PCI configuration space must be set to 1 to indicate that the device supports the Master IDE capability.

2) The control registers for the controller are allocated via the devices Base Address register at offset 0x20 in PCI configuration space.

3) In the controller Status register the Error bit will be set and the Active bit reset if any of the following conditions occur on the PCI bus while the controller is doing a master operation on the bus. The exact cause can be determined by examining the Configuration Space Status register.

Error Condition	Configuration Space Status bits
Target Abort	Any time bit 12 of the Config Space Status register is set.
Master Abort	Any time bit 13 of the Config Space Status register is set.
Data Parity	Any time bit 8 of the Config Space Status register is set.
Error Detected	

10.10 UPDATE HISTORY FOR IDE INTERFACE CHAPTER

The following changes have been made to the IDE Interface Chapter.

All Reference to Bus Master has been removed. This includes all the following text;

The IDE controller provides DMA bus master transfer between IDE devices and system memory with scatter/gather capability. By performing the IDE data transfer as a bus master, the Bus Master Device offloads the CPU (no programmed IO for data transfer) and improves system performance in multitasking environments.

Before issuing the DMA command the system software must first create a physical region descriptor (PRD) table in system memory. This table contains a list of pointers and byte counts for each entry. A register in the IDE controller is set to point to this table. The DMA controller will read from system memory during DMA operation. Each entry in the PRD table is 8 bytes long and will have the format below:

PRD Table Entry

This is illustrated in table 7.1 below

63	62......48	47...............33	32	31..1	0
EOT	Reserved	Number of words	ignored	Memory region physical address start	ignored

Bit 63 **EOT.** Is ‘1’ if last entry in table.

Bits 62-48 *Reserved.*

Bits 47-33 **Number of words.**

Bit 32 *Ignored.* The byte count must be even.

Bits 31-1 **Memory region physical address start.**

Bit 0 *Ignored.* The memory region must be word aligned.

The table must be aligned on a 4 byte boundary and should not cross a 64k boundary. A memory region also should not cross a 64k boundaryboundary neither. An example of a PRD table is shown in Figure ..

The primary and secondary channels each have an a PRD address pointer register.

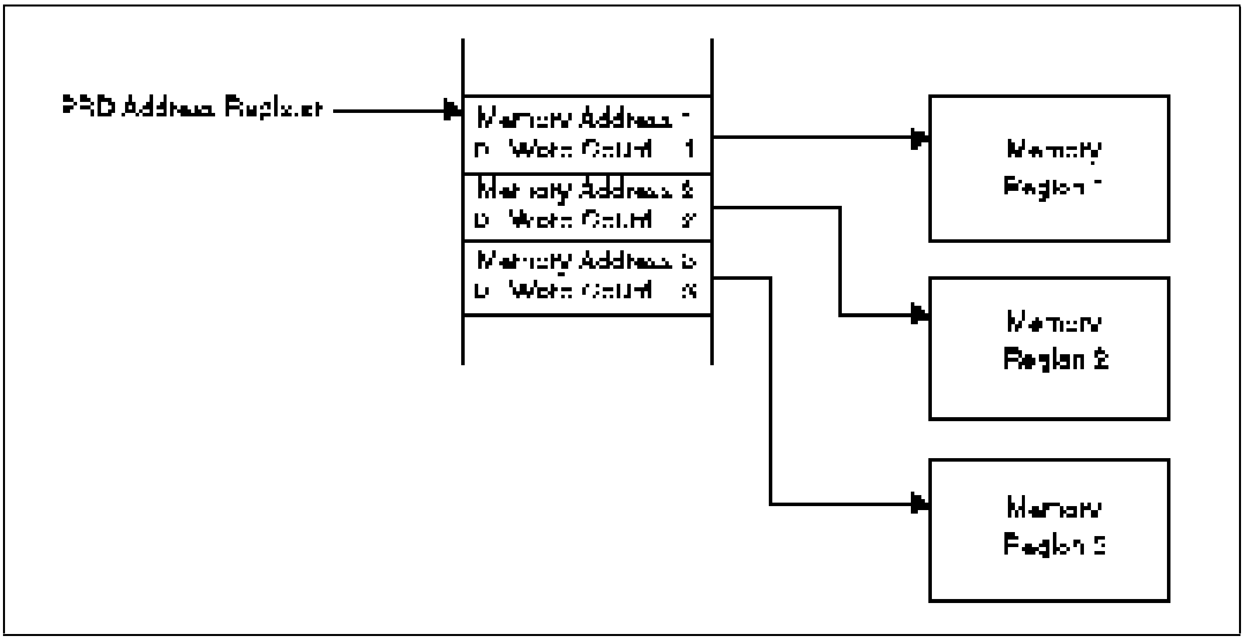
To save pins the IDE controller shares pins with the ISA interface. The On the IDE data bus, CS1 & CS3 signals are shared with the ISA address bus and keyboard controller/RTC pins. These signals are isolated by external transceiver devices. The ISAOE signal selects whether the pins are in IDE or ISA mode. The East Bridge arbitrates between the IDE controller and the ISA bus bridge to select which has control of the shared pins.

IDE Bus Master Registers

Bus Master This document defines a register level programming interface for a busmaster ATA compatible (IDE) disk controller that directly moves data between IDE devices and main memory.

Controllers that implement this programming interface will benefit from bundled software shipped with major OS’s limiting the amount of software development required to provide a complete product.

PRD Table Entry Example



The master mode programming interface is an extension of the standard IDE programming model. This means that devices can always be dealt with using the standard IDE programming model, with the master mode functionality used when the appropriate driver and devices are present. Master operation is designed to work with any IDE device that support DMA transfers on the IDE bus. Devices that only work in PIO mode can be used through the standard IDE programming model.

The programming interface defines a simple scatter/gather mechanism allowing large transfer blocks to be scattered to or gathered from memory. This cuts down on the number of interrupts to and interactions with the CPU. The interface defined here supports two IDE channels (primary and secondary). Individual controllers that support more than two channels will need to appear to software as multiple controllers if the standard drivers are to be used. Master IDE controllers should default to Mode 0 Multiword DMA timings to ensure operation with DMA capable IDE devices without the need for controller-specific code to initialize controller-specific timing parameters.

Physical Region Descriptor Table

Before the controller starts a master transfer it is given a pointer to a Physical Region Descriptor Table. This table contains some a number of Physical Region Descriptors (PRD) which describe define areas of memory that are involved in the data transfer. The descriptor table must be aligned on a 4 byte boundary and the table cannot cross a 64K boundary in memory.

Physical Region Descriptor

The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The data transfer will proceed until all regions described by the PRDs in the table have been transferred.

Each Physical Region Descriptor entry is 8 bytes in length. The first 4 bytes specify the byte address of a physical memory region. The next two bytes specify the count of the region in bytes (64K byte limit per region). A value of zero in these two bytes indicates 64K. Bit 7 of the last byte indicates the end of the table; bus master operation terminates when the last descriptor has been retired.

Note : The memory region specified by the descriptor is further restricted such that the region cannot straddle a 64K boundary. This means that the byte count can be limited to 64K, and the incrementer for the current address register need only extend from bit [1] to bit [15]. Also, the total sum of the descriptor byte counts must be equal to, or greater than the size of the disk transfer request. If greater than, then the driver must terminate the Bus Master transaction (by resetting bit zero of the command register to zero) when the drive issues an interrupt to signal transfer completion.

Bus Master IDE Register Description

The bus master IDE function uses 16 bytes of IO space. All bus master IDE IO space registers can be accessed as byte, word, word or Dword DWORD quantities. The description of the 16 bytes of IO registers followsfollows in table 7.2:

Offset	Register	R/W Status
00h	Bus Master IDE Command register Primary	R/W
01h	Device Specific	
02h	Bus Master IDE Status register Primary	RWC
03h	Device Specific	
04h-07h	Bus Master IDE PRD Table Address Primary	R/W
08h	Bus Master IDE Command register Secondary	R/W
09h	Device Specific	
0Ah	Bus Master IDE Status register Secondary	RWC
0Bh	Device Specific	
0Ch-0Fh	Bus Master IDE PRD Table Address Secondary	R/W

Table 7.2

Offset	Register	R/W Status
00h	Bus Master IDE Command register Primary	R/W
01h	Device Specific	
02h	Bus Master IDE Status register Primary	RWC
03h	Device Specific	
04h-07h	Bus Master IDE PRD Table Address Primary	R/W
08h	Bus Master IDE Command register Secondary	R/W
09h	Device Specific	
0Ah	Bus Master IDE Status register Secondary	RWC
0Bh	Device Specific	
0Ch-0Fh	Bus Master IDE PRD Table Address Secondary	R/W

Bus Master IDE Command Register

IDE Command Register

The IDE Command Register is illustrated in table 7.3.

7..........4	3	2...........1	0
Reserved	Read/Write controle	Reserved	Start/Stop bus master

This 8-bit Register is addressed at offset Base + 00h for the Primary IDE Channel and Base + 08h for the Secondary IDE Channel.

Bits 7-4 *Reserved.* These bits return ‘0’ when read.

Bit 3 **Read or Write Control.** This bit sets the direction of the bus master transfer: when set to zero, PCI bus master reads are performed. When set to one, PCI bus master writes are performed. This bit must NOT be changed when the bus master function is active.:

0 = PCI bus master read

1 = PCI bus master write

This bit must NOT be changed when the bus master function is actif

Bits 2-1*Reserved.* These bits return ‘0’ when read.

Bit 0 **Start/Stop Bus Master.**

1 = Start Bus Master.

Bus master operation begins when this bit is detected changing from ‘0’ to ‘1’. The controller will transfer data between the IDE device and memory only when this bit is set.

0 = Stop BUS Master.

Bit 0 **Start/Stop Bus Master.** Writing a ‘1’ to this bit enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from a zero to a one. The controller will transfer data between the IDE device and memory only when this bit is set. Master operation can be halted by writing a ‘0’ to this bit. All state information is lost when a ‘0’ is written; . Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., the IDE Active bit of the IDE Status register for that IDE channel is set) and the drive has not yet finished its data transfer (The Interupt bit in the IDE Status register for that IDE channel is not set), the bus master command is said to be aborted and data transfered from the drive may be discarded before being written to the system memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the IDE Status register for that IDE channel being set, or both.

Reset value is 00h.

IDE Status Register

This 8-bit Register is addressed at offset Base + 02h for the Primary IDE Channel and Base + 0Ah for the Secondary IDE Channel.

The IDE Status Register is illustrated in table 7.4 ;

Table 7.4

7	6	5	4.........3	2	1	0
Simplex only	Drive 1 DMA capable	Drive0 DMA capable	Reserved	Read/Write Interupt	Read/Write Error	Read/Write Master IDE Bus actif

Bit 7 **Simplex only.** This read-only bit indicates whether or not both bus master channels (primary and secondary) can be operated at the same time.

‘0’ = The channels operate independently and can be used at the same time.

‘1’ = Only one channel may be used at a time.

IDE Status Register

This 8-bit Register is addressed at offset Base + 02h for the Primary IDE Channel and Base + 0Ah for the Secondary IDE Channel.

Bit 7 **Simplex only.** This read-only bit indicates whether or not both bus master channels (primary and secondary) can be operated at the same time. If the bit is a ‘0’, then the channels operate independently and can be used at the same time. If the bit is a ‘1’, then only one channel may be used at a time.

Bit 6 **Drive 1 DMA Capable.** This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.

Bit 5 **Drive 0 DMA Capable.** This read/write bit is set by device dependent code (BIOS or device driver) to indicate that drive 0 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance.

Bits 4-3 *Reserved.* These bits return ‘0’ when read.

Bit 2 Read/Write **Interrupt.** This bit is set by the rising edge of the IDE interrupt line. This bit is cleared when a ‘1’ is written to it by software. Software can use this bit to determine if an IDE device has asserted its interrupt line. When this bit is read as a one, all data transfered from the drive is visible in system memory.

Bit 1 Read/Write **Error.** This bit is set when the controller encounters an error in transferring data to/from memory. The exact error condition is bus specific and can be determined in a bus specific manner. This bit is cleared when a ‘1’ is written to it by software.

Bit 0 Read/Write **Bus Master IDE Active.** This bit is set when the Start bit is written to the Command register. This bit is cleared when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared when the Start bit is cleared in the Command register. When this bit is read as a zero, all data transfered from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.

Reset value is 00h.

Descriptor Table Pointer Register

This 32-bit Register is addressed at offset Base + 04h for the Primary IDE Channel and Base + 0Ch for the Secondary IDE Channel.

This register is defined in Section

Bits 32-0 **Region Descriptor Pointer.**

Bits 31-2 **Base address of Descriptor table.** This field corresponds to A[31-2].

Bits 1-0 *Reserved.*

The Descriptor Table must be Dword DWORD aligned. The Descriptor Table must not cross a 64K boundary in memory.

Reset value is 0000 0000h

Operation

11. VGA CONTROLLER

11.1. INTRODUCTION

The STPC integrates a full VGA Controller with Extended Functions together with a Color Digital to Analog output (RAMDAC) and a Graphics Engine. The VGA Controller provides the basic video display function. It generates the timing and logic required to create an output data stream from the video buffer and the appropriate horizontal and vertical synchronisation pulses. The Frame video buffer uses the first 4Mb of the DRAM space. This Frame buffer area is selected upon configuration of the VGA video output and, once selected the function of the Frame buffer area can not be eassely changed back to normal DRAM memory program/data functions until the next reset cycle because of complete memory remapping.

The on-chip triple RAMDAC runs at up to 135MHz, using an external frequency synthesizer, allowing a display up to 1280x1024 at 75Hz. Color is handled using 8-, 16-, 24- or 32-bits per pixel. VDU Graphics standards can be read through the on-chip Display Data Channel (DDC) link.

11.2. VGA CONTROLLER

The VGA controller of the STPC is 100% backward compatible with the VGA standard specification. In addition, enhancements made to the VGA standard are detailed in the following sub-sections.

Resolutions of up to 1024 x 768 and color depths of 8, 16, 24 and 32 bits per pixel are supported. The integrated RAMDAC supports digital to analog conversion rates up to 135 MHz. This along with peak video bandwidth of 320 MBytes/sec (using EDO DRAM) enables the VGA controller to support 1024 x 768 x24 and 800 x 600 x 32 resolutions at 75 Hz refresh rate.

To support vertical resolutions up to 1024 pixels, vertical timing parameters have been extended from 10 to 11 bits. The VGA defined horizontal timing parameters are compatible with the above resolutions. The horizontal and vertical timing counters and the sync and blank generation logic operate synchronously to DCLK which can be up to 135MHz in frequency.

Pixel color depths are specified by programming the Palette Control register (CR28) appropriately. Eight bit color modes use the RAMDAC look-up table to form 18 or 24 bit colors. All other modes bypass the look-up table and drive the DACs directly.

The Graphics Core is capable of using up to 4Mb of available memory as its frame buffer. The Cathode Ray Tube Controller (CRTC) Start Address uses 20-bits to allow for locating the frame buffer at any double word boundary within this 4Mb of memory. This frame buffer sits within the 16MBytes Graphics buffer area. Refer to the Graphics Engine Section for further details on the Graphics Memory Architecture.

Video data is automatically extracted from the frame buffer by the CRTC, a FIFO structure ensures that the video display is continually refreshed without loss of data and visual artifacts. Independent high and low level watermarks can be programmed to accelerate or decelerate the demands on the memory arbitration logic.

The CRTC can be programmed to support interlaced monitors and timings. It also supports hardware generated cursor in text mode and a 64x64 bit cursor in Graphics modes. This graphics mode cursor is software programmable with separate programmable XOR and AND masks in memory.

If an external add-in VGA card is placed in the system, the on-chip VGA controller can be disabled in order to work with this external card. It is possible to enable / disable the system back to dual use VGA controller if necessary.

11.3. VGA REGISTERS

The following sections describe both the standard VGA compatible register definitions and the definitions of register extensions specific to the STPC VGA controller.

The ‘X’ within some IO addresses represents a ‘B’ if monochrome operation is enabled and a ‘D’ if color operation is in effect.

11.4. GENERAL VGA REGISTERS

11.4.1. MOTHERBOARD ENABLE REGISTER (RW)

MBEN Access = 0x094h Regoffset =

7	6	5	4	3	2	1	0
Rsv		ME	Rsv	MBEN	Rsv		
Default value after reset = 28h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**, read as ‘0’s.
Bits 5	**ME**	**Motherboard Enable.** If the VGA is configured to operate on the motherboard, then when this bit is set to ‘0’, it allows read and write access to port 102h. All other IO and memory accesses are ignored. When set to a ‘1’, this bit allows access to all IO and memory, but access to port 102h is ignored.
Bits 4	**Rsv**	**Reserved**, reads as ‘0’.
Bits 3	**MBEN**	**MBEN Video System Enable**. When ‘0’ this bit disables all IO and memory accesses to the VGA as well as the DAC registers. Accesses to 94h remain enabled. When ‘1’, Video system enable bits of port 0102h and 03C3h determine the accessibility of the VGA. The VGA continues to display video data while disabled.
Bits 2-0	**Rsv**	**Reserved**, read as ‘0’s.

Programming notes

The contents of this register are not altered by drawing operations.

11.4.2. ADD-IN VGA ENABLE REGISTER (RW)

ADDEN Access = 0x46E8h Regoffset =

7	6	5	4	3	2	1	0
Rsv			AE	ADDEN VSE	Rsv		
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-5	**Rsv**	**Reserved***,* read as 0's.
Bits 4	**AE**	**Addin Enable.** If the VGA is configured to operate on an add-in card, then when this bit is set to '1', it allows read and write access to port 102h. All other IO and memory accesses are ignored. When set to a '0', this bit allows access to IO and memory, but access to port 102h is ignored.
Bits 3	**ADDEN VSE**	**ADDEN Video System Enable.** When '0' this bit disables all IO and memory accesses to the VGA as well as the DAC registers. Accesses to 46E8h remain enabled. When '1', Video system enable bits of port 0102h determine the accessibility of the VGA. The VGA continues to display video data while disabled.
2-0	**Rsv**	**Reserved,** read as '0's.

Programming notes

The contents of this register are not altered by drawing operations.

11.4.3. VIDEO SUBSYSTEM ENABLE 1 REGISTER (RW)

VSE1 Access = 0x102h Regoffset =

7	6	5	4	3	2	1	0
Rsv							VSE 1
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-1	**Rsv**	**Reserved**, read as '0's.
Bit 0	**VSE 1**	**Video System Enable.** When '0', this bit disables all IO and memory accesses to the VGA as well as DAC registers except port 102h. Port 102h remains accessible to allow enabling of the VGA. Ports 46E8h and 94h are also not affected by this bit. The VGA continues to display video data while disabled.

Programming notes

The contents of this register are not altered by drawing operations.

11.4.4. VIDEO SUBSYSTEM ENABLE 2 REGISTER (RW)

VSE2 Access = 0x3C3h Regoffset =

7	6	5	4	3	2	1	0
Rsv							VSE 2
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-1	**Rsv**	**Reserved**, read as ’0’s.
Bit 0	**VSE 2**	**Video System Enable.** When ‘0’, this bit disables all IO and memory accesses to the VGA as well as DAC registers except ports 102h and 3C3h. Ports 102h and 03C3h remain accessible to allow enabling of the VGA. Port 94h is also not affected by this bit. The VGA continues to display video data while disabled.

Programming notes

The contents of this register are not altered by drawing operations.

11.4.5. MISCELLANEOUS OUTPUT REGISTER (RW)

MISC			Access = 0x3CCh/0x3C2h				Regoffset =
7	6	5	4	3	2	1	0
VRP	HRP	OEPS	Rsv	CS		E RAM	IO A
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**VRP**	**Vertical retrace polarity.** (see table below: Table 11-1.).
Bit 6	**HRP**	**Horizontal retrace polarity**. (see table below: Table 11-2.).For older IBM compatible color monitors, the polarity of the vertical and horizontal retrace pulses was used to define the vertical scan rate, as follows in Table 11-3.
Bit 5	**OEPS**	**Odd/Even Page Select.** This bit selects between two 64K pages of memory (of a 128K plane) when the VGA is in odd/even mode (replaces the least significant bit of the memory address). ‘0’ = low 64K page. ‘1’ = high 64K page. This bit is only effective in Mode 0, 1, 2, 3, or 7.
Bit 4	**Rsv**	**Reserved**, reads as ‘0’.
Bits 3-2	**CS**	**Clock Selects.** Selects one of the four synthesizer pairs when DCLK source is onchip PLL’s.
Bit 1	**E RAM**	**Enable RAM.** When ‘0’, this bit disables host accesses to the display RAM. The access to the ROM, however, remains enabled. Setting this bit to ‘1’ enables accesses to the display buffer.
Bit 0	**IO A**	**IO Address.** This bit defines the address map of the following registers (see table below: Table 11-4.).

Bit 7	Vertical retrace polarity
0	active high
1	active low

Table 11-1. Vertical retrace polarity

Bit 6	Horizontal retrace polarity
0	active high
1	active low

Table 11-2. Horizontal retrace polarity

Bit7	Bit6	Active Lines	Vertical Total
0	0	Reserved	Reserved
0	1	400 lines	414 lines
1	0	350 lines	362 lines
1	1	480 lines	496 lines

Table 11-3. Vertical and horizontal polarities combination

Register	Bit 0 = ’0’	Bit 0 = ’1’
CRTC Registers	03BXh	03DXh
Input #1 Register	03BAh	03DAh

Table 11-4. IO Address

Programming notes

The contents of this register are not altered by drawing operations.

11.4.6. INPUT STATUS REGISTER #0 (R)

INP0 Access = 0x3C2h Regoffset =

7	6	5	4	3	2	1	0
VRF	Rsv		R S	Rsv			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**VRF**	**Vertical Retrace Flag.** This bit is set at the beginning of the vertical retrace period if bit 4 of CR11 (Vertical Retrace End register) is set to one. Once set, this bit is cleared when bit 4 of CR11 is reset to 0. This recording of the vertical retrace interrupt is independent of bit 5 (disable vertical interrupt) of CR11. See the description of CR11 for more details.
Bits 6-5	**Rsv**	**Reserved.** These bits read as ones.
Bit 4	**RS**	**RAMDAC Sense.** This bit is connected to the SENSE signal of the RAMDAC. It is used by the BIOS to auto-detect the monitor type.
Bits 3-0	**Rsv**	**Reserved.** These bits read as zero.

11.4.7. INPUT STATUS REGISTER *#1 (R)*

INP1 Access = 0x3XAh Regoffset =

7	6	5	4	3	2	1	0
Rsv		DU		VR	Rsv		R
Default value after reset =							

Bit Number	Mnemonic	Description
Bit 7-6	**Rsv**	**Reserved.** These bits read as zero.
Bits 5-4	**DU**	**Diagnostic Use.** These bits reflect 2 of the 8 bit video output data during display periods and overscan color data during non-display periods. Selection of one of four pairs of bits is controlled by bits 5-4 of the AR12 as in Table 11-5.
Bit 3	**VR**	**Vertical Retrace.** A one in this position indicates that a vertical retrace is in progress.
Bits 2-1	**Rsv**	**Reserved.** Bit 2 reads as one; bit 1 reads as zero.
Bit 0	**R**	**Retrace.** A one in this position indicates that a horizontal OR vertical retrace is in progress.

AR12		**Diagnostic Bits**	
Bit 5	**Bit 4**	**Bit 5**	**Bit 4**
0	0	Video2	Video0
0	1	Video5	Video4
1	0	Video3	Video1
1	1	Video7	Video6

Table 11-5. Video output data Selection

11.5. VGA SEQUENCER REGISTERS

11.5.1. SEQUENCER INDEX REGISTER (RW)

SRX Access = 0x03C4h/0x03C5h Regoffset =

7	6	5	4	3	2	1	0
Rsv					SI		
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as 0's.
Bit 3		
Bits 2-0	**SI**	**Sequencer Index.** These bits point to the register that is accessed by the next read or write to port 03C5h.

Programming notes

The contents of this register are not altered by drawing operations.

11.5.2. SEQUENCER RESET REGISTER (RW)

SR0 Access = 0x03C4h/0x03C5h Regoffset = 000h

7	6	5	4	3	2	1	0
Rsv						SR	AR
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-2	**Rsv**	**Reserved**, reads as ’0’s.
Bit 1	**SR**	**Synchronous Reset.** When set to ‘0’ terminates display memory accesses. This bit, as well as bit 0 of this register, must be set to ‘1’ to enable sequencer operations. The Clocking Mode register (SR1) bits 0 and 3, and Miscellaneous Output register bits 2-3 must not be changed unless this bit is set to ‘0’ to avoid loss of memory contents.
Bit 0	**AR**	**Asynchronous Reset.** This bit performs the same function as bit 1 except when set from ‘1’ to ‘0’, it also clears the Character Map select register (SR3) to ‘0’.

Programming notes

The contents of this register are not altered by drawing operations.

11.5.3. SEQUENCER CLOCKING MODE REGISTER (RW)

SR1			Access = 0x03C4h/0x03C5h			Regoffset = 001h	
7	6	5	4	3	2	1	0
Rsv		SO	S4	DC	SL	Rsv	8/9 DC
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**, reads as '0's.
Bit 5	**SO**	**Screen Off.** Setting this bit to '1' blanks the screen by driving black color (not overscan) on the screen. This facilitates the CPU to access video memory at maximum possible bandwidth.
Bit 4	**S4**	**Shift4.** Along with Shift Load (bit 2) this bit controls the loading of the video serializers as in Table 11-6.
Bit 3	**DC**	**Dot Clk.** When '0' sets the dot clock to be the same as the input dot clock. When '1', divides the input dot clock by 2 to derive the dot clock. The input dot clock is divided by 2 for 320 and 360 horizontal pixel modes 0, 1, 4, 5, D and 13. This is can not be used when using an external DCLK
Bit 2	**SL**	**Shift Load**, see Bit 4 - Shift4.
Bit 1	**Rsv**	**Reserved,** reads as '0'.
Bit 0	**8/9 DC**	**8/9 Dot Clock.** When '0', this bit causes the character clock to be 9 dots wide. When '1', an 8-dot wide character clock is selected.

Bit4	Bit2	Video Serializer Load clock	Resolution
0	0	Every character	720 dots/line
0	1	Every second character	360 dots/line
1	X	Every fourth character	180 dots/line

Table 11-6. Video Serializer Load clock

Programming notes

The contents of this register are not altered by drawing operations.

11.5.4. SEQUENCER PLANE MASK REGISTER (RW)

SR2				Access = 0x03C4h/0x03C5h			Regoffset = 002h
7	6	5	4	3	2	1	0
Rsv				EP3	EP2	EP1	EP0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as ’0’.
Bit 3	**EP3**	**Enable Plane 3**, Write enable for plane 3. A ‘0’ in this bit disables writes to plane 3.
Bit 2	**EP2**	**Enable Plane 2**
Bit 1	**EP1**	**Enable Plane 1**
Bit 0	**EP0**	**Enable Plane 0**

The planes are used in different manners by the various modes. These are shown in Table 11-7.

Mode	Plane 0	Plane 1	Plane 2	Plane 3
Text Modes 0,1, 2, 3, 7	Character Data	Attribute Data	Font Data	Unused
16-bit Color Graphics Modes D, E, 10, 12	Pixel Bit 0	Pixel Bit 1	Pixel Bit 2	Pixel Bit 3
4-Color Mono Graphics Mode F	Video	Ignored	Intensity	Ignored
4-Color Modes 4, 5	Even Byte	Odd Byte	Unused	Unused
2-Color Mono Graphics Mode 6	Even Byte	Odd Byte	Unused	Unused
2-Color Mono Graphics Mode 11	All Bytes	Unused	Unused	Unused
256-Color Graphics Mode 13	Byte 0	Byte 1	Byte 2	Byte 3

Table 11-7. Various modes

Programming notes

The contents of this register are not altered by drawing operations.

11.5.5. SEQUENCER CHARACTER MAP REGISTER (RW)

SR3 Access = 0x03C4h/0x03C5h Regoffset = 003h

5	4	3	2	1	0
SFB	PFB	SFB	SFB	PFB	PFB
Default value after reset = 00h					

Bit Number	Mnemonic	Description
Bit 5	**SFB**	**Secondary Font Block Select bit 0.**
Bit 4	**PFB**	**Primary Font Block Select bit 0.**
Bit 3	**SFB**	**Secondary Font Block Select bit 2.**
Bit 2	**SFB**	**Secondary Font Block Select bit 1.**
Bit 1	**PFB**	**Primary Font Block Select bit 2.**
Bit 0	**PFB**	**Primary Font Block Select bit 1.**

Programming notes

Used in text mode to select the primary and secondary font tables when the attribute bit 3 is ‘0’ (for primary) or ‘1’ (for secondary) as per the Table 11-8. and Table 11-9. following*:*

Bit 1	Bit 0	Bit 4	Font block #	Table Location
0	0	0	0	0K
0	0	1	4	8 K
0	1	0	1	16 K
0	1	1	5	24 K
1	0	0	2	32 K
1	0	1	6	40 K
1	1	0	3	48 K
1	1	1	7	56 K

Table 11-8. Primary Font

Bit 3	Bit 2	Bit 5	Font block #	Table Location
0	0	0	0	0 K
0	0	1	4	8 K
0	1	0	1	16 K
0	1	1	5	24 K
1	0	0	2	32 K
1	0	1	6	40 K
1	1	0	3	48 K
1	1	1	7	56 K

Table 11-9. Secondary Font

This register is reset to ‘0’ by the asynchronous reset via SR0 register.

The contents of this register are not altered by drawing operations.

11.5.6. SEQUENCER MEMORY MODE REGISTER (RW)

SR4 Access = 0x03C4h/0x03C5h Regoffset = 004h

7	6	5	4	3	2	1	0
Rsv				C4 A	OE	EM	Rsv
Default value after reset = 04h							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as ’0’s.
Bit 3	**C4 A**	**Chain-4 Addressing.** When set to ‘1’, this bit forces the two least significant host address bits to select the display buffer plane to be accessed by a host read or write. HA1-0 = ‘00’ selects plane 0, HA1-0 = ‘01’ selects plane 1, etc. For writes, the plane selected by the two address bits still must be enabled via the Plane Mask Register (SR2) in order for the writes to take place. During read transfers, when this bit is set to ‘1’, the Graphics Control Read Map register (GR4) is ignored and the Byte from the plane selected by the two least significant host address bits is returned.
Bit 2	**OE**	**Odd/Even# Addressing.** Similar to the Chain-4 bit in that when set to ‘0’ forces the least significant host address bit to select two of the four display planes for host transfers. HA0 = ‘0’ selects planes 0 and 2, and HA1 = ‘1’ selects planes 1 and 3. Selected planes are ANDed with the Plane Mask register (SR2) to generate the plane write enables during write transfers. Read transfers use Map Select bit 1 from GR4 along with HA0 to select one of the 4 Bytes to be returned to the host. Read Map select bit 0 is not used when odd/even addressing mode is enabled.
Bit 1	**EM**	**Extended Memory.** When this bit is ‘0’ it indicates 64K of display memory is present. When ‘1’, indicates that 256K of display memory is present.
Bit 0	**Rsv**	**Reserved,** reads as ‘0’.

Programming notes

The contents of this register are not altered by drawing operations.

11.5.7. EXTENDED REGISTER LOCK/UNLOCK REGISTER (RW)

SR6 Access = 0x03C4h/0x03C5h Regoffset = 006h

7	6	5	4	3	2	1	0
ER LU							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	**ER LU**	**Extended Registers Lock/Unlock.** When written to with 57h, all extended registers are unlocked. When written to with any value other than 57h, all extended registers are locked. When the extended registers are in the locked state, reads to this register return a zero. When the extended registers are in the unlocked state, reads to this register return a ‘1’.

Programming notes

The contents of this register are not altered by drawing operations.

11.6. GRAPHICS CONTROLLER REGISTERS

11.6.1. GRAPHICS CONTROLLER INDEX REGISTER (RW)

GRX Access = 0x03CEh/0x03CFh Regoffset =

7	6	5	4	3	2	1	0
Rsv				GCI			
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as ’0’s.
Bits 3-0	**GCI**	**Graphics Controller Index.** These bits point to the register that is accessed by the next read or write to port 03CFh.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.2. GRAPHICS SET/RESET REGISTER (RW)

GR0 Access = 0x03CEh/0x03CFh Regoffset = 000h

7	6	5	4	3	2	1	0
Rsv				GCSR			
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as ’0’s.
Bits 3-0	**GCSR**	**Graphics Controller Set/Reset.** These bits define the value written to the four memory planes. In Write Mode 0, only the planes enabled by the Enable Set/Reset Register (GR1) are written to. In Write Mode 3, the contents of the Set/Reset register are always enabled. Bit 0 corresponds to memory plane 0, bit 1 to memory plane 1, etc.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.3. GRAPHICS ENABLE SET/RESET REGISTER (RW)

GR1 Access = 0x03CEh/0x03CFh Regoffset = 001h

7	6	5	4	3	2	1	0
Rsv				GCSR			
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as ’0’s.
Bits 3-0	**GCSR**	**Graphics Controller Enable Set/Reset.** These bits define which memory planes are to be written to with the value of the corresponding Set/Reset Register (GR0) in Write Mode 0. In Write Mode 3, the Enable Set/Reset register has no affect. Bit 0 corresponds to memory plane 0, bit 1 to memory plane 1, etc.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.4. GRAPHICS COLOR COMPARE REGISTER (RW)

GR2				Access = 0x03CEh/0x03CFh			Regoffset = 002h
7	6	5	4	3	2	1	0
Rsv				GCCCR			
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved**, reads as ’0’.
Bits 3-0	**GCCCR**	**Graphics Controller Color Compare Register.** These bits are compared with the 4-bit color of up to 8 pixels in Read Mode 1. The 8-bit (1-bit per pixel) result of the comparison is returned to the host. (A bit of ‘1’ is returned for a match, and ‘0’ for a non-match.) Only those bits enabled by the Color Don’t Care Register (GR7) are matched.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.5. RASTER OP/ROTATE COUNT REGISTER (RW)

GR3 Access = 0x03CEh/0x03CFh Regoffset = 003h

4	3	2	1	0
GCRO		GCRC		
Default value after reset = undefined				

Bit Number	Mnemonic	Description
Bits 4-3	**GCRO**	**Graphics Controller Raster Op.** These bits define the logical operation to apply to the Host data with the data in the Graphics Controller data latch. The possible values of this field are shown in Table 11-10.
Bits 3-0	**GCRC**	**Graphics Controller Rotate Count.** These bits specify the number of bits that the Host data is rotated before the Raster Op is applied. A count of 0 passes the data through unmodified, a count of 1 rotates the Host data 1 bit to the right.

Bit 4	Bit 3	Raster Operation
0	0	NOP - Host data passes through unmodifed
0	1	Logical AND of Host and latched data
1	0	Logical OR of Host and latched data
1	1	Logical XOR of Host and latched data

Table 11-10. Raster operations

Programming notes

The contents of this register are not altered by drawing operations.

11.6.6. GRAPHICS READ MAP SELECT REGISTER (RW)

GR4 Access = 0x03CEh/0x03CFh Regoffset = 004h

7	6	5	4	3	2	1	0
Rsv						GCRMS	
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-2	**Rsv**	**Reserved**, reads as '0'.
Bits 1-0	**GCRMS**	**Graphics Controller Read Map Select.** These bits define the memory plane from which the CPU reads data in Read Mode 0. A value of '00' selects plane 0, '01' selects plane 1, etc. This field also selects one of the 4 Bytes of the Graphics Control Read Data latches.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.7. GRAPHICS MODE REGISTER (RW)

GR5 Access = 0x03CEh/0x03CFh Regoffset = 005h

7	6	5	4	3	2	1	0
Rsv	SM		OE	RM	Rsv	WM	
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	Rsv	**Reserved**, reads as '0'.
Bits 6-5	SM	**Shift mode** These values are given in Table 11-11.
Bit 4	OE	**OddEven.** This bit performs no function. It is, however, readable and writable.
Bit 3	RM	**Read Mode.** If this bit is set to '0', a host read transfer returns the data Byte corresponding to the plane selected by the Read Map Select Register (GR4). This is also called Read Mode 0. When this bit is set to '1', a host read transfer returns the result of the logical comparison between the data in the four planes selected by the Color Don't Care Register (GR7) and the contents of the Color Compare Register (GR2). This is also called Read Mode 1.
Bit 2	Rsv	**Reserved**, reads as '0'.
Bits 1-0	WM	**Write Mode.** These bits select the write mode as follows in Table 11-12.

Bit 6	Bit 5	Shift Register Behaviour
1	X	The shift registers are loaded in the manner to support 256 colors. This bit should be set to "1" for mode 13 operation.
0	1	2-bit packed pixel mode (modes 4 and 5) support. The data in the four serial shift registers are formatted as ATR0-3.
0	0	Normal shift mode. M0d7-0, M1d7-0, M2d7-0 and M3d7-0 are shifted out with address to the Attributed Controller.

Table 11-11. Shift Register Behaviour

2-bit packed pixel modes:

ATR0: M1d0 M1d2 M1d4 M1d6 M0d0 M0d2 M0d4 M0d6

ATR1: M1d1 M1d3 M1d5 M1d7 M0d1 M0d3 M0d5 M0d7

ATR2: M3d0 M3d2 M3d4 M3d6 M2d0 M2d2 M2d4 M2d6

ATR3: M3d1 M3d3 M3d5 M3d7 M2d1 M2d3 M2d5 M2d7

Bit 1	Bit 0	Write Behaviour
0	0	Write Mode 0
0	1	Write Mode 1
1	0	Write Mode 2
1	1	Write Mode 3

Table 11-12. Write Behaviour

Where:

Write Mode 0: each of the four display memory planes are written with the host data rotated by the rotate count value specified in GR3.
If the Enable Set/Reset register (GR1) enables any of the four planes, the corresponding plane is written with the data stored in the Set/Reset register (GR0). The raster operation specified in GR3 and the bit mask register (GR8) contents alter data being written.

Write Mode 1: each of the four display memory planes are written with the data from the Graphics Controller read data latches. These latches should be loaded by the host via a previous read. The Raster Operation, Rotate Count, Set/Reset Data, Enable Set/Reset and Bit Mask registers have no effect.

Write Mode 2: memory planes 3-0 are filled with the value of the host data bits 3-0, respectively. Data on the host bus is treated as the color value. The Bit Mask register (GR8) is effected in this mode. A “1” in a bit position in the Bit Mask register sets the corresponding pixel in the addressed Byte to the color specified by the host data bus. A “0” set the corresponding pixel in the addressed Byte to the corresponding pixel in the Graphics Controller read latches. The Set/Reset, Enable Set/Reset and Rotate Count register have no effect.

Write Mode 3: each of the four video memory planes is written with 8-bits of the color value contained in the Set/Reset register for that plane. The Enable Set/Reset register as no effect, all bits are enabled. The host data is rotated and ANDed with the Bit Mask register to form an 8-bit value that performs the same function as the Bit Mask register in Write Modes 0 and 2. This write mode can be used to fill an area with a single color or pattern.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.8. GRAPHICS MISCELLANEOUS REGISTER (RW)

GR6 Access = 0x03CEh/0x03CFh Regoffset = 006h

7	6	5	4	3	2	1	0
Rsv				MM		C	GM
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved,** read as ‘0’.
Bits 3-2	**MM**	**Memory Map**. These bits specify the map of the display memory buffers in the CPU address space. They are defined as follows in Table 11-13.
Bit 1	**C**	**Chain2.** This bit performs no function. It is, however, readable and writable.
Bit 0	**GM**	**Graphics Mode.** When this bit is set to ‘1’ graphics mode is selected; otherwise when set to ‘0’ alphanumeric mode is selected. This bit is duplicated in AR10[0].

Bit 3	Bit 2	Address Map
0	0	A0000h to BFFFFh (128K)
0	1	A0000h to BFFFFh (128K)
1	0	B0000h to B7FFFh (32K)
1	1	B8000h to BFFFFh (32K)

Table 11-13. Address Map

Programming notes

The contents of this register are not altered by drawing operations.

11.6.9. GRAPHICS COLOR DON'T CARE REGISTER (RW)

GR7 Access = 0x03CEh/0x03CFh Regoffset = 007h

7	6	5	4	3	2	1	0
Rsv				DCPS			
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved,** read as '0'.
Bits 3-0	**DCPS**	**Dont_care Color Plane Selects.** One bit per plane determine whether the corresponding color plane becomes a don't care when a CPU read from the video memory is done in Read Mode 1. A '1' makes the corresponding plane a don't care plane.

Programming notes

The contents of this register are not altered by drawing operations.

11.6.10. GRAPHICS BIT MASK REGISTER (RW)

GR8 Access = 0x03CEh/0x03CFh Regoffset = 008h

7	6	5	4	3	2	1	0
BM							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**BM**	**Bit Mask.** Any bit programmed to a ‘0’ inthis register will cause the corresponding bit in each of the four memory planes to be left unchanged by all operations. The data written into memory in this case will be the data which was read in the previous read operation and stored in the Graphics Controller’s read latch. The bit mask is applicable to any data written by the host. The bit mask applies to all four planes simultaneously.

Programming notes

The contents of this register are not altered by drawing operations.

11.7. ATTRIBUTE CONTROLLER REGISTERS

11.7.1. ATTRIBUTE CONTROLLER INDEX (RW)

The Attribute Controller Index register is used to index into the Attribute Data register array.

Port 3C0h is used for write access to both this index register and, in a subsequent write to this address, to the data register pointed to by the index. There is a flipflop which changes state after each write to this port. The state of the flipflop determines whether the next IO write to 3C0h will be to the index register or to a data register. The flipflop may be initialized - to point to the index register - by performing a read from Input Status Register #1 (IO address 3XAh).

ARX Access = 0x3C0h Regoffset =

7	6	5	4	3	2	1	0
Rsv		PAS	ACI				
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be written as zero.
Bit 5	**PAS**	**Palette Address Source.** When set to zero, allows host write access to the Attribute Palette Registers. The CRT display is turned off while this bit stays zero and overscan color is displayed. Setting this bit to one allows normal video pixel display and disables host write access to the Palette registers.

Bits 4-0	**ACI**	**Attribute Controller Index.** Points to the data register which will be accessed by the next write to port 3C0h or the next read from port 3C1h.
		A sample program could be as follows:
		mov DX, 3DAh
		in AL, DX
		mov AL, Index
		mov DX, 3C0h
		out DX, AL
		mov AL, Data
		out DX, AL

11.7.2. ATTRIBUTE PALETTE REGISTERS (RW)

These sixteen registers provide one level of indirection between the color data stored in the display frame buffer and the displayed color on the CRT screen. In all modes except 256 color mode, the (maximum) 4-bit raw color values select one of these sixteen Palette registers. The six bit output of the Palette registers is combined with bits 3-2 of AR14 to form the 8-bit output of the VGA controller. In addition, bits 5-4 of the VGA output may come from either Palette register bits 5-4 or from AR14 bits 1-0 depending on the state of the V54 bit (bit 7) of register AR10.

AR0-ARF Access = 0x3C1h/0x3C0h Regoffset =

7	6	5	4	3	2	1	0
Rsv		CV					
Default value after reset =							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be written as zero.
Bits 5-0	**CV**	**6-bit Color Value.**

11.7.3. ATTRIBUTE CTRL MODE REGISTER (RW)

AR10 Access = 0x3C1h/0x3C0h Regoffset =

7	6	5	4	3	2	1	0
V54	PW	PPC	Rsv	BE	LGE	MGE	GM
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**V54**	**V54 Select.** This bit determines whether bits 5-4 of the VGA pixel output come from the Video54 field of AR14 (bits 1-0) or from the normal output of the VGA Palette registers. Setting this bit to one selects the Video54 field.
Bit 6	**PW**	**Pixel Width.** When this bit is set to one, pixels are clocked at half the normal rate. The effect is to double the width of pixels displayed on the CRT.
Bit 5	**PPC**	**Pixel Panning Compatibility.** When VGA split screen is in effect, this bit controls whether both screens or just the top one are affected by Pixel and Byte Panning fields. When set to zero, both screens pan together.
Bit 4	**Rsv**	**Reserved.**
Bit 3	**BE**	**Blink Enable.** Setting this to one enables blinking in both text and graphics modes. When this bit is set to one in text mode, character attribute bit 7 is used on a character by character basis to enable or disable blinking. When this bit is set to zero in text mode, character attribute bit 7 controls character intensity. The blinking rate is equal to the vertical retrace rate divided by 32 (about twice per second). Setting this bit to one in graphics modes causes the VGA palette input bit 3 to toggle (approx twice per second) if the incoming pixel bit 3 is high.
Bit 2	**LGE**	**Line Graphics Enable.** Setting this bit to one forces the ninth pixel of a line graphics character (ascii codes C0h through DFh) to be the same as the eighth pixel. Setting it to zero forces the ninth pixel to be displayed as the background color. Ninth pixels of all other ascii codes are always displayed as background color. This bit has no meaning when character width is not set to nine or during graphics modes.
Bit 1	**MGE**	**Mono Attributes Enable.** Setting this bit to one in graphics modes while Bit 3 of this register is also one causes the VGA palette input bit 3 to toggle regardless of the incoming pixel's bit 3.
Bit 0	**GM**	**Graphics Mode.** Set this bit to one for graphics mode, zero for text mode.

11.7.4. ATTRIBUTE CTRL OVERSCAN COLOR REGISTER (RW)

AR11 Access = 0x3C1h/0x3C0h Regoffset =

7	6	5	4	3	2	1	0
BC							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	**BC**	**Border Color.** These bits define the color of the CRT border if there is one. The border or overscan region is that part of the display between where active pixels are displayed and those where the blank signal is active.

11.7.5. ATTRIBUTE COLOR PLANE ENABLE REGISTER (RW)

AR12 Access = 0x3C1h/0x3C0h Regoffset =

7	6	5	4	3	2	1	0
Rsv		VSMC		CPE			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** These bits should be written as zero.
Bits 5-4	**VSMC**	**Video Status Mux Control.** These bits select two of the eight output bits of the Attribute Controller to be read via Input Status Register #1 (3XAh) bits 5-4. The selection is as follows in Table 11-14.
Bits 3-0	**CPE**	**Color Plane Enable.** These four bits are ANDed with the frame buffer data before being input into the Palette. If any of these bits are zero, the corresponding plane from the frame buffer will be masked out of the Palette look up.

AR12[5]	AR12[4]	ISR[5]	ISR[4]
0	0	PD[2]	PD[0]
0	1	PD[5]	PD[4]
1	0	PD[3]	PD[1]
1	1	PD[7]	PD[6]

Table 11-14. Video Status Mux Control

11.7.6. ATTRIBUTE HORZ PIXEL PANNING REGISTER (RW)

AR13 Access = 0x3C1h/0x3C0h Regoffset =

7	6	5	4	3	2	1	0
Rsv				HPP			
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved.** These bits should be written as zero.
Bits 3-0	**HPP**	**Horizontal Pixel Panning.** These bits specify the number of pixels by which to shift the display left (see table below: Table 11-15.).

H Pixel Pan	**Shift**		
	9 pixels/chr	**8 pixels/chr**	**mode 13**
0	1 pixel left	0 pixels	0 pixels
1	2 pixels left	1 pixel left	0 pixels
2	3 pixels left	2 pixels left	1 pixel left
3	4 pixels left	3 pixels left	1 pixel left
4	5 pixels left	4 pixels left	2 pixels left
5	6 pixels left	5 pixels left	2 pixels left
6	7 pixels left	6 pixels left	3 pixels left
7	8 pixels left	7 pixels left	3 pixels left
8-15	0 pixels	undefined	undefined

Table 11-15. Horizontal Pixel Panning

11.7.7. ATTRIBUT COLOR SELECT REGISTER (RW)

AR14 Access = 0x3C1h/0x3C0h Regoffset =

7	6	5	4	3	2	1	0
Rsv				V76		V54	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-4	**Rsv**	**Reserved.** These bits should be written as zero.
Bits 3-2	**V76**	**Video76.** In all modes except 256 color mode (mode 13), these bits are output onto bits 7-6 of the VGA pixel data output port.
Bits 1-0	**V54**	**Video54.** When bit 7 of AR10 is set to '1', these bits are output onto bits 5-4 of the VGA pixel data output port.

11.8. CRT CONTROLLER REGISTERS

The STPC implements an extension of the VGA CRTC controller. The CRTC controller supports up to 1024x768 display resolutions at 75HZ refresh rates as defined by VESA Monitor Timing Standard. The horizontal timing control fields are all VGA compatible.

The vertical timings are extended by 1-bit to accommodate above display resolution. The address registers are extended to allow locating the frame buffer in anywhere within the first 4Mb of physical main memory.

11.8.1. INDEX REGISTER (RW)

The CRTC Index register points to an internal register of the CRT controller. The seven least significant bits determine which register will be pointed to in the next register read/write operation to IO port 3B5/3D5.

CRX Access = 0x3X4h/0x3X5h Regoffset =

7	6	5	4	3	2	1	0
Rsv	CRTC I						
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** Must be written to ‘0’s. Read back is undefined.
Bits 6-0	**CRTC I**	**CRTC Index.** Points to the CRTC register that will be accessed by an IO cycle at 03B5h/03D5h.

11.8.2. HORIZONTAL TOTAL REGISTER (RW)

The horizontal total register defines the total number of characters in a horizontal scan line, including the retrace time. The characters displayed on the screen are counted by a character counter. A count of 0 corresponds to the first displayed character at the left side of the screen. The value of the character counter is compared with the value in this register to provide the horizontal timing. A character is composed of 8 or 9 pixels as defined in Sequencer clocking mode register. All horizontal and vertical timing is based on the contents of this register.

The maximum horizontal resolution possible with this field is approximately 260*8*0.8 = 1664. (260 is 255+5, 0.8 is the fraction of a horizontal scan period during which active pixels are displayed).

CR0 Access = 0x3X4h/0x3X5h Regoffset = 000h

7	6	5	4	3	2	1	0
Default value after reset = 00h							

Programming notes

The 8-bit value in this register = Total number of characters - 5.

11.8.3. HORIZ DISPLAY END REGISTER (RW)

This 8-bit read/write register defines the total number of displayed characters in a scan line.

CR1 Access = 0x3X4h/0x3X5h Regoffset = 001h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

Programming notes

The 8-bit value in this register = Total number of displayed characters - 1.

11.8.4. HORIZ BLANCKING START REGISTER (RW)

This 8-bit read/write register defines when the horizontal blanking will start. The horizontal blanking signal becomes active when the horizontal character count equals the contents of this register.

CR2 Access = 0x3X4h/0x3X5h Regoffset = 002h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

11.8.5. HORIZ BLANCKING END REGISTER (RW)

CR3 Access = 0x3X4h/0x3X5h Regoffset = 003h

7	6	5	4	3	2	1	0
Rsv	DESC		HBEV				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** This readable and writable bit must be written to as ‘1’ to ensure proper VGA operation. It resets to one.
Bits 6-5	**DESC**	**Display Enable Skew Control.** These bits delay the display enable by the specified number of character clocks. The result is that the video output stream is delayed by the same amount resulting in wider left border and shrunk right border. This field is in unknown state after reset.
Bits 4-0	**HBEV**	**Horizontal Blanking End Value Bits 4-0.** These bits specify the least significant 5-bits of the 6-bit wide Horizontal Blanking End value. The sixth bit is located in CRTC Horizontal Retrace End register. This field is in unknown state after reset.

Programming notes

This field controls the width of the horizontal blanking signal as follows:
Horizontal Blanking start register + width of the blanking signal = 6-bit Horizontal blanking end value.

The blanking signal set in CR2 and CR3 should start at least 23 GCLKs prior to the start of video window.

11.8.6. HORIZ RETRACE START REGISTER (RW)

This 8-bit register defines the character position at which the horizontal sync becomes active.

CR4 Access = 0x3X4h/0x3X5h Regoffset = 004h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

11.8.7. HORIZONTAL RETRACE END REGISTER (RW)

CR5 Access = 0x3X4h/0x3X5h Regoffset = 005h

7	6	5	4	3	2	1	0
HBEV	HRSC		HRWV				
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**HBEV**	**Horizontal Blanking End Value Bit 6.** This is the sixth bit of the Horizontal Blanking end field. Refer to CRTC Horizontal Blanking end register for more details.
Bits 6-5	**HRSC**	**Horizontal Retrace Skew Control.** This field delays the start of the horizontal sync by the specified number of character clocks. For text mode operation, this field should be programmed to ‘1’.
Bits 4-0	**HRWV**	**Horizontal Retrace Width Value.** These 5-bits specify the width of the horizontal sync signal as follows: Horizontal Retrace Start register + width of the horizontal sync = 5-bit Horizontal retrace end value.

11.8.8. VERTICAL TOTAL REGISTER (RW)

This register contains the least significant 8-bits of the 11-bit wide Vertical Total value. Next most significant 2-bits are located in CRTC overflow register CR7 and the 11th bit is located in Repaint Control Register 4.

CR6 Access = 0x3X4h/0x3X5h Regoffset = 006h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

Programming notes

The value programmed in this register = Total number of scan lines - 2.

11.8.9. OVERFLOW REGISTER (RW)

CR7 Access = 0x3X4h/0x3X5h Regoffset = 007h

7	6	5	4	3	2	1	0
VR	VD	VT	L	VB	VR	VD	VT
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**VR**	**Bit-9 of the 11-bit wide Vertical Retrace start register.**
Bit 6	**VD**	**Bit-9 of the 11-bit wide Vertical Display end register.**
Bit 5	**VT**	**Bit-9 of the 11-bit wide Vertical Total register.**
Bit 4	**L**	**Bit-8 of the 11-bit wide Line compare register.**
Bit 3	**VB**	**Bit-8 of the 11-bit wide Vertical Blanking start register.**
Bit 2	**VR**	**Bit-8 of the 11-bit wide Vertical Retrace start register.**
Bit 1	**VD**	**Bit-8 of the 11-bit wide Vertical Display end register.**
Bit 0	**VT**	**Bit-8 of the 11-bit wide Vertical Total register.**

11.8.10. SCREEN A PRESET ROW SCAN REGISTER (RW)

CR8 Access = 0x3X4h/0x3X5h Regoffset = 008h

7	6	5	4	3	2	1	0
Rsv	DS		SS				
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved**. Must be written to a '0'. Read back is undefined.
Bits 6-5	**DS**	**Display Shift**. This field is added to the memory address generated during the display. As a result, the display shifts left by one, two or three Bytes. For both alphanumeric and graphics modes, this implies a left shift by 8, 16 or 24 pixels respectively. This field is encoded as follows in Table 11-16. When the line compare condition becomes true and pixel panning compatibility bit (AR10 bit 5) is a '1', the outputs of bits 5 and 6 are forced '0' until the start of the next vertical sync pulse.
Bits 4-0	**SS**	**Smooth Scroll.** This field can be used to implement smooth vertical scrolling. It specifies the starting row scan count of the character cell after a vertical retrace (assuming the scan lines of a character row are numbered starting with 0). Smooth vertical scrolling can be implemented by setting this register to a value between 1 and the value in CR9. As a result, after a vertical retrace, the display will start from the scan line specified in this field instead of 0. This field is effective only for the top half of the screen (Screen A) if split screen mode is in effect. Each horizontal scan increments the horizontal row scan counter and is reset to 0 when it reaches the character cell height value programmed in CR9. If this field is programmed to a value larger than the character cell height, the row scan counter will count up to 1Fh before rolling over. A '0' in this field means no scrolling.

Bit 6	Bit 5	Byte Panning
0	0	0 Byte (display shifts 0 pixels left)
0	1	1 Byte (display shifts 8 pixels left)
1	0	2 Bytes (display shifts 16 pixels left)
1	1	3 Bytes (display shifts 24 pixels left)

Table 11-16. Byte Panning

11.8.11. CHARACTER CELL HEIGHT REGISTER (RW)

CR9 Access = 0x3X4h/0x3X5h Regoffset = 009h

7	6	5	4	3	2	1	0
SC	LC	VB	SLPR				
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	SC	**Scan Double.** When set to a ‘1’, this bit allows a 200-line mode to be displayed on 400 display scan lines by dividing the row scan counter clock by 2 to duplicate each scan line. Thus all row scan address counter based timing (including character height and cursor and underline locations) double, as measured in scan lines, when scan doubling is enabled. Scan doubling only effects the way in which data is displayed; it does not effect display timing. If this bit is set without changing anything else, data currently displayed will appear twice as tall; horizontal and vertical sync, blanking etc., will remain the same.
Bit 6	LC	**Bit-9 of the 11 bit wide Line Compare field.**
Bit 5	VB	**Bit-9 of the 11 bit wide Vertical Blank Start field.**
Bits 4-0	SLPR	**Scan Lines Per Row.** This field specifies the number of scan lines per character row minus one.

11.8.12. CURSOR START REGISTER (RW)

CRA Access = 0x3X4h/0x3X5h Regoffset = 00Ah

7	6	5	4	3	2	1	0
Rsv		CDE	CSSL				
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be written to ‘0’s. Read back is undefined.
Bit 5	**CDE**	**Cursor Display Enable.** When set to a ‘0’, this bit enables displaying the cursor. Cursor is displayed only in alphanumeric mode. In graphics mode, the cursor is always disabled and this bit has no effect.
Bits 4-0	**CSSL**	**Cursor Start Scan Line.** This field, in conjunction with the Cursor End scan line, defines the shape of the cursor. The hardware cursor is represented as a block of pixels occupying a character position. This field determines the first scan line within the character box that should be filled in (the first scan line is numbered as 0). If the cursor start and end scan line numbers are the same, one scan line wide cursor will be displayed. If starting scan line number is larger than the end, no cursor will be displayed. This is illustrated in Cursor start and end registers Figure 11-1.

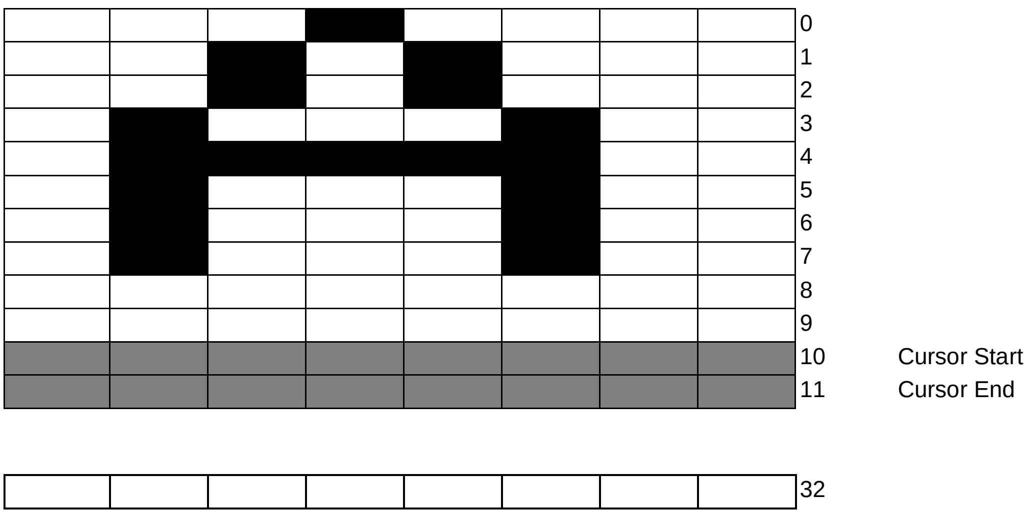


Figure 11-1. Cursor start and end registers

11.8.13. CURSOR END REGISTER (RW)

CRB Access = 0x3X4h/0x3X5h Regoffset = 00Bh

7	6	5	4	3	2	1	0
Rsv	CSC		CESL				
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** Must be written to ‘0’. Read back is undefined.
Bits 6-5	**CSC**	**Cursor Skew Control.** This field skews the cursor location (defined by the cursor location register) by the specified number of character clocks to the right.
Bit 5	**CESL**	**Cursor End Scan Line.** This field, in conjunction with the Cursor Start Scan line field defines the cursor shape. This is illustrated in Figure 11-1.

11.8.14. START ADDRESS HIGH REGISTER (RW)

This 8-bit register specifies bits 15-8 of the 20-bit display buffer address which will be displayed on the screen after a vertical retrace. Register CRD contains the lower 8-bits and the CRTC extended register CR19 contains the upper 4 bits.

If split screen mode is in effect, this address is the start address of the first of the two (the top one) screens (Screen A). The start address of Screen B (the bottom one) is always 0. The starting scan line for the Screen B is determined by the line compare register (CR18).

CRC Access = 0x3X4h/0x3X5h Regoffset = 00Ch

7	6	5	4	3	2	1	0
Default value after reset =							

11.8.15. START ADDRESS LOW REGISTER (RW)

This 8-bit register specifies bits 7-0 of the 20-bit display buffer address which will be displayed on the screen after a vertical retrace. Register CRC contains bits 15-8 and the CRTC extended register CR19 contains the upper 4 bits.

The start address in the CRC, CRD and CRIG does not define the offset within the frame buffer of the 1st displayed pixel but this value divided by 4.

CRD Access = 0x3X4h/0x3X5h Regoffset = 00Dh

7	6	5	4	3	2	1	0
Default value after reset =							

11.8.16. TEXT CURSOR OFFSET HIGH REGISTER (RW)

CRE Access = 0x3X4h/0x3X5h Regoffset = 00Eh

7	6	5	4	3	2	1	0
CO							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CO**	**Cursor offset bits 15-8.** This field contains the upper half of the 16-bit cursor offset. The offset is relative to the left-most character on the top of the screen and is specified in terms of character positions. For example, an offset of 0 will place the cursor on the left-most character on the top. An offset of 2 will place the cursor at the third character from the left in the top most row and so on. Since the information is stored in the display memory as character-attribute pairs, the address of the character under the cursor will be exactly twice the cursor offset + the screen base address.

11.8.17. TEXT CURSOR OFFSET LOW REGISTER (RW)

This register is the VGA compatible Cursor Offset Low register.

CRF Access = 0x3X4h/0x3X5h Regoffset = 00Fh

7	6	5	4	3	2	1	0
CO							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CO**	**Cursor offset bits 7-0.** This is the lower half of the cursor offset.

11.8.18. VERTICAL RETRACE START REGISTER (RW)

This register contains the lower 8 bits of the 11-bit wide vertical retrace start value. Register CR7 contains bits 8 and 9. Repaint Control Register 4 contains the msb of this 11-bit field. The retrace value is specified in horizontal scan lines where top most scan line on the screen is line 0.

CR10 Access = 0x3X4h/0x3X5h Regoffset = 010h

7	6	5	4	3	2	1	0
Default value after reset =							

11.8.19. VERTICAL RETRACE END REGISTER RW)

CR11 Access = 0x3X4h/0x3X5h Regoffset = 011h

7	6	5	4	3	2	1	0
CR P	Rsv	VGA IE	VGA IR	VRW			
Default value after reset = 0x10xxxx							

Bit Number	Mnemonic	Description
Bit 7	**CR P**	**CR Protect.** This bit when set to a ‘1’, write protects CR0-7 registers except CR7 bit 4.
Bit 6	**Rsv**	**Reserved.** This bit is both readable and writeable.
Bit 5	**VGA IE**	**VGA Interrupt Enable.** When set to a ‘0’, this bit enables the interrupt assertion of the VGA core. Setting this bit to a ‘1’ disables the interrupts.
Bit 4	**VGA IR**	**VGA Interrupt Reset.** Setting this bit to a ‘0’, clears the vertical retrace interrupt flip-flop and deasserts the interrupt output (if it was asserted). Setting this bit back to ‘1’ enables the interrupt flip-flop to record the next vertical retrace. The interrupt flip-flop, if enabled by this bit, is set to one scan line after vertical blank is asserted. The flip-flop will not be set and the vertical retrace interrupt will be lost if this bit is set to a ‘0’ when the interrupt occurred. The vertical interrupt flip-flop can be read as bit 7 of Input status register #0.
Bits 3-0	**VRW**	Bits 3-0 **Vertical Retrace Width.** These bits determines the width of the vertical retrace output as follows: Value in the Vertical Retrace Start register (CR10) + Width of the vertical retrace pulse = 4-bit value to be programmed into this field.

11.8.20. VERTICAL DISPLAY END REGISTER (RW)

This register contains the lower 8-bits of the 11-bit wide Vertical display end value which specifies the scan line position where the display on the screen ends. Bits 8 and 9 are specified in CRTC overflow register and the bit-10 in the Repaint Control Register 4.

CR12 Access = 0x3X4h/0x3X5h Regoffset = 012h

7	6	5	4	3	2	1	0
Default value after reset =							

Programming notes

The value in this register = Total number of displayed scan lines - 1.

11.8.21. OFFSET REGISTER (RW)

This register defines bits 7-0 of the 10-bit wide logical width of the line displayed on the screen. Extended register CR19 contains the upper two bits. The first scan line displayed on the screen, starts at the address specified in registers CRC, CRD and extended register CR19. The starting address of the next scan line is computed as the Byte starting address of the current row + 2*offset.

CR13 Access = 0x3X4h/0x3X5h Regoffset = 013h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

11.8.22. UNDERLINE LOCATION REGISTER (RW)

CR14 Access = 0x3X4h/0x3X5h Regoffset = 014h

7	6	5	4	3	2	1	0
Rsv	DWM	C 4	UL				UE
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** Must be written to a ‘0’. Read back is undefined.
Bit 6	**DWM**	**Double Word Mode.** If this bit is set to a ‘1’, the address generated by the CRTC memory address counter is shifted up two bits to provide the frame buffer address and bits 1-0 of the frame buffer address are driven from CRTC memory address counter bits 13 and 12 respectively. The logical screen width is multiplied by 8 and added to the starting address of the current scan line to compute the starting address of the next scan line.
Bit 5	**C 4**	**Count by 4.** Setting this bit to one causes the memory address counter to increment every four character clocks.
Bits 4-1	**UL**	**Underline Location.** This field specifies the horizontal row scan of the character cell at which the underlining will occur assuming that top line of the character cell is numbered 0. Underlining occurs in text (alphanumeric) modes only when an attribute value of ‘b000i001’ binary is detected (where b indicates blink and i indicates intensified).
Bit 0	**UE**	**Underline Enable.** Setting this bit to ‘1’ enables underlining.

Programming notes

Underlining is enabled only in alphanumeric mode and then it is meaningful only for monochrome display (mode 7). For color modes the bit 0 of the attribute Byte is interpreted as foreground color. There is no explicit bit to disable underlining for color alphanumeric modes. Instead, it is disabled by programming this field to a value larger that the character cell height programmed in CR9.

11.8.23. VERTICAL BLANCKING START REGISTER (RW)

This register contains the lower 8-bits of the 11-bit scan line valued where the vertical blanking is to begin. The 9th and 10th bits are located in CR7 and CR9 and the 11th bit is located in Repaint Control Register 4.

CR15 Access = 0x3X4h/0x3X5h Regoffset = 015h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

11.8.24. VERTICAL BLANCKING END REGISTER

This 8-bit register defines the width of the vertical blanking pulse as follows:

CR16 Access = 0x3X4h/0x3X5h Regoffset = 016h

7	6	5	4	3	2	1	0
Default value after reset =undefined							

Programming notes

Start Vertical Blank value + width of the blanking pulse = Value programmed in this register.

While the register is 8-bits wide, and all bits are readable/writeable, only the least significant 7 bits are used in the generation of the vertical pulse.

11.8.25. MODE REGISTER (RW)

CR17 Access = 0x3X4h/0x3X5h Regoffset = 017h

7	6	5	4	3	2	1	0
H V RE	B WM	VGA MAS	Rsv	C 2	DVT	MS	MS
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**H V RE**	**H/V Retrace Enable.** A '0' in this bit position disables the horizontal and vertical retraces and a '1' enables them.
Bit 6	**B WM**	**Byte/Word# Mode**. A '1' value in this bit position selects the Byte mode and a '0' selects the word mode. Following Table 11-17. lists the memory address generation for Byte, word and double-word addressing modes. iA24-0 refer to the output of the internal memory address counter and the memory address is the address presented to the address lines of a 4-Byte wide display buffer memory, that is, each memory address selects 4-Bytes.
Bit 5	**VGA MAS**	**VGA Memory Address Size.** When set to a '0', this bit, in Word addressing mode (see above) propagates bit iA13 on the least significant memory address bit. If set to a '1', it propagates bit iA15 instead. iA13 should be used if total display buffer memory is 64K and iA15 should be used if total memory is 256K. It is expected that the VGA Controller will always be used with 256K or larger memory. Therefore this bit should be programmed to a '1'.
Bit 4	**Rsv**	***Reserved***. Must be written to a '0'. Read back is undefined.
Bit 3	**C 2**	**Count by 2.** Setting this bit to one causes the memory address counter to increment every second character clock.
Bit 2	**DVT**	**Double Vertical Total.** This bit when set to '1', causes all the vertical timing counters to operate at half the horizontal retrace rate. The result is that the vertical resolution doubles. The Vertical Total, Vertical Retrace Start, Vertical Display End, Vertical Blanking Start and Line Compare registers can be programmed at half their normal value if this bit is set to a '1'. The vertical timing counters operate at their normal frequency if this bit is set to a '0'.
Bit 1	**MS**	**Memory Segmentation Bit 14.** If set to a '0', the row scan counter bit 1 is substituted for memory address bit 14 during display refresh. This has the affect of segmenting the address space such that every other scan line pair is 8K apart. In combination with bit 0, this bit can segment the address space in 4-banks. No such substitution takes place if this bit is set to a '1'.
Bit 0	**MS**	**Memory Segmentation Bit 13.** This bit is similar to Bit 1 above in that when set to a '0', row scan counter bit 0 is substituted for display memory address bit 13 during active display time. No such substitution takes place if this bit is set to a '1'.

Internal Memory Address counter	Byte Mode	Word Mode	Double Word Mode
iA24	iA24	iA23	iA22
iA23	iA23	iA22	iA21
:	:	:	:
:	:	:	:
iA3	iA3	iA2	iA1
iA2	iA2	iA1	iA0
iA1	iA1	iA0	iA13
iA0	iA0	iA13/iA15	iA12

Table 11-17. Memory Address Generation

The least significant memory address bit in Word mode is selected between iA13 and iA15 based on bit 5 of this register.

This bit is ignored if Double-word mode bit in CR14 is set to a ‘1’.

11.8.26. LINE COMPARE REGISTER (RW)

This register contains the lower 8 bits of the 10-bit wide Line Compare field. The 9th and 10th bits of this field are held in CR7 and CR9 registers respectively. When the horizontal scan line counter value is equal to the contents of the Line Compare register, the memory address generator and the character row scan count are cleared. As a result the display is split into the two halves. The top half, Screen A displays the contents of the display buffer starting from Start address (CRC and CRD registers) while the bottom half, Screen B, displays the contents of the display buffer starting from address 0.

CR18 Access = 0x3X4h/0x3X5h Regoffset = 018h

7	6	5	4	3	2	1	0
Default value after reset = undefined							

Programming Notes

Screen A can be smooth scrolled vertically but Screen B can not. Control is provided via bit 5 of AR10 register to allow Screen B to pan horizontally with Screen A or not.

Split screen function can be disabled by programming the Line compare field to a value larger, typically 3FFh, than the Vertical Total field. This field must be programmed to 3FFh for optimal system performance in native display modes.

11.8.27. GRAPHICS CONTROL DATA (R)

CR22 Access = 0x3X4h/0x3X5h Regoffset = 019h

7	6	5	4	3	2	1	0
GCDL N							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**GCDL N**	**Graphics Controller Data Latch N.** These bits, when read, provide the state of one of the 4 Graphics Controller's Data Latches. The Graphics Controller Read Map Select register (GR4) specify which latch is read.

Programming notes

The contents of this register are not altered by drawing operations.

11.8.28. ATTRIBUTE ADDRESS FLIPFLOP (R)

CR24 Access = 0x3X4h/0x3X5h Regoffset = 020h

7	6	5	4	3	2	1	0
AF	Rsv						
Default value after reset =							

Bit Number	Mnemonic	Description
Bit 7	**AF**	**Attribute Flipflop.** This read-only bit indicates the state of the Attribute Controller index flipflop. When this bit is zero, the next access to IO port 3C0h will be to the Attribute Index register. When this bit is one, the next access will be to an Attribute data register.
Bits 6-0	**Rsv**	**Reserved**. Read as zero.

11.8.29. ATTRIBUTE INDEX READBACK (R)

CR26 Access = 0x3X4h/0x3X5h Regoffset = 021h

7	6	5	4	3	2	1	0
Rsv		PAS	ACI				
Default value after reset =							

Bit Number	Mnemonic	Description
Bits 7-6	Rsv	**Reserved.** Read as zero.
Bit 5	PAS	**Palette Address Source.** This is a read-only copy of Attribute Controller Index register (ARX) bit 5.
Bits 4-0	ACI	**Attribute Controller Index.** This is a read-only copy of bits 4-0 of the Attribute Controller Index register.

11.9. VGA EXTENDED REGISTERS

The following registers are additions to those found in the standard VGA specification. They can only be accessed after register SR6 has been written to with 57h.

A typical sequence in 80X86 assembly could be:

```
max     DX, 3C4h
mov     OX, 5706h
out     DX, AX
```

11.9.1. REPAINT CONTROL REGISTER 0 (RW)

CR19 Access = 0x3X4h/0x3X5h Regoffset = 019h

7	6	5	4	3	2	1	0
Rsv		CRTC O		CRTC SAF			
Default value after reset =00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**. Must be written to ‘0’.
Bits 5-4	**CRTC O**	**CRTC Offset register Bits 9-8.** See CRTC register 13 for details.
Bits 3-0	**CRTC SAF**	**CRTC Start Address Field Bits 19-16.** See CRTC register C, D for explanation of the Start Address.

11.9.2. REPAINT CONTROL REGISTER 1 (RW)

CR1A			Access = 0x3X4h/0x3X5h				Regoffset = 01Ah
7	6	5	4	3	2	1	0
HTD	VTD	Rsv	CTM	Rsv	LCE	SBP	VGA AW
Default value after reset = 3Fh							

Bit Number	Mnemonic	Description
Bit 7	**HTD**	**Hsync Toggle Disable.** When set to a ‘1’, this bit forces the Hsync to inactive state (high or low as programmed in the bit 6 of the Miscellaneous output register). See note below.
Bit 6	**VTD**	**Vsync Toggle Disable.** When set to a ‘1’, this bit forces the Vsync to inactive state (high or low as programmed in the bit 7 of the Miscellaneous output register). See note below.
Bit 5	**Rsv**	***Reserved**.* This bit always reads as a one.
Bit 4	**CTM**	**Compatible Text Mode.** When this bit is set to one, the CRT controller expects the font data format within the frame buffer to be identical to that used by the standard VGA chip. Setting this bit to zero enables “Enhanced Text Mode” as described under CR1C bit 7. Note, though that this bit has the opposite sense of CR1C bit 7.
Bit 3	**Rsv**	**Reserved.** This bit always reads as a one.
Bit 2	**LCE**	**Line Compare Enable.** Set this bit to zero for 1280x1024 mode and one for all others.
Bit 1	**SBP**	**Six Bit Palette.** Set this bit to one to enable VGA compatible 6 bit palette functionality and zero to enable the 8 bit palette.
Bit 0	**VGA AW**	**VGA Address Wrap.** Setting this bit to one enables VGA compatible address wrapping such that the CRTC will only address 256kb of frame buffer memory (address bits 16 and above are zeroed). Set this bit to zero for SVGA modes. This bit has no effect on CPU reads or writes to the frame buffer - only CRTC accesses.

Programming notes

Note: Vertical / Horizontal Synch not toggling is defined by the VESA specification for Monitor Power Down State.

11.9.3. REPAINT CONTROL REGISTER 2 (RW)

CR1B Access = 0x3X4h/0x3X5h Regoffset = 01Bh

7	6	5	4	3	2	1	0
FIFO LWM					Rsv	V FIFO U	W FIFO U
Default value after reset = 20h							

Bit Number	Mnemonic	Description
Bits 7-3	**FIFO LWM**	**FIFO Low Water Mark.** When the FIFO occupancy falls below twice this value, the CRTC will restart frame buffer read cycles to refill the FIFO. This field should only ever be set by the BIOS during mode switches. Setting this field too small results in random pixels being displayed to the screen; setting it too large results in decreased CPU - DRAM bandwidth. Also, do not set this register to a value greater than that programmed into the high water mark (register CR27).
Bit 2	**Rsv**	**Reserved.**This bit is not writable. It reads as zeroes.
Bit 1	**V FIFO U**	**Video FIFO Underflow.** This read-only bit is set to one when the video refresh FIFO underflows. As with bit 0, writes to this register clear this bit to zero.
Bit 0	**W FIFO U**	**Warning: FIFO Underflow.** This read-only bit is set to one when the CRTC refresh FIFO nderflows (the memory subsystem did not keep up with pixel requests. Sampling this bit as a one means that a serious problem exists and the low water mark above should be incremented. Writes to this register (presumably with a larger low water mark value) reset this bit to zero.

11.9.4. REPAINT CONTROL REGISTER 3 (RW)

CR1C			Access = 0x3X4h/0x3X5h				Regoffset = 01Ch
7	6	5	4	3	2	1	0
ETFL	Rsv		Rsv		SC	PSC	EOP
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	ETFL	**Enhanced Text Font Load.** This bit should be set to one prior to loading fonts for 132 column high speed text mode. It warps the frame buffer addresses such that what appear to be standard text font writes actually get stored into frame buffer plane 2 in a more optimized manner. Specifically, frame buffer address bits 15-5 are swapped down to become bits 10-0 and bits 4-0 are moved up to become bits 15-11. The sequence of events to load 132 column enhanced text fonts is as follows. First, Odd-Even and Chain 4 modes should be turned off, then this bit should be set to one. Fonts should then be loaded in the normal VGA manner and finally this bit should be reset to zero and Text/Odd-Even mode entered. Note that the corresponding address warp for the CRT Controller is performed during font table look-ups when bit 4 of CR1A is set to zero.
Bits 6-5	Rsv	**Reserved,** read as ‘0’s.
Bits 4-3	Rsv	**Reserved.** These bits read as ‘0’s.
Bit 2	SC	**Sequential Chain-4.** When this bit is set to ‘1’, allows the display buffer memory to appear as a normal memory with a Byte address in the host address space mapping into a Byte address in the display buffer address space. Chain-4 in SR4 must be set for Sequential Chain-4 to work
Bit 1	PSC	**Page Select Control.** This bit provides control over whether the cycle type (read or write) or the upper address bit controls the page selection. The VGA implements two 7-bit page registers, Page 0 and Page 1, to allow mapping the VGA address space anywhere in the 4 MB address space. If this bit is ‘1’, the page selection is based on bit 15 or 16 of host address and the Memory Map bits of GR6 as follows in Table 11-18.
Bit 0	EOP	**Enable Overlapped Paging.** This bit should be turned on to solve the broken line problem. When software wants to draw a line that crosses the current page boundary it turns this bit on to form a page out of half of the current page and half of the next page. Since the hardware adds half page to the address when this bit is on, the software should subtract half page for passing on the address. When this bit is ‘1’, the memory address bits MA17 and MA18 are changed as follows for Normal, Odd/Even and Chain-4 cases, see as follows in Table 11-19. For Sequential Chain-4, MA16 and MA15 are changed as follows in Table 11-20.

GR6			Page Selection
Bit3	Bit2	Size	
0	0	128K	HA16=0 → Page 0; HA16=1 → Page 1
0	1	64K	HA15=0 → Page 0; HA15=1 → Page 1
1	0	32K	Not allowed
1	1	32K	Not allowed

Table 11-18.

GR6			Added to MA18	Added to MA17
Bit3	Bit2	Size		
0	0	128K	1 (256K added)	0
0	1	64K	0	1 (128K added)
1	0	32K	Not allowed	
1	1	32K	Not allowed	

Table 11-19.

GR6			Added to MA16	Added to MA15
Bit3	Bit2	Size		
0	0	128K	1 (64K added)	0
0	1	64K	0	1 (32K added)
1	0	32K	Not allowed	
1	1	32K	Not allowed	

Table 11-20.

Programming notes

The contents of this register are not altered by drawing operations.

11.9.5. PAGE REGISTER 0 (RW)

CR1D Access = 0x3X4h/0x3X5h Regoffset = 01Dh

7	6	5	4	3	2	1	0
Rsv	P 0						
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved,** read as '0'.
Bits 6-0	**P 0**	**Page 0 Bits 6-0.** 7-bit Page register 0 is used to extend the host address to allow the VGA buffer to be located anywhere in the 4 MB frame buffer space. The pages are located on 32K boundaries for Normal, Odd/Even and Chain-4 modes and on 8K boundaries for Sequential Chain-4 mode. This is illustrated in Figure 11-2.

Programming notes

Register = A000h mapped to the frame buffer and can be either 8 KBytes or 32 KBytes.

The contents of this register are not altered by drawing operations.

11.9.6. PAGE REGISTER 1 (RW)

CR1E Access = 0x3X4h/0x3X5h Regoffset = 01Eh

7	6	5	4	3	2	1	0
Rsv	P 1						
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved,** read as ‘0’.
Bits 6-0	**P 1**	**Page 1 Bits 6-0**. 7-bit Page register 1 is used to extend the host address to allow the VGA to be located anywhere in the 4 MB frame buffer space. The pages are located on 32K boundaries for Normal, Odd/Even and Chain-4 modes and on 8K boundaries for Sequential Chain-4 mode. This is illustrated in Figure 11-2.

Programming notes

Register = A800h mapped to the frame buffer and can be either 8 KBytes or 32 KBytes.

The contents of this register are not altered by drawing operations.

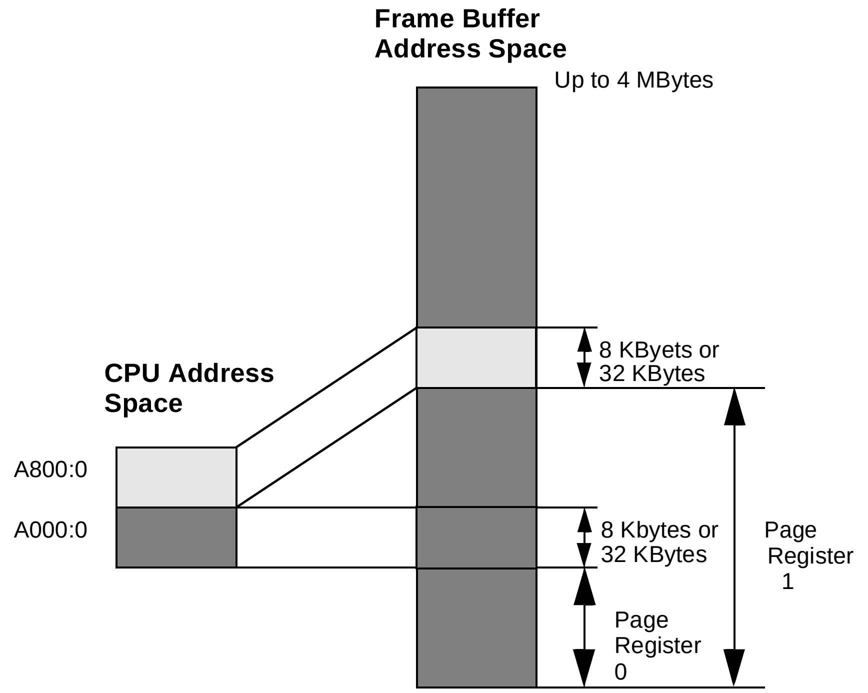


Figure 11-2. **Illustration of Page Register 0 and Page Register 1**

11.9.7. GRAPHICS EXTENDED ENABLE REGISTER (RW)

CR1F Access = 0x3X4h/0x3X5h Regoffset = 01Fh

7	6	5	4	3	2	1	0
E	Rsv						
Default value after reset =00h							

Bit Number	Mnemonic	Description
Bit 7	**E**	**Exen.** Writing a ‘1’ in this bit enables the GE extended functionality and also direct access to the frame buffer as defined by GBASE (CR20). Writing a ‘0’ disables it. After reset, this bit is set to ‘0’.
Bits 6-0	**Rsv**	**Reserved,** read as ‘0’.

Programming notes

The contents of this register are not altered by drawing operations.

11.9.8. GRAPHICS EXTENDED GBASE REGISTER (RW)

CR20 Access = 0x3X4h/0x3X5h Regoffset = 020h

7	6	5	4	3	2	1	0
Rsv					G		
Default value after reset =00h							

Bit Number	Mnemonic	Description
Bits 7-3	**Rsv**	**Reserved,** these bits read as ‘0’.
Bits 2-0	**G**	**Gbase.** This range defines the bits 26 to 24 of the CPU address space where the GE Extended Frame Buffer and registers are located.

Programming notes

The contents of this register are not altered by drawing operations.

11.9.9. GRAPHICS EXTENDED APERTURE REGISTER (RW)

CR21 Access = 0x3X4h/0x3X5h Regoffset = 021h

7	6	5	4	3	2	1	0
A							
Default value after reset =FFh							

Bit Number	Mnemonic	Description
Bits 7-0	**A**	**Aperture.** The lower 6 address bits are prepended to the 16 least significant address bits in A0000h addresses to form a 22-bit address. This address is then used to map into the 4 MBytes Extended GE Register space. To use this feature, bits 7-6 must be set to ‘01’. Setting this register to FFh disables the aperture. Other values of this register can cause undefined results. The purpose of the aperure is to enable access to the extended memory mapped register in real mode.

Programming notes

The contents of this register are not altered by drawing operations.

11.9.10. REPAINT CONTROL REGISTER 4 (RW)

CR25 Access = 0x3X4h/0x3X5h Regoffset = 025h

7	6	5	4	3	2	1	0
Rsv		Rsv	HB	VB	VR	VD	VT
Default value after reset =FFh							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved**, read as ‘0’s.
Bit 5	**Rsv**	**Reserved.**
Bit 4	**HB**	**Bit 6 of the 7-bit wide Horizontal Blanking End register.**
Bit 3	**VB**	**Bit 10 of the 11-bit wide Vertical Blanking start register.**
Bit 2	**VR**	**Bit 10 of the 11-bit wide Vertical Retrace start register.**
Bit 1	**VD**	**Bit 10 of the 11-bit wide Vertical Display end register.**
Bit 0	**VT**	**Bit 10 of the 11-bit wide Vertical Total register.**

11.9.11. REPAINT CONTROL REGISTER 5 (RW)

CR27 Access = 0x3X4h/0x3X5h Regoffset = 027h

7	6	5	4	3	2	1	0
FIFO HWM					Rsv		
Default value after reset =D0h							

Bit Number	Mnemonic	Description
Bits 7-3	**FIFO HWM**	**FIFO High Water Mark**. When the FIFO occupancy rises above this value, the CRTC will stop filling the FIFO. This field should only ever be set by the BIOS during mode switches. Do not set this register to a value greater than the default value D0h - nothing will work.
Bits 2-0	**Rsv**	**Reserved**, read as ‘0’.

11.9.12. PALETTE CONTROL REGISTER (RW)

CR28 Access = 0x3X4h/0x3X5h Regoffset = 028h

7	6	5	4	3	2	1	0
DAC PD	DAC S	SPD	Rsv	LUT B	P F		
Default value after reset =08h							

Bit Number	Mnemonic	Description
Bit 7	**DAC PD**	**DAC Power Down.** Setting this bit to one turns off the digital to analog converters. This is useful for when a second graphics card is installed in the system and power needs to be saved by turning the motherboard graphics off.
Bit 6	**DAC S**	**DAC Setup.** This bit specifies the blanking pedestal. Zero indicates a blanking pedestal of 0 IRE, one indicates 7.5 IRE.
Bit 5	**SPD**	**Sense Power Down.** Setting this bit to one forces the DDC monitor sense circuits to power down.
Bit 4	**Rsv**	**Reserved**. Must be writtezn to ‘1’.
Bit 3	**LUT B**	**LUT Bypass.** Setting this bit to one bypasses the RAMDAC look up table (LUT) and allows pixels to drive the DACs directly. When this bit is set to zero the look up table is used to compute final pixel colors. This provides palette functionality for 8 bit and other low color modes and gamma correction (non-linearity compensation) for the 24 and 32 bit true color modes.
Bits 2-0	**P F**	**Pixel Format.** These 3-bits specify the pixel color depth and are encoded as follows in Table 11-21.

Bit 2	Bit 1	Bit 0	Pixel Format
0	0	0	VGA standard 8 bit
0	0	1	8 bit color (non-VGA)
0	1	0	15-bit (555) direct color
0	1	1	16-bit (565) direct color
1	0	0	24-bit (888) direct color
1	0	1	32-bit (8888) direct color
	1	X	Reserved

Table 11-21. Pixel Format

Programming notes

For 15-bit and 16-bit pixels, the 5 or 6 bits per color are shifted left by 3 or 2 bits and then presented to the 8-bit DACs or LUT address (depending on bit 3 above). The least significant bits are zeroed.

The following resolutions are supported at 75 Hz refresh rate for each of the above color depths when a 64 bit bank of DRAM is used for the frame buffer.

Pixel Format	Maximum Resolution
VGA (other than mode 13)	1024 x 768
VGA (mode 13)	640 x 480
8 bit (non-VGA)	1024 x 768
15 bit	1024 x 768
16 bit	1024 x 768
24 bit	800 x 600
32 bit	640 x 480

Interlaced monitors and timings are supported.

11.9.13. CURSOR HEIGHT REGISTER

For the description of this register, see section 12.11.1. on page 350.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.14. CURSOR COLOR 0 REGISTER A

For the description of this register, see section 12.11.2. on page 351.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.15. CURSOR COLOR 0 REGISTER B

For the description of this register, see section 12.11.3. on page 352.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.16. CURSOR COLOR 0 REGISTER C

For the description of this register, see section 12.11.4. on page 353.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.17. CURSOR COLOR 1 REGISTER A

For the description of this register, see section 12.11.5. on page 354.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.18. CURSOR COLOR 1 REGISTER B

For the description of this register, see section 12.11.6. on page 355.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.19. CURSOR COLOR 1 REGISTER C

For the description of this register, see section 12.11.7. on page 356.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.20. GRAPHICS CURSOR ADDRESS REGISTER 0

For the description of this register, see section 12.11.8. on page 357.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.21. GRAPHICS CURSOR ADDRESS REGISTER 1

For the description of this register, see section 12.11.9. on page 358.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.22. GRAPHICS CURSOR ADDRESS REGISTER 2

For the description of this register, see section 12.11.10. on page 359.

Must be written to ‘0’ when not using a Hardware cursor.

11.9.23. URGENT START REGISTER (RW)

CR33 Access = 0x3X4h/0x3X5h Regoffset = 033h

7	6	5	4	3	2	1	0
USP							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0	USP	**Urgent Start Position.** These bits represent the horizontal character count value at which urgency information will start to be generated for CRTC fetch requests. Prior to this position and after display enable negates, any CRTC fetches performed will be generated at low priority - ie. any CPU or blit operation will take precedence over CRTC regardless of CRTC FIFO occupancy. Once the horizontal character counter reaches this value, if CRTC FIFO occupancy is still below its low water mark then urgent fetches will be performed. Thereafter (and until the next display enable drops), as the CRTC FIFO drains, CRTC fetches will be marked urgent whenever the FIFO occupancy drops below its low water mark.

Programming notes

A value of 0xFFh means “always urgent” and should be used if the VGA screen is showing “Glitches”. By setting this value, the CPU and GE bandwidth will be reduced.

11.9.24. DISPLAYED FRAME Y OFFSET 0 REGISTER (RW)

CR34 Access = 0x3X4h/0x3X5h Regoffset = 034h

7	6	5	4	3	2	1	0
F Y O							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**F Y O**	**Frame Y Offset 0 Bits 7-0.** These bits represent bits 7-0 of a (2's complement) 16 bit scan line offset of the displayed frame relative to the Graphics Engine destination base.

11.9.25. DISPLAYED FRAME Y OFFSET 1 REGISTER (RW)

CR35 Access = 0x3X4h/0x3X5h Regoffset = 035h

7	6	5	4	3	2	1	0
F Y O							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**F Y O**	**Frame Y Offset 1 Bits 7-0.** These bits represent bits 15-8 of the displayed frame scan line offset relative to the Graphics Engine destination base.

11.9.26. INTERLACE HALF FIELD START REGISTER (RW)

CR39 Access = 0x3X4h/0x3X5h Regoffset = 039h

7	6	5	4	3	2	1	0
I H C							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**I H C**	**Interlace Horizontal Count.** This register defines the horizontal character count at which vertical timing is clocked during odd frames. When using interlaced operation, this register should be programmed to approximately half of the horizontal total value (CR0). There is no explicit interlace enable bit. Rather, when this register is programmed to FFh, interlace is disabled. The value of this register is defined to be FFh after reset (interlace disabled).

11.9.27. IMPLEMENTATION NUMBER REGISTER (R)

CR3A Access = 0x3X4h/0x3X5h Regoffset = 03Ah

7	6	5	4	3	2	1	0
I N							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**I N**	**Implementation Number.** Indicates the hardware implementation number for the graphics drawing and display subsystem. Table 11-22. below describes the interpretation of each value.

.

Value	Implementation
01h	STPC Implementation

Table 11-22. Implementation Number

11.9.28. GRAPHICS VERSION REGISTER (R)

CR3B			Access = 0x3X4h/0x3X5h				Regoffset = 03Bh
7	6	5	4	3	2	1	0
GVN							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**GVN**	**Graphics Version Number.** Indicates the hardware version number for the graphics drawing and display subsystem. Table 11-23. below describes the interpretation of each value.

.

Value	Implementation
01h	STPC Implementation

Table 11-23. Graphics Version Number

11.9.29. DRAM TIMING PARAMETER REGISTER (RW)

This register defines the type of memory populated and controls RAS# and CAS# timing.

CR3C			Access = 0x3X4h/0x3X5h				Regoffset = 03Ch
7	6	5	4	3	2	1	0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6		Memory type (see table below: Table 11-24.).
Bits 5-3		These three bits control the allowed timing templates as in Table 11-25. While the suggested use of different templates is listed in the table, any template can be used with either the FPM or EDO dram. The timings expressed here are in units of Graphics clock.
Bit 2		This bit determines the CAS pulse width as follows in Table 11-26. This parameter is only used for FPM DRAM and should be programmed to 1 (0.75 clk wide pulse) for both -60 and -70 ns DRAM.
Bit 1		This bit determines the delay between back to back read followed by write page hit cycles. This is only used for EDO DRAM. It has to be programmed to 1 (2-clock delay) for -60 EDO parts (see table below: Table 11-27.).
Bit 0		**RASoff**. This bit controls if RAS is kept active after the current DRAM access (see table below : Table 11-28.).

Bit 7	Bit 6	Description
0	0	Fast Page Mode (Default)
0	1	Extended Data Out
1	0	reserved for burst EDO if implemented
1	1	reserved for SDRAM in future product

Table 11-24. Memory type

Bit 5	Bit 4	Bit 3	cas-to-cas	ras-precharge	ras-to-cas	Intended use
1	0	0	0.5	1.5	1.5	FPM-60/70, EDO-70
1	0	1	0.5	2.0	2.0	EDO-60

Table 11-25. Timing Template Settings

Bit 2	CAS pulse width
0	0.5 clock wide pulse width.
1	0.75 clock wide pulse width.

Table 11-26. CAS pulse width

Bit 1	Clock delay
0	1 clock delay
1	2 clock delay

Table 11-27. Clock delay

Bit 0	RAS state
0	Keep RAS# active
1	deassert RAS#

Table 11-28. RAS state

11.9.30. DRAM ARBITRATION CONTROL REGISTER 0 (RW)

CR3D Access = 0x3X4h/0x3X5h Regoffset = 03Dh

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**Rsv**	**Reserved**. Must be set to ‘0’.

11.9.31. MISCELLANEOUS TEST REGISTER

CR3E Access = 0x3X4h/0x3X5h Regoffset = 03Eh

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**Rsv**	**Reserved**. Must be set to ‘0’.

11.9.32. DDC CONTROL REGISTER (RW)

CR3F Access = 0x3X4h/0x3X5h Regoffset = 03Fh

7	6	5	4	3	2	1	0
DDC WD		DDC RD		Rsv			
Default value after reset = FFh							

Bit Number	Mnemonic	Description
Bits 7-6	**DDC WD**	**DDC Write Data.** These two bits drive the DDC[1:0] open collector outputs. Writes to these bits affect the DDC[1:0] pins. The DDC[1:0] pins are open collector outputs which are externally pulled up. Thus, programming either of these bits to a one disables the output driver and allows the pin to act as an input whose status can be read via bits 5-4 of this register. Note that reads from these bits return the value of data last written to this register. This may not be the same as the data actually on the bus if another master is driving it. Bits 5-4 of this register accurately reflect the data on the bus no matter who is driving it.
Bits 5-4	**DDC RD**	**DDC Read Data.** These read-only bits return the read status of the DDC[1:0] pins.
Bits 3-1	**Rsv**	**Reserved.** These bits read are both readable and writable and must be programmed to ones to ensure future compatibility.
Bit 0	**Rsv**	**Reserved.** This bit must be programmed to ‘1’ for correct operation.

11.9.33. TV INTERFACE CONTROL REGISTER (RW)

CR40			Access = 0x3X4h/0x3X5h				Regoffset = 040h
7	6	5	4	3	2	1	0
TV IE	CCIR 656 E	VE	BTOE	VTV_HSYNC	FFA		
Default value after reset = FFh							

Bit Number	Mnemonic	Description
Bit 7	**TV IE**	**TV Interface Enable.** This bit enables the TV interface when set high.
Bit 6	**CCIR 656 E**	**CCIR-656 Enable.** When set to one, this bit enables the generation of CCIR-656 compatible timing codes onto the output pixel stream.
Bit 5	**VE**	**Video Enable** (active low). This bit multiplexes the TV output port between the graphics pipeline (bit 5 = ‘1’) and the video input port (bit 5 = ‘0’).
Bit 4	**BTOE**	**Bottom/Top Output Enable.** This bit controls the direction of the VTV_BT signal. When set to a one, VTV_BT is an output and is driven by the TV interface’s timing generator. Note that Video Input Port register 2B, bit 29 also controls the direction of this signal. The truth table is as follows (see table below:).
Bit 3	**VTV_HSYNC**	VTV_**HSYNC Output Enable.** This bit controls the direction of the VTV_HSYNC signal. When set to a one, VTV_HSYNC is an output and is driven by the TV interface’s timing generator. Note that Video Input Port register 2B, bit 28 also controls the direction of this signal. The truth table is as follows (see table below: Table 11-30.).
Bits 2-0	**FFA**	**Flicker Filter Algorithm.** These bits control the operation of the anti-flicker filter according to Table 11-31.

CR40[4]	VIP2B[29]	VTV_BT direction
0	0	input
0	1	output from VIP t/g
1	X	output from TVO t/g

Table 11-29. VTV_BT Signal direction

CR40[3]	VIP2B[28]	VTV_HSYNC direction
0	0	input
0	1	output from VIP t/g
1	X	output from TVO t/g

Table 11-30. VTV_HSYNC Signal direction

Bit 2	Bit 1	Bit 0	Function
X	0	0	Filter disabled
0	0	1	0:1:1 2 tap filter
0	1	0	1:2:1 3 tap filter
0	1	1	1:3:1 3 tap filter
1	0	1	0:1:1 2 tap filter -Y only
1	1	0	1:2:1 3 tap filter - Y only
1	1	1	1:3:1 3 tap filter - Y only

Table 11-31. Anti-Flicker Filter operation

11.9.34. TV HORIZONTAL ACTIVE VIDEO START A REGISTER (RW)

CR41 Access = 0x3X4h/0x3X5h Regoffset = 041h

7	6	5	4	3	2	1	0
TV							
Default value after reset =							

Bit Number	Mnemonic	Description
Bits 7-0	**TV**	These are bits [7:0] of the eleven bit wide horizontal active video start field. This field controls the positioning of the left hand side of the active **TV** display - to a one pixel granularity.

11.9.35. TV HORIZONTAL ACTIVE VIDEO START B REGISTER (RW)

CR42 Access = 0x3X4h/0x3X5h Regoffset = 042h

7	6	5	4	3	2	1	0
Rsv	Rsv			TV	WH		
Default value after reset =							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** For test and/or debug purposes, this bit may be used to reverse the order of chrominance and luminance bytes in the output stream. Normal operation requires this bit to be written as a zero.
Bits 6-4	**Rsv**	**Reserved.** These read/write bits should be written as zeroes to ensure future compatibility.
Bit 3	**TV**	**TV** output mode. This bit defines the output format on the four analog DAC outputs of the TV encoder. This bit defaults to zero on reset (see table below: Table 11-32.).
Bits 2-0	**WH**	These represent bits 10:8 of the eleven bit **Wide Horizontal** active video start field. See 11.9.34. on page 282.

Bit 3	RED_TV	GREEN_TV	BLUE_TV	CVBS
0	**red**	**green**	**blue**	**composite**
1	**chrominance**	**luminance**	**composite**	**composite**

Table 11-32. TV output mode

11.9.36. TV HORIZONTAL SYNC END A REGISTER (RW)

CR43 Access = 0x3X4h/0x3X5h Regoffset = 043h

7	6	5	4	3	2	1	0
WH							
Default value after reset =							

Bit Number	Mnemonic	Description
Bits 7-0	**WH**	These are bits [7:0] of the eleven bit **Wide Horizontal** sync end field. This field represents the width, in pixels, of the (interlaced) HSYNC signal produced by the TV interface. Bit 3 of CR40 controls whether this interlaced hsync is actually output.

11.9.37. TV HORIZONTAL SYNC END B REGISTER (RW)

CR44 Access = 0x3X4h/0x3X5h Regoffset = 044h

7	6	5	4	3	2	1	0
Rsv	Rsv	Rsv			WH		
Default value after reset =							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.** This bit should be programmed to be the same as CR10 bit 0. It is seperatly programmable purely for test and / or debug purposes.
Bit 6	**Rsv**	**Reserved.** This bit, when set to one, allows the luminance output of the colorspace converter (progressive scan) to be directly output onto the TV_YUV output bus. This is for test/debug use only and this bit should normally only ever be programmed to zero.
Bits 5-3	**Rsv**	**Reserved.** These read/write bits should be written as zeroes to ensure future compatibility.
Bits 2-0	**WH**	These represent bits 10:8 of the eleven bit **Wide Horizontal** sync end field.

11.10. ADDITIONAL MODES

11.10.1. FAST 132 CHARACTER WIDE TEXT MODE.

To meet the high bandwidth requirements of 132 column text mode, VGA Controller supports a special high speed text mode. For column widths of 96 characters and greater, bit 7 of extended register CR1C - the Repaint Control Register #3 must be set to one prior to loading the font tables into frame buffer plane two. Fonts may then be loaded in the standard VGA manner one Byte at a time at the end of which bit 7 of CR1C should be reset to zero.

Setting bit 7 of CR1C to one performs an address warping such that standard VGA font load cycles actually store fonts into plane two the following way:

Byte 0: Character Set 0, Font (ASCII) 0, Line 0

Byte 1: Character Set 0, Font (ASCII) 1, Line 0

...

Byte 255: Character Set 0, Font 255, Line 0

Byte 256: Character Set 1, Font (ASCII) 0, Line 0

...

Byte 511: Character Set 1, Font 255, Line 0

Byte 512: Character Set 2, Font (ASCII) 0, Line 0

...

Byte 2047: Character Set 7, Font 255, Line 0

Byte 2048: Character Set 0, Font (ASCII) 0, Line 1

Applications which load their own fonts independent of the motherboard BIOS will not be supported in 132 column modes because of the above requirements.

Note that the above organization of font data will ensure that 132 column mode bandwidth requirements are low enough to be satisfied by 64 bit wide DRAM frame buffers only. If the frame buffer is 32 bits wide, then the primary and secondary character map selects (SR3) should only ever be programmed such that both of the primary and secondary fonts are in the range 0-3 or both are in the range 4-7. Failure to observe this requirement will result in a garbaged screen.

11.11. INTERLACED MONITOR SUPPORT

Section 4.7.6.26a describes the “interlace half field start” register field. Setting this field to a value other than FFh (the power on reset default) enables interlaced CRT timing generation.

In interlaced timing mode, the horizontal and vertical timing parameters (CR0-CR7, CR10-CR12, CR15, CR16) should be programmed to values equal to what they would otherwise take in non-interlaced modes with the following modifications:

- Horizontal period must be an even number of character clocks. This results in the requirement that CR0[0] must equal ‘1’.

- Interlace half field start (CR39) must be set equal to CR4 - (CR0 + 5)/2.

- Vertical period must be an odd number of scan lines. That is, CR6[0] must be set to 1.

- Vertical overscan period should be an even number of scan lines. That is the vertical blank start field must be odd (CR15[0] = ‘1’) and vertical blank end field must be even (CR16[0] = ‘0’). If this is not observed the top and bottom lines of the border will be only half a scan line wide on alternate fields. If no border will be displayed then there is no retriction on vertical blank start and end.

All other registers should be programmed as they would for the same resolution and color depth in non-interlaced mode.

11.12. RAMDAC REGISTERS

11.12.1. PALETTE PIXEL MASK REGISTER (RW)

This eight-bit mask register is ANDed with the pseudo-color pixel before doing the palette look-up. This provides an alternate way of altering the displayed colors without changing the display memory or color palette.

Pixel_Mask Access = 0x3C6h Regoffset =

7	6	5	4	3	2	1	0
Default value after reset = FFh							

11.12.2. PALETTE READ INDEX REGISTER (W)

This register contains the index value for the read access to the 256 entries of the color palette. Each entry is 24-bits wide (8-bits each for R, G and B) and is read as sequence of 3-Bytes. After writing the index of the entry to be read, the actual contents of the selected palette entry are read by doing 3 consecutive Byte reads from the DAC Data port (3C9h) in sequence: 1) Red, 2) Green and 3) blue. This 3-Byte read sequence is aborted and a new one is started if either the Read or Write Index register is written before reading the third Byte.

After the third Byte of the sequence is read, the index register is automatically incremented to point to the next entry of the look-up table. If the index is FFh, it rolls over to 00h.

This register is a write only register. Reads from this address do not return the contents of the index register. The Palette state register contents are returned instead.

Read_Index Access = 0x3C7h Regoffset =

7	6	5	4	3	2	1	0
Default value after reset =							

11.12.3. PALETTE STATE REGISTER (R)

This is a read only register and contains the two least significant bits of the last IO writes to IO address 3C6h-3C9h.

Palette_State Access = 0x3C7h Regoffset =

7	6	5	4	3	2	1	0
Rsv						IO	
Default value after reset =							

Bit Number	Mnemonic	Description
Bits 7-2	**Rsv**	**Reserved.** The read back of this register is undefined.
Bit 1-0	**IO**	These bits contain the 2 LSBs of the address of the last IO write to ports 3C7h or 3C8h. ‘00’ indicate that the last write was to port 3C8h, ‘11’ indicate 3C7h.

11.12.4. PALETTE WRITE INDEX REGISTER (RW)

This register is similar to the Read index register and contains the index value for the write access to the 256 entries of the color palette. Each entry is 24-bit wide and are written as a sequence of 3-Bytes. After writing the index of the entry to be modified, the data values may be written to the DAC Data port (3C9h) in the sequence: 1) Red, 2) Green, and 3) Blue. This 3-Byte write sequence is aborted and new one is started if either the Read or Write Index register is written before writing the third Byte.

After the third Byte of the sequence is written, the index register is automatically incremented to point to the next entry of the look-up table. If the index is FFh, it rolls over to 00h.

Both the Read and the Write index registers, physically map to a single index register. However only the Write Index register can be read. Reads from the Read Index register return the contents of Palette state register.

Write_Index Access = 0x3C8h Regoffset =

7	6	5	4	3	2	1	0
Not initialized by reset							

11.12.5. PALETTE DATA REGISTER (RW)

This register is used in conjunction with the Read and the Write index register to access the look-up table. Reads from this port return the contents of the entry pointed to by the Read Index register and writes to this port modify the content of the look-up table entry pointed to by the Write Index register. Each look-up table entry is 24-bit wide and is read or written as a sequence of 3 Bytes. The read or write sequence is always Red, Green and Blue. The normal procedure for accessing the look-up table is to initialize one of the Index registers and follow it with an uninterruptible sequence of 3 reads/writes from this register.

For VGA backward compatibility, when bits 2-0 of CR28 are programmed to 000 (as they are after reset), the palette look up table is treated as if each entry was only 18 bits wide. In this case, writes to port 3C9h map data such that bits 5-0 of host data are written into bits 7-2 of the look-up table while bits 1-0 are zeroed out. Similarly, reads return bits 7-2 of look-up table data onto host bits 5-0 and zero out bits 7-6.

When bit 5 of register CR3E is set to one, reads and writes to this port access red, green and blue signature data instead of look-up table data.

To minimize the sparkle while accessing the look-up table, all 3-Bytes are read or written in a single video clock interrupting the screen repaint for one clock only. The interrupted pixel is painted with the same color as the previous pixel.

Palette_Data Access = 0x3C9h Regoffset =

7	6	5	4	3	2	1	0
Not initialized by reset							

11.13 DCLK Control registers

These registers control the Dot Clock or pixel clock which the VGA uses to display the pixels on the screen.

11.13.1. DCLK Control Register 00

This is one of the 4-pairs of Dot Clock Control registers. This pair (00 and 01) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 00.

DCLK00 Access = 022h/023h Regoffset =42h

7	6	5	4	3	2	1	0
Rsv	4BD				8BN		
Default value after reset = 0x76h							

Bit Number	Mneumonic	Description
Bit 7	**Rsv**	**Reserved.**
Bits 6-3	**4BD**	This the 4-bit M (divisor) value of the Dot Clock Synthesizer.
Bits 2-0	**8BN**	These are bits 7-5 of the 8 bit N (mulitiplier) of the Dot Clock synthesizer.

Programming notes

This register defaults to 0x76h at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 25.18 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.2. DCLK control register 01

This is one of the 4-pairs of Dot Clock Control registers. This pair (00 and 01) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 00.

DCLK01 Access = 022h/023h Regoffset =43h

7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0x95h							

Bit Number	Mneumonic	Description
Bit 7	**3BP0**	This is the bit 0 of the 3-bit P (exponent) value of the Dot clock synthesizer.
Bits 6-2	**8BN**	These are bits 4-0 of the 8-bit N (multiplier) value of the Dot Clock Synthesizer.
Bits 1-0	**3BP1**	These are bits 2-1 of the 3-bit P (exponent) value of the Dot clock synthesizer.

Programming notes

This register defaults to 0x95h at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 25.18 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.3. DCLK control register 10

This is one of the 4-pairs of dot clock control registers. This pair (10 and 11) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 01.

DCLK10 Access = 022h/023h Regoffset =44h

7	6	5	4	3	2	1	0
Rsv	4BM				8BN		
Default value after reset = 0x76h							

Bit Number	Mneumonic	Description
Bit 7	**Rsv**	**Reserved.**
Bits 6-3	**4BM**	This the 4-bit M (divisor) value of the Dot Clock Synthesizer.
Bits 2-0	**8BN**	These are bits 7-5 of the 8 bit N (mulitiplier) of the Dot Clock Synthesizer.

Programming notes

This register defaults to 0x76h at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 28.0 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.4. DCLK control register 11

This is one of the 4-pairs of Dot Clock Control registers. This pair (10 and 11) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 01.

DCLK11 Access = 022h/023h Regoffset =45h

7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0xEDh							

Bit Number	Mneumonic	Description
Bit 7	**3BP0**	This is the bit 0 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.
Bits 6-2	**8BN**	These are bits 4-0 of the 8-bit N (multiplier) value of the Dot Clock Synthesizer.
Bits 1-0	**3BP1**	These are bits 2-1 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.

Programming notes

This register defaults to 0xEDh at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 28.0 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.5. DCLK control register 20

This is one of the 4-pairs of Dot Clock Control registers. This pair (20 and 21) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 10.

DCLK20 Access = 022h/023h Regoffset =46h

7	6	5	4	3	2	1	0
Rsv	4BM				8BN		
Default value after reset = 0x5Bh							

Bit number	Mneumonic	Description
Bit 7	**Rsv**	**Reserved.**
Bits 6-3	**4BM**	This the 4-bit M (divisor) value of the Dot Clock Synthesizer.
Bits 2-0	**8BN**	These are bits 7-5 of the 8 bit N (mulitiplier) of the Dot Clock Synthesizer.

Programming notes

This register defaults to 0x5Bh at reset. This value when combined with the default value of the other half of this pair results in a dot clock of 40 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.6. DCLK control register 21

This is one of the 4-pairs of Dot Clock Control registers. This pair (20 and 21) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 10.

DCLK21 Access = 022h/023h Regoffset =47h

7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0x6Dh							

Bit Number	Mneumonic	Description
Bit 7	**3BP0**	This is the bit 0 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.
Bits 6-2	**8BN**	These are bits 4-0 of the 8-bit N (multiplier) value of the Dot Clock Synthesizer.
Bits 1-0	**3BP1**	These are bits 2-1 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.

Programming notes

This register defaults to 0x6Dh at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 40 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.7. DCLK control register 30

This is one of the 4-pairs of dot clock control registers. This pair (30 and 31) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 11.

DCLK30 Access = 022h/023h Regoffset =48h

7	6	5	4	3	2	1	0
Rsv	4BM				8BN		
Default value after reset = 0x6Eh							

Bit Number	Mneumonic	Description
Bit 7	**Rsv**	**Reserved.**
Bits 6-3	**4BM**	This the 4-bit M (divisor) value of the Dot Clock Synthesizer.
Bits 2-0	**8BN**	These are bits 7-5 of the 8 bit N (mulitiplier) of the Dot Clock Synthesizer.

Programming notes

This register defaults to 0x6Eh at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 60 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.13.8. DCLK control register 31 DCKL31 Index 49

This is one of the 4-pairs of dot clock control registers. This pair (30 and 31) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 11.

DCLK31 Access = 022h/023h Regoffset =49h

7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0x69h							

Bit Number	Mneumonic	Description
Bit 7	**3BP0**	This is the bit 0 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.
Bits 6-2	**8BN**	These are bits 4-0 of the 8-bit N (multiplier) value of the Dot Clock Synthesizer.
Bits 1-0	**3BP1**	These are bits 2-1 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.

Programming notes

This register defaults to 0x69h at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 60 MHz assuming 14.318 MHz oscillator clock as the reference input.

11.14 UPDATE HISTORY FOR VGA CONTROLLER CHAPTER

The following changes have been made to the VGA Controller Chapter on 07/12/99.

Section	Change	Text
11.9.29.	Added	Added support text.

The following changes have been made to the VGA Controller Chapter on 12/10/99.

Section	Change	Text
11.13	Added	Section 9.13. dclk control registers descriptions

The following changes have been made to the VGA Controller Chapter on 20/10/99.

Section	Change	Text
11.1.	Replaced	"~~Five~~" With "Seven"

The following changes have been made to the VGA Controller Chapter from Revision 1.0 to Release 1.2.

Section	Change	Text
11.1.	Replaced	"~~cycle~~" With "cycle because of complete memory remapping."
11.1.	Removed	The Graphics Engine provides a high performance 64-bit windows accelerator. Hardware acceleration is provided for BITBLTs, transparent BLTs and fills and text (generalised bit map expansion). The Graphics Display Controller is fully integrated with the input and display of video signals (see Next Chapter).
11.2.	Added	This along with peak video bandwidth of 320 MBytes/sec (using EDO DRAM) enables the VGA controller to support 1024 x 768 x24 and 800 x 600 x 32 resolutions at 75 Hz refresh rate
11.2.	Removed	The RAMDAC supports an external voltage reference source as well as DDC monitor sense functionality.
11.2.	Replaced	"~~If an external add-in VGA card is placed in the system, the on-chip VGA controller can be disabled~~." With If an external add-in VGA card is placed in the system, the on-chip VGA controller can be disabled in order to work with this external card. It is possible to enable / disable the system back to dual use VGA controller if necessary.
11.4.4.	Replaced	"0X4E8h" With "0X3C3h (RW)"
11.4.5.	Replaced	"~~Bits 3-2 **Clock Selects.** These register bits drive two clock select output pins of the DPC chip. The pins are named CLOCK0 and CLOCK1~~." With "Bits 3-2 **Clock Selects.** Selects one of the four synthesizer pairs when DCLK source is onchip PLL's."
11.4.7.	Replaced	"~~Bits 6-5 **Shift mode**~~" With "Bits 6-5 **Shift mode** These values are given in Table 11-9 "

Section	Change	Text
11.7.1.	Added	A sample program could be as follows; *mov DX, 3DAh* *in AL, DX* *mov AL, Index* *mov DX, 3C0h* *out DX, AL* *mov AL, Data* *out DX, AL*
11.8.	Replaced	"~~1280x1024~~" With "1024x768"
11.8.1.	Replaced	"~~7-6~~" With "7"
11.8.1.	Replaced	"~~5-0~~" With "6-0"
11.8.15.	Added	"The start address in the CRC, CRD and CRIG does not define the offset within the frame buffer of the 1st displayed pixel but this value divided by 4."
11.8.25.	Added	"Bit 1 **Memory Segmentation Bit 14.** If set to a '0', the row scan counter bit 1 is substituted for memory address bit 14 during display refresh. This has the affect of segmenting the address space such that every other scan line pair is 8K apart. In combination with bit 0, this bit can segment the address space in 4-banks. No such substitution takes place if this bit is set to a '1'."
11.8.25.	Added	Bit 0 **Memory Segmentation Bit 13.** This bit is similar to Bit 1 above in that when set to a '0', row scan counter bit 0 is substituted for display memory address bit 13 during active display time. No such substitution takes place if this bit is set to a '1'.
11.8.25.	Removed	Bit 0 **Memory Segmentation Bit 13.** This bit is similar to Bit 1 above in that when set to a '0', row scan counter bit 0 is substituted for display memory address bit 13 during active display time. No such substitution takes place if this bit is set to a '1'.
11.9.	Added	A typical sequence in 80X86 assembly could be; max DX, 3C4h movAX, 5706h outDX, AX
11.9.1.	Replaced	"~~Bit 7 **"Turbo Mode" Enable.** Programming this bit to '1' enables enhanced performance mode. The effect is to increase system performance by reducing the bandwidth required by the display graphics datapath thus giving more bandwidth back to the CPU. This bit resets to zero (performance enhancement disabled).~~ ~~Note that after "turbo mode" is turned on, any software that alters CRTC registers which "move the screen" (such as start address, display offset etc.) should also program bit 6 to a one. Alternatively, the software can do a dummy host write to the VGA frame buffer.~~" With "Bit 7**-**6 *Reserved.* Must be written to '0'."
11.9.2.	Added	"Note; Vertical / Horizontal Synch not toggling is defined by the VESA specification for Monitor Power Down State."
11.9.5.	Added	"Register = A000h mapped to the frame buffer and can be either 8 KBytes or 32 KBytes The contents of this register are not altered by drawing operations."
11.9.5.	Removed	"~~The contents of this register are not altered by drawing operations~~."

Section	Change	Text
11.9.6.	Added	"Register = A800h mapped to the frame buffer and can be either 8 KBytes or 32 KBytes "
11.9.6.	Added	Figure 14
11.9.7.	Replaced	"~~Bit 7 **Exen.** Writing a '1' in this bit enables the GE extended functionality. Writing a '0' disables it. After reset, this bit is set to '0'~~." With "Bit 7 **Exen.** Writing a '1' in this bit enables the GE extended functionality and also direct access to the frame buffer as defined by GBASE (CR20). Writing a '0' disables it. After reset, this bit is set to '0'."
11.9.10.	Replaced	"~~Bit 5 **Blank With Display Enable.** When this bit is set to one use display enable to generate the blank signal. This eliminates the border but allows the blank timing generator to be used for sophisticated low water mark control when video in a window is displayed.~~ ~~Consider the following diagram in Figure 9-1:~~ ~~In this scenario, the CRTC low water mark is determined as follows:1~~ ~~Where:~~ ~~If crtc_lwm is set to 255 bytes, then the above operation can be used to guarantee CRTC ownership of the bus at the beginning of the video window: Setting crtc_lwm to 255 will cause the CRTC to request ownership of the DRAM at the rising edge of "blank". If "blank" is high prior to the active video window for approximately 23 GCLKs or more, the CPU or any other potential owner will be guaranteed by design to relinquish the bus in time for the start of the video window~~." With "Bit 5 *Reserved.*"
11.9.12.	Replaced	"~~Bit 4 **Enable On Chip Voltage Reference.** When this bit is set to one, the on-chip analog voltage reference is used for the analog to digital converters. In this case the VREF pin should be a no-connect. When this bit is zero, the on-chip reference is disabled and an external reference must be supplied - connected to the VREF pin. This bit defaults to zero after reset~~ " With "Bit 4 *Reserved.* Must be writtezn to '1'."
11.9.12.	Removed	"~~These are the resolutions supported at 75 Hz when only a 32 bit wide frame buffer is available:~~"
11.9.13.	Replaced	"~~Bit 7 **Cursor XOR Pre/Post Look Up Table.** When this bit is set to one, the graphics cursor XOR operation is performed before the look up table. The default behavior, when this bit is set to zero, is for the XOR operation to happen after the look up table. This is correct for 15, 16, 24 bit per pixel modes but not 8bpp.~~ ~~Bits 6-0 **Cursor height.** This field represents the vertical extent of the graphics cursor in scan lines. Setting this to zero effectively turns the graphics cursor off. Values greater than 40h (decimal 64) are meaningless and produce unpredictable results.~~ ~~Note: there is no cursor width register - the width is always 64 pixels. If a narrower cursor is required, pad the bitmap on the right with transparent cursor color (pad the AND plane with '1's on the right and the XOR plane with '0's).~~ ~~This register is set to 00h after reset~~." With For the description of this register, see section section 12.11.1. on page 350 Must be written to '0' when not using a Hardware cursor

Update History for VGA Controller chapter

Section	Change	Text
11.9.14.	Replaced	"~~Bits 7-0~~ ~~**Cursor Color 0 Red.** These bits are the red component of cursor color 0~~." With "For the description of this register, see section section 12.11.2. on page 351 Must be written to '0' when not using a Hardware cursor"
11.9.15.	Replaced	"~~Bits 7-0~~ ~~**Cursor Color 0 Green.** These bits are the green component of cursor color 0.~~ ~~This register is undefined after reset~~." With "For the description of this register, see section section 12.11.3. on page 352 Must be written to '0' when not using a Hardware cursor"
11.9.16.	Replaced	"~~Bits 7-0~~ ~~**Cursor Color 0 Blue.** These bits are the blue component of cursor color 0.~~ ~~This register is undefined after reset~~." With "For the description of this register, see section section 12.11.4. on page 353 Must be written to '0' when not using a Hardware cursor"
11.9.17.	Replaced	"~~Bits 7-0~~ ~~**Cursor Color 1 Red.** These bits are the red component of cursor color 1.~~ ~~This register is undefined after reset~~." With "For the description of this register, see section section 12.11.5. on page 354 Must be written to '0' when not using a Hardware cursor"
11.9.18.	Replaced	"~~Bits 7-0~~ ~~**Cursor Color 1 Green.** These bits are the green component of cursor color 1.~~ ~~This register is undefined after reset~~." With "For the description of this register, see section section 12.11.6. on page 355 Must be written to '0' when not using a Hardware cursor"
11.9.19.	Replaced	"~~Bits 7-0~~ ~~**Cursor Color 1 Blue.**These bits are the blue component of cursor color 1.~~ ~~This register is undefined after reset~~." With "For the description of this register, see section section 12.11.7. on page 356 Must be written to '0' when not using a Hardware cursor"
11.9.20.	Replaced	"~~Bits 7-1~~ ~~**Cursor AND Address Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the cursor's AND mask. The cursor's XOR mask begins at this address + 512. This memory must be aligned on a 1 KByte boundary. For a discussion of DRAM linear addresses, see section tbc.~~ ~~Bit 0 *Reserved.*This bit should be written as zero~~." With "For the description of this register, see section section 12.11.8. on page 357 Must be written to '0' when not using a Hardware cursor"

Section	Change	Text
11.9.20.	Removed	“~~Note that the cursor bitmap is ordered such that the top left hand corner of the cursor is represented by bit 7 of the byte addressed by this field (AND) and bit 7 of the byte at 512 plus this address (XOR plane). The next pixel right is represented by bit 6 of these bytes and so on until the bottom right hand pixel is represented by bit 0 of the byte locat ed at this address plus 511 (AND) and bit 0 of the byte at 1023 plus this address.~~”
11.9.22.	Replaced	“~~Bits 7-0 **Cursor AND Address Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the cursor’s AND mask. For a discussion of DRAM linear addresses, see section 5.4.5.~~” With “For the description of this register, see section section 12.11.9. on page 358“
11.9.23.	Replaced	“~~Bits 7-3 *Reserved.* These bits should be written as zero.~~ ~~Bits 2-0 **Cursor AND Address Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the cursor’s AND mask. For a discussion of DRAM linear addresses, see section tbc.~~ ~~This register is undefined after reset.~~” With “For the description of this register, see section section 12.11.10. on page 359“
11.9.23.	Added	“A typical value to be programmed is the one in CR00. “
11.9.26.	Removed	“~~Housekeeping Address 0CR36 3X5h Index 36(RW)~~ ~~Bits 7-1 **Housekeeping Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the “turbo mode” housekeeping region. See description of CR19 bit 7.~~”
11.9.26.	Added	“Interlace Half Field Start CR39 3x5h Index 39 (RW) Bits 7-0 **Interlace Horizontal Count.** This register defines the horizontal character count at which vertical timing is clocked during odd frames. When using interlaced operation, this register should be programmed to approximately half of the horizontal total value (CR0). There is no explicit interlace enable bit. Rather, when this register is programmed to FFh, interlace is disabled. The value of this register is defined to be FFh after reset (interlace disabled).”
11.9.27.	Added	“Implementation Number Register CR3A 3X5h Index 3A (R) Bits 7-0 **Implementation Number.** Indicates the hardware implementation number for the graphics drawing and display subsystem. Table 11-21 below describes the interpretation of each value”
11.9.28.	Added	“Graphics Version Register CR3B 3X5h Index 3B (R) Bits 7-0 **Graphics Version Number.** Indicates the hardware version number for the graphics drawing and display subsystem. Table 11-22 below describes the interpretation of each value.”
11.9.29.	Removed	“~~4.7.6.25 Housekeeping Address 1~~ ~~CR37 3X5h Index 37 (RW)~~ ~~Bits 7-0 **Housekeeping Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the “turbo mode” housekeeping region.~~”
11.9.29.	Added	“4.7.6.25 DRAM Timing Parameter register CR3C 3X5h Index 3C (RW) See section 4.4.1 for the description of this register”

Update History for VGA Controller chapter

Section	Change	Text
11.9.33.	Removed	"~~Housekeeping Address 2~~ ~~CR38 3X5h Index 38 (RW)~~ ~~Bits 7-3 *Reserved.*These bits should be written as zero.~~ ~~Bits 2-0 **Housekeeping Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the "turbo mode" housekeeping region.~~ ~~This register is undefined after reset~~."
11.9.33.	Replaced	"MPEG" With "Video"
11.9.33.	Replaced	"HSYNC" WXith "VTV_HSYNC"
11.9.34.	Removed	"~~Interlace Half Field Start~~ ~~CR39 3x5h Index 39 (RW)~~ ~~Bits 7-0 **Interlace Horizontal Count.** This register defines the horizontal character count at which vertical timing is clocked during odd frames. When using interlaced operation, this register should be programmed to approximately half of the horizontal total value (CR0).~~ ~~There is no explicit interlace enable bit. Rather, when this register is programmed to FFh, interlace is disabled. The value of this register is defined to be FFh after reset (interlace disabled).~~"
11.9.35.	Removed	"~~Implementation Number Register~~ ~~CR3A 3X5h Index 3A (R)~~ ~~Bits 7-0 **Implementation Number.** Indicates the hardware implementation number for the graphics drawing and display subsystem. The table below describes the interpretation of each value~~."
11.9.36.	Removed	"~~Graphics Version Register~~ ~~CR3B 3X5h Index 3B (R)~~ ~~Bits 7-0 **Graphics Version Number.** Indicates the hardware version number for the graphics drawing and display subsystem. The table below describes the interpretation of each value~~."
11.9.37.	Removed	"~~DRAM Timing Parameter register~~ ~~CR3C 3X5h Index 3C (RW)~~ ~~See section 4.4.1 for the description of this register~~"
11.10.1.	Removed	"~~1. CR25[5] should be set to 1.7~~ ~~2. Horizontal Blank start register should be programmed to be at least 23 gclks earlier than the rising edge of h_win (or the X coordinate of video window).~~ ~~3. Horizontal Blank End register value should be later than the rising edge of h_win and earlier than the falling edge of h_win (The value should be within the video window). This can require up to 7 bits. CR3[4-0] specify the least significant 5-bits of the 7-bit wide Horizantal Blank End value.~~ ~~The sixth bit is located in CRTC Horizantal Retrace End register CR5[7], and the 7th bit is located in CRTC Extended register CR25[4].~~ ~~Programming CR25[5] will disable the display of overscan color by ignoring horizontal and vertical blank signal, and instead use the invert of horizontal and vertical display enable as the blank signals. CR2 and CR3 (blank start and blank end registers) are then used to generate a *new* blank signal used in the arbitration scheme.~~"
13.4.2	Removed	"~~Turbo Graphics Mode.~~ ~~Proprietary techniques are used to reduce the bandwidth required for the screen repaint in extended graphics modes. This mode is totally user transparent and is entered simply by programming bit 7 of extended register CR19 to a one. The only requirement made by this mode is that 32K of frame buffer space must be made available for exclusive use by VGA Controller for house-keeping functions.~~ "

12. GRAPHICS ENGINE

12.1. INTRODUCTION

The Graphics Engine (GE) performs limited graphics drawing operations. The results of these operations changes the contents of the on-screen or off-screen frame buffer areas of DRAM memory.

Pixel depths of 8, 16, 24 and 32 bits are fully supported by the GE.

12.2. MEMORY ADDRESS SPACE

The extended (non-VGA) graphics and video functions of Graphics Engine occupy 16 MBytes of memory address space. This space can be located anywhere in the memory on any 16 MByte boundary between 128 MBytes and 256 MBytes. The 16 MByte region is divided into four parts as shown in Figure 12-1.

In this figure, the GBASE is Extended CRTC Register 20 (CR20) and provides bits 26 through 24 of the starting address where the CPU sees the extended graphics and video functionality.

This 16-MByte space can be linearly (one to one) mapped in the CPU address space or can be accessed via 64K aperture located at A0000h-AFFFFh. The aperture access method is described in more detail in section “CR21 Graphics Extended Aperture Register” and facilitates the access to extended functionality in real mode.

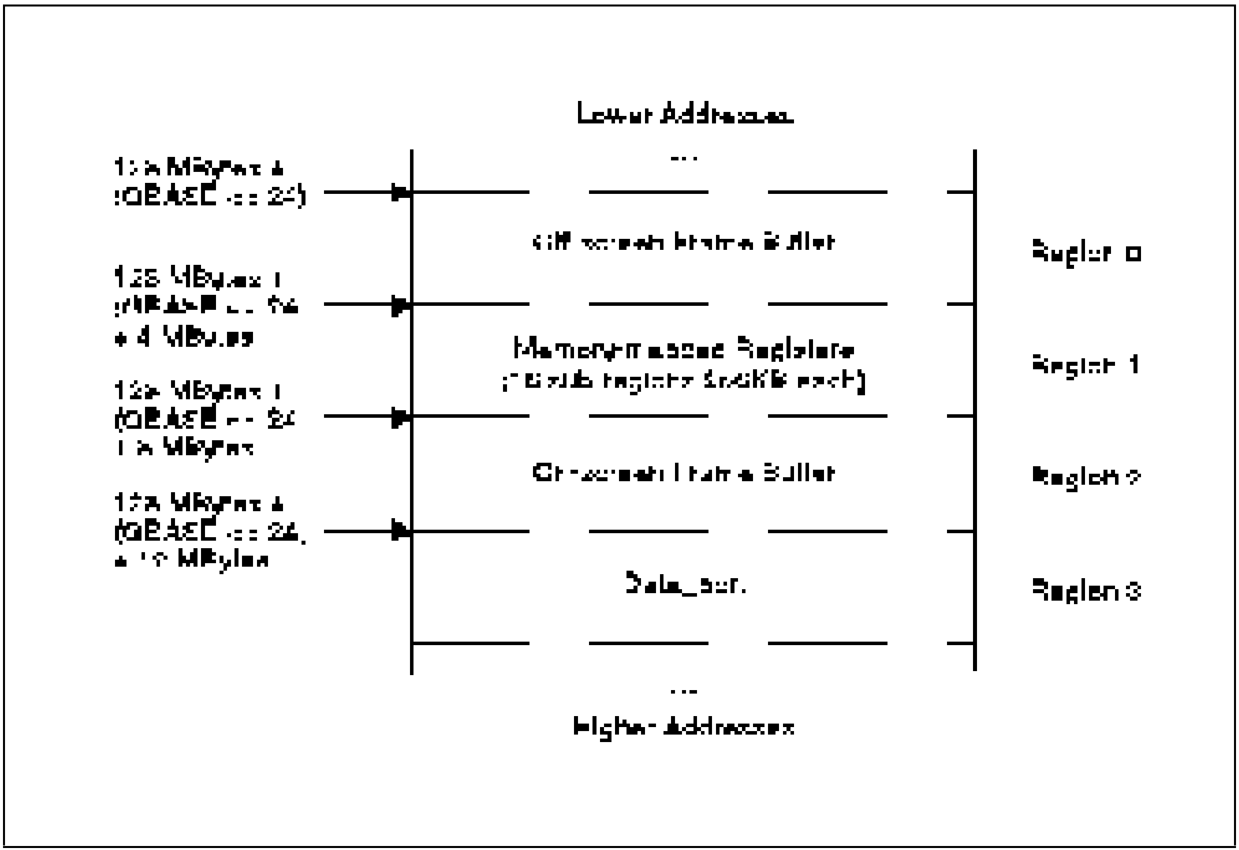


Figure 12-1. GE memory Map

Two 4-MByte regions are dedicated to the frame buffer. The frame Buffer reads from either: 128MBytes + (GBASE<<24) or 128MBytes + (GBASE<<24) + 8MBytesand they are done identically.

However, writes to any area of the Frame Buffer that might be displayed should be done to the region 128MBytes + (GBASE<<24) + 8MBytes. Writes to areas of the Frame Buffer that are not displayed should be done to 128MBytes + (GBASE<<24). The Frame Buffer address used by the host also informs the GE whether the access is being made to an on-screen or off-screen drawing area. It is important that the software never draws using an off-screen area address to a portion of the Frame Buffer that is currently being displayed in order to be compatible with future versions of the GE.

The Frame Buffer addresses loaded into GE registers are from 0 to 4MBytes. The source, pattern or destination of a GE operation can be located anywhere in Frame Buffer DRAM. However, the DRAM physical address must be known; the entire operand must be contiguous in physical memory (the GE does not do scatter/gather), and the operands must not move from the specified memory location until the drawing operation is completed.

All registers needed for the extended graphics and video functionality are mapped in a 4-MByte region of its own. This region is further divided in 16 256KByte sub-regions illustrated in Table 12-1. below:

Sub-region	Region function
0	2-D Graphics Engine registers
1	Reserved
2	Video overlay registers
3-7	Reserved for future functionality
8	Video Input Port Registers
9-15	Reserved for future functionality

Table 12-1. Graphic memory subdivisions

Writes done to any double word between (128MBytes + (GBASE << 24) + 12 MBytes) to (128MBytes + (GBASE << 24) + 16 MBytes) will be the same as a write to the Data_port register of the 2-D graphics engine. This region of 1 million aliases of the Data_port is provided to allow the use of string move instructions for Host-to-screen BitBlts.

All registers can be read with accesses of any width. The CPU can read any register via Byte (8-bit), word (16-bit), or double-word (32-bit) accesses. Writes, must be done using double-word (32-bit) transfers.

Note that the contents of all GE registers are not defined after reset.

Software must initialize all registers upon power-up before attempting any drawing operation.

12.3. DUMB FRAME BUFFER ACCESS

The CPU can access the frame buffer memory as ordinary memory. It can read from or write to the frame buffer using any memory access instruction, and any data width. Thus, the CPU can access the frame buffer as if it were an unaccelerated display subsystem. This access by the CPU is permitted regardless of the GE's busy status. Therefore, software must be careful to avoid race conditions or clashes if writing to the frame buffer when the GE is busy.

12.4. ADDRESSING

The GE's frame buffer and extended registers may be accessed by the CPU via two methods: extended addresses, or A0000h-AFFFFh addresses. The former method allows direct access to the 16 MByte GE address range via 32-bit addresses. The A0000h-AFFFFh addressing method maps a 64K window of the 16 MByte GE address space into the address range A0000h-AFFFFh. For additional detail on the A0000h-AFFFFh addressing, see CR21 Graphics Extended Aperture Register.

The Frame Buffer address used by the host also informs the GE whether the access is being made to an on-screen or off-screen drawing area. It is important that the software never draws using an off-screen area address to a portion of the Frame Buffer that is currently being displayed.

The addresses that are loaded into GE registers are the physical DRAM addresses after the OS address translation and GE host address remapping is done. The Frame Buffer addresses loaded into GE registers are from 0 to 4 MBytes. The source, pattern or destination of a GE operation can be located anywhere in the Frame Buffer space.

12.5. VGA OPERAND SOURCES

The GE operates on data which can originate in one of three possible areas:

1) The frame buffer memory (i.e. a location in the DRAM memory that is dedicated to the graphics subsystem, and which may or may not be currently displayed by the CRT controller)

2) The host-supplied data

3) The on-chip color registers

12.5.1. OPERAND SELECTION

Some operands are color pixels and others are monochrome bitmaps. In general, the data written to the Destination address is the result of a Raster Operation (ROP) performed upon three pixel-depth color inputs:

1) The Source, which can originate from frame buffer memory (for Screen-to-screen BitBlts), from the Host (for Host-to-screen BitBlts), or from the Foreground and Background color registers.
2) The pattern, must originate from the frame buffer.
3) The destination, must originate from the frame buffer.

When one or more of these operands are the inputs to an 8-bit Windows' ROP, the result is written to the destination.

If the ROP does not use a source operand, then the "Source" field of the ROP register must be set to CONSTANT_FILL (see ROP register 9.5.5.11) to prevent wasted performance due to needless operand fetching.

If the ROP requires destination data reads, then the "Dst" field of the ROP register must be set to '1'. If destination reads are not required, then this field should be set to '0'.

If the ROP requires pattern data or uses color transparent mode, then the "Pat" field of the ROP register must be set to '1'. If no pattern or color transparency is being used in the operation, then this field should be set to '0'.

12.5.2. TRANSPARENT MODE

Transparent mode drawing leaves some of the destination pixels untouched. The GE supports four types of transparent mode drawing:

1) Bitmap transparency, where bits that are ‘1’ are expanded to the Foreground color register value and drawn, but bits that are expanded to ‘0’ are not drawn.

2) Pattern transparency, where any Pattern Bytes that are ‘0’ suppresses writing to the corresponding destination pixel Bytes.

3) Source transparency, where any Source pixel which either matches the value or do not match the value of the source transparency register is not drawn.

4) Destination transparency, where any Destination pixel that either matches the value or does not match the value of the source transparency register is not drawn.

These modes are controlled by fields in the ROP register.

12.6. VGA OPERAND FRAME BUFFER ADDRESSES

The GE fetches needed data from the frame buffer area. The software identifies these areas with an operand base address, unsigned X and Y Indeces from this base address, and a pitch for that region. The pitch is the Byte distance between two pixels which are in the same X position of adjacent scan lines.

The frame buffer is addressed using DRAM linear addresses. These are the addresses that the DRAMs are presented with. The frame buffer starts at DRAM linear address 0 and continues until the top of the frame buffer. The system’s physical addresses are mapped to above the frame buffer. To accommodate a more natural view of the frame buffer, the GE implements X-Y addressing. An operand’s base address, pitch, X and Y components are combined in the GE, to form the associated DRAM linear address. The base component of an operand is the DRAM linear address at the start of that operand. That address can range from 0 to the maximum size of the frame buffer, depending upon where the operand is located in the frame buffer.

A pixel’s X coordinate is usually expressed as an unsigned Byte quantity, the number of Bytes from the left edge of a scan line.

If the X_dir field of the Pixel_depth register is ‘0’, advancing from left-to-right, then X points to the least-significant Byte of the starting pixel.

If the X_dir field of the Pixel_depth register is ‘1’, advancing from right-to-left, then X points to the most-significant Byte of the starting pixel.

Mathematically, consider a BitBlt region that starts at (x0, y0), where “x0” is in pixels. This region is W+1 Bytes wide, is H+1 scan lines high and has BPP (Bytes-per-pixel).

Then the starting address that must be programmed into the GE is depend on the X_dir and Y_dir fields of the Pixel_depth register. This is illustrated in Table 12-2.

X_dir	Y_dir	Starting Address
0	0	(x0 * BPP, y0)
0	1	(x0 * BPP, y0 + H)
1	0	(x0 * BPP + W, Y0)
1	1	(x0 * BPP + W, Y0 + H)

Table 12-2. Detail GE starting address register

Note that movement in the negative X direction (i.e. X_dir set to ‘1’) is only defined for Screen-to-screen color BitBlts.

When bitmap expansion is enabled, the X field of the Src_XY register is a bit address and not a Byte address. In other words, the least significant three bits of Src_XY.X refers to the bit within a Byte of bitmap data.

Internally, the GE performs its calculations using the X and Y coordinates. When a DRAM linear address is needed, for example to write a destination pixel, the address is computed using:

linear_address = operand_base + (Y * pitch) + X *BPP

The multiplication by pitch is done using hardwired shifts and adds. The pitch is actually specified as a group of 4 shift codes. For each non-zero shift code, the Y address is shifted by a corresponding number of bits and then added to the total. The resulting sum is then added to X and the base address to obtain the DRAM linear address. The shift values supported are shown in Table 12-3.

Value	Shift0	Shift1	Shift2	Shift3
000	0	0	0	0
001	0	0	32 * Y	1024 * Y
010	0	64 * Y	64 * Y	2048 * Y
011	128 * Y	128 * Y	128 * Y	4096 * Y
100	256 * Y	256 * Y	256 * Y	n/a
101	512 * Y	512 * Y	512 * Y	n/a
110	1024 * Y	1024 * Y	0	n/a
111	2048 * Y	0	0	n/a

Table 12-3. Shift values supported

The operand base addresses must be aligned to 32 Bytes (that is, the 5 least significant bits of the address must be zeros).

The supported pitches (in Bytes) are:

0, 32, 64, 96, 128, 160, 192, 224, 256, 288, 320, 352, 384, 416, 448, 512, 544, 576, 608, 640, 672, 704, 768, 800, 832, 896, 1024, 1056, 1088, 1120, 1152, 1184, 1216, 1248, 1280, 1312, 1344, 1376, 1408, 1440, 1472, 1536, 1568, 1600, 1632, 1664, 1696, 1728, 1792, 1824, 1856, 1920, 2048, 2080, 2112, 2144, 2176, 2208, 2240, 2272, 2304, 2336, 2368, 2400, 2432, 2464, 2496, 2560, 2592, 2624, 2656, 2688, 2720, 2752, 2816, 2848, 2880, 2944, 3072, 3104, 3136, 3168, 3200, 3232, 3264, 3328, 3360, 3392, 3456, 3584, 3616, 3648, 3712, 3840, 4096, 4128, 4160, 4192, 4224, 4256, 4288, 4320, 4352, 4384, 4416, 4448, 4480, 4512, 4544, 4608, 4640, 4672, 4704, 4736, 4768, 4800, 4864, 4896, 4928, 4992, 5120, 5152, 5184, 5216, 5248, 5280, 5312, 5376, 5408, 5440, 5504, 5632, 5664, 5696, 5760, 5888, 6144, 6176, 6208, 6240, 6272, 6304, 6336, 6400, 6432, 6464, 6528, 6656, 6688, 6720, 6784, 6912, 7168, 7200, 7232, 7296, 7424, 7680.

12.6.1. COMMAND INITIATION

The destination coordinate register, Dst_XY, appears multiple times in the address space. Reading from any of these appearances, or aliases, is equivalent. Writing to most of these aliases also has the effect of initiating a drawing command. Which command is begun depends upon the address written to. There is also an address that just provides write access to the Destination register, with no other side-effects.

The operations are encoded in the Dst_XY register shown in Table 12-5.

Where “GBASE” is the contents of the Extended CRTC Register 20 (CR20). Bits 23 through 16 are ‘01000001’ to identify this as a Dst_XY register access. This is shown in Table 12-4.

CMD	Operation
00	Simple BitBlt, all registers must be set up before this command is issued (the Count field is ignored)
01	Width-specified BitBlt, the Count field of the address is used as the width for the operation, all other relevant registers (height, ROP, etc.) must be set up before this command is issued
10	Height-specified BitBlt, the Count field of the address is used as the height for the operation, all other relevant registers (width, ROP, etc.) must be set up before this command is issued
11	Write to Dst_XY without starting a BitBlt operation, (used for diagnostic applications)

Table 12-4. CMD operations

When the CPU writes to the Dst_XY register using one of these operation aliases, the register write is completed and then the associated operation is begun. Thus, to perform an operation, the CPU writes to all but the Destination register. Then the last write is done to the Dst_XY register using one of the above aliases.

The ROP register also has fields that control the Source data (Screen, Host or Foreground color register), enables/disables bitmap expansion, determines if the drawing is done in one of the transparent modes.

Screen-to-screen BitBlts are done as a BitBlt with the Source set as the Screen and bitmap expansion disabled.

Host-to-screen BitBlts are done as a BitBlt with the Source set as Host and the bitmap expansion disabled.

Rectangular fills are done as a BitBlt with the Source set to the Foreground color register and the bitmap expansion disabled.

Text drawing using bit-packed font data provided by the Host is done as BitBlt with the Source set as Host and the bitmap expansion enabled. A transparent mode may also be specified.

Lines are not generally supported by the GE, but horizontal and vertical line segments can be quickly implemented as Width-specified BitBlts with the Height register set to 0 (to indicate a single pixel high BitBlt), or as Height-specified BitBlts with the Width register set to one less than the pixel depth.

		4-MByte region 1 (memory mapped regs)					
			256KByte sub-region 0				
31 27	26 24	23 22	21 18	17 - 16	15 14	13 2	1 - 0
00001	<GBASE>	0 1	0000	0 - 1	<Cmd>	<Count>	1 - 0

Table 12-5. Encoded Dst_XY registers

12.7. DRAWING ENGINE REGISTERS

The software controls the graphic drawing by writing to the GE's registers to set-up and initiate an operation. Any data that must be provided by the host is written to the Data_port. The Data_port can be referenced via the Data_port "register" or via the 4 MB window of aliases.

All registers can be read with accesses of any width. The CPU can read any register via Byte (8-bit), word (16-bit), or double-word (32-bit) accesses.

A register that is exclusively for the use of software, "Xtra", is included in the GE but has no influence on any drawing operation or on the display.

12.8. REGISTER ACCESS

Except for the Dst_XY register, discussed in the previous section, the memory-mapped GE registers and the Data Port are accessed by reading or writing to an address of the form. This is illustrated in Table 12-6.

Where “Index” specifies the offset of the register to be accessed from the start of the GE memory-mapped register address space. The least significant 2 bits of Index will always be ‘00’. The following sections will list the “Index” value along with a description of each register. Reads may be done in any width, but writes must be done as 32 or 64 bit transfers.

In general, the CPU should not write to any of the GE registers when the GE is busy. If such an access is done, the CPU may be held for a long period of time, possibly for the duration of a large BitBlt. Reads of GE registers (except for Status) may return invalid data if the GE is busy.

The Dst_XY, Src_XY, Width and Height registers are double buffered and the CPU may write the next values to these registers while a prior operation is being performed. The last write to any double buffered register done before a write to Dst_XY will be the one used for the next operation. Any writes done to a GE register after a write is done to the Dst_XY while the GE is busy will hold the CPU until the first operation is completed and the pending register values are used for the second operation. During normal operation, the CPU writes to the Dst_XY register for text and line segments, reducing the hold period as much as possible.

The GE Status register may be accessed at any time.

12.8.1. DATA PORT ACCESS

The CPU writes Host data to the GE through the Data Port for Text and Host-to-screen BitBlt operations. The Data Port appears as one of the registers, as discussed in the previous section. Behind the Data Port, the Data FIFO buffers incoming data from the CPU. The Data Port is also repeatedly aliased in the upper 4 MBytes of the GE's 16 MByte address space.

In normal operation for text drawing (done as a bitmap expanded Host-to-screen BitBlts), the CPUwrites exact amount of data to the Data Port for the current character, then starts on the next text character by writing to the Dst_XY register and finally writes data for the next text character. The current operation reads from the Data Port FIFO until its needs are met. Then the next BitBlt operation reads its data from the Data Port FIFO. To assure correct results, the software must write the correct number of 32-bit double words to the Data Port FIFO for all BitBlt operations.

The CPU can write to the Data Port at any time. If the GE is unable to accept any additional writes to the Data Port, the CPU is held off until the write command can be accepted. If a PCI master requests bus access when the CPU has been held off for a long period of time (128 clocks cycles), then the GE forces a CPU retry via the backoff (BOFF) mechanism.

4-MByte region 1 (memory mapped regs)

256KByte sub-region 0

31 27	26 24	23 22	21 18	17 12	11 0
00001	<GBASE>	0 - 1	0000	000000	<Index>

Table 12-6. GE and Data_Port access

12.9. REGISTER SPECIFICATION

The GE registers are listed in alphabetical order and defined below.

12.9.1. BACKGROUND COLOR REGISTER

This register contains the full-color value(s) that a ‘0’ bit is expanded to. This expansion is done for BitBlt operations with bitmap expansion specified in the Expand field of the ROP register, and for operations with the Source field of the ROP register set to CONSTANT_FILL.

For 8-bit pixels, only bits 7-0 are significant. For 16-bit pixels, only bits 15-0 are significant. For 24-bit pixels, only bits 23-0 are significant.

Background Access = 8400000h Regoffset = 0x004h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BC															
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BC															
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-0	**BC**	**Background Color.** This is the color to be used as the background when expanding bitmap ‘0’ values or when using CONSTANT_FILL as the source operand.

Programming notes

The content of this register is not altered by drawing operations.

12.9.2. CURSOR COORDINATE REGISTER

This register contains the address of the upper-left-hand corner of the cursor. To eliminate the cursor, its address should be set to a value large enough so that none of the cursor is on the displayed screen. Note that when set to (0,0), the entire cursor may be displayed on the upper-left hand corner of the display.

Cursor_XY Access = 8400000h Regoffset = 0x11Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv			CYUL												
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv			CXUL												
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-29	**Rsv**	**Reserved.**
Bits 28-16	**CYUL**	**CYUL**. The Y location of the upper-left-hand corner of the cursor.
Bits 15-13	**Rsv**	**Reserved.**
Bits 12-0	**CXUL**	**CXUL**. The X location of the upper-left-hand corner of the cursor.

Programming notes

To suppress cursor display, enter one more than the number of display scan lines into the Y field.

The contents of this register remain unaltered throughout drawing and display operations.

12.9.3. TOP OF DATA FIFO REGISTER

This write-only register is the port through which the CPU provides Host data.

Data_Port Access = 8400000h Regoffset = 0x804h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DP															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DP															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-0	**DP**	**Data_Port.**

Programming notes

The CPU can write to the Data Port at any time. If the GE is unable to accept any additional writes to the Data Port, the CPU will be held off until the write can be accepted.

Note that writing to this address is the same as writing to any double word between:

(128MBytes+(GBASE << 24) + 12 MBytes) to (128MBytes+(GBASE << 24) + 16 MBytes).

The Data FIFO is empty after reset.

12.9.4. DESTINATION OPERAND BASE ADDRESS REGISTER

This register specifies the starting DRAM linear address of the destination operand (aligned to a 32 Byte boundary).

Dst_Base Access = 8400000h Regoffset = 0x018h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv											DOB				
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOB															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-21	**Rsv**	**Reserved.**
Bits 20-0	**DOB**	**DstOp_Base.** Base DRAM linear address of the destination operand with 16 Byte alignement. Lower five Bytes are reserved and are set to ‘0’.

Programming notes

The contents of this register are not altered by drawing operations.

12.9.5. DESTINATION PITCH REGISTER

This register specifies the number of Bytes needed to advance from a pixel in one scan line of the Destination to the corresponding pixel in the next scan line. This value is always positive. The Y_dirfield of the Pixel_depth register controls how the operation advances to the next scan line.

Only a limited number of pitches are supported. The supported ones (in Bytes) are listed in the Src_pitch description.

This register can be accessed via 32-bit or 16-bit transfers.

Dst_Pitch Access = 8400000h Regoffset = 0x028h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv					DS3		DS2			DS1			DS0		
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-11	**Rsv**	**Reserved**.
Bits 10-9	**DS3**	**Dst_shift3.** These bits specify an amount to multiply Dst_XY.Y, this result along with the other shift results, is added to the Dst_base and Dst_XY.X to compute the DRAM linear address of the destination pixel. See Table 12-7. for the multiplication values that this field can specify.
Bits 8-6	**DS2**	**Dst_shift2**. See Dst_shift3, above.
Bits 5-3	**DS1**	**Dst_shift1**. See Dst_shift3, above.
Bits 2-0	**DS0**	**Dst_shift0.** See Dst_shift3, above.

Value	Shift0	Shift1	Shift2	Shift3
000	0	0	0	0
001	0	0	32 * Y	1024 * Y
010	0	64 * Y	64 * Y	2048 * Y
011	128 * Y	128 * Y	128 * Y	4096 * Y
100	256 * Y	256 * Y	256 * Y	n/a
101	512 * Y	512 * Y	512 * Y	n/a
110	1024 * Y	1024 * Y	0	n/a
111	2048 * Y	0	0	n/a

Table 12-7. DRAM address multiplication factor

Programming notes

The contents of this register are not altered by drawing operations.

12.9.6. DESTINATION OPERAND COORDINATE REGISTER

This register contains the coordinate address of the starting corner of the destination operand. The "starting" corner is controlled by the X_dir and Y_dst_dir fields of the Pixel_depth register.

Dst_XY Access = 8410000h Regoffset =

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DY															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DX															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-16	**DY**	**Dst_Y.** The unsigned Y coordinate of the starting corner of the destination operand.
Bits 15-0	**DX**	**Dst_X**. The unsigned X location of the starting corner of the destination operand. This value must be a multiple of Pixel_depth. The complete addressing of this register is described in Section 12.6.1. "Command Initiation".

Programming notes

The address of this register is also used to determine which of the BitBlt operations is to be performed (Simple BitBlt, Width-specified BitBlt or Height-specified BitBlt). Writing to this register initiates a graphics operation.

This register is double-buffered. While the GE is busy executing one operation, a new value may be safely written to this register for the next operation. If the buffer is full, then the CPU will be held off. This feature is implemented specifically to accelerate the Text and Line Segment operations.

The contents of this register are not altered by drawing operations.

12.9.7. FOREGROUND COLOR REGISTER

This register contains the full-color value(s) that a “1” bit is expanded to.

Foreground Access = 8400000h Regoffset = 0x034h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCI															
Default value after reset = FFFFFFFFh															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCI															
Default value after reset = FFFFFFFFh															

Bit Number	Mnemonic	Description
Bits 31-0	**FCI**	**Frg_Cl.** This is the color to be used as the foreground when expanding bit-map ‘1’ values.

Programming notes

This expansion is done for BitBlt operations with bitmap expansion specified in the Expand field of the ROP register.

For 8-bit pixels, only bits 7-0 are significant. For 16-bit pixels, only bits 15-0 are significant. For 24-bit pixels, only bits 23-0 are significant.

The contents of this register are not altered by drawing operations.

12.9.8. HEIGHT REGISTER

This register contains one less than the number of scan lines in the Source and Destination areas. The contents of this register will not change during the execution of a command.

Height Access = 8400000h Regoffset = 0x048h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-16	**Rsv**	**Reserved.**
Bits 15-0	**H**	**Height.** The value set in these bits must be one less than the height, in scan lines, of the source and destination areas.

Programming notes

This register can be accessed via 32-bit or 16-bit transfers.

This register can be loaded by writing to the Dst_XY register using one the Height-specified address alias. For this case, the Height register is loaded with the value in the Count field of the address described in 12.6.1. "Command initiation".

This register is double-buffered. While the GE is busy executing one operation, a new value may be safely written to this register for the next operation. If the Dst_XY double-buffered register is full, then the CPU will be held off. This feature is specifically implemented to accelerate the Text and Line Segment operations.

The contents of this register are not altered by drawing operations.

12.9.9. PATTERN BASE ADDRESS OPERAND REGISTER

This register contains the starting DRAM linear address of the Pattern operand, including the aligned base address, the first row to be displayed and a starting Byte number for 24-bit pixels.

Pattern Access = 8400000h Regoffset = 0x058h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv										PB					
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PB								Rsv			PXS		Rsv		
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.**
Bits 21-8	**PB**	**Pattern base.** These bits specify the starting DRAM physical address of the Pattern, operand, aligned to a 256 Byte boundary.
Bits 7-5	**Rsv**	**Reserved.**
Bits 4-3	**PXS**	**Pat_X_start.** For 24-bit pixels, these bits must be set to: (Dst_X / 8) modulo 3 where Dst_X is the Byte address of the first 24-bit pixel in the destination row. For all other pixel depths, the values must be set to “00”.
Bits 2-0	**Rsv**	**Reserved.**

Programming notes

The start of Pattern data must be aligned to a 256-Byte boundary. Advancing to the next Pattern data row will be done modulo 8 rows. Regardless of the number of Pixel_depth, the Pattern row is 32 Bytes long.

The Pattern register can be loaded with the address of the last row of Pattern data and the GE will wrap-around to the start of the pattern on the second row. Note that the Pattern register advances by increasing the address regardless of the X_dir, Y_src_dir or Y_dst_dir fields of the Pixel_depth register.

For further discussion of the Pattern Data, see Section 12.10.1. “Pattern Data”.

The contents of this register are not altered by drawing operations.

12.9.10. PIXEL DEPTH OPERAND REGISTER

This register contains the number of Bytes in a pixel, and bits that control the direction of Screen-to-screen BitBlts.

Pixel_Depth Access = 8400000h Regoffset = 0x07Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv								Y	Y	X	Rsv			PD	
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-8	**Rsv**	**Reserved.**
Bits 7	**Y**	**Y_src_dir.** When this bit is set to ‘0’, source pixels advance from upper scan lines to lower scan lines (from smaller linear to larger linear addresses). Setting this bit to ‘1’ reverses the direction of BitBlt source operations. This bit should be set to ‘0’ for all operations other than reverse-direction non-bit-map-expanded screen-to-screen BitBlts.
Bits 6	**Y**	**Y_dst_dir.** When this bit is set to ‘0’ destination pixels advance from upper scan lines to lower scan lines (from smaller to larger linear addresses). Setting this field to ‘1’ reverses the direction of BitBlt destination operations. This field should be set to ‘0’ for all operations other than reverse-direction non-bitmap-expanded screen-to-screen BitBlts.
Bit 5	**X**	**X_dir.** When this bit is set to ‘0’, pixels advance from left to right, and when set to ‘1’ they advance from right to left. This field can be set to ‘1’ only for Screen-to-screen BitBlts and horizontal scan line fills.
Bits 4-2	**Rsv**	**Reserved.**
Bits 1-0	**PD**	**Pixel depth.** The only supported values for this field are shown in Table 12-8.

Bit 1	Bit 0	Pixel Depth
0	0	1 Byte per pixel
0	1	2 Bytes per pixel
1	0	3 Bytes per pixel
1	1	4 Bytes per pixel

Table 12-8. Supported Pixel depth values

Programming notes

Note that the 4th Byte of 4-Byte pixels is not used in the display, but is processed by drawing operations. Zeros should be written to this 4th Byte to preserve compatibility with future versions of this architecture.

This register can be accessed via 32-bit or 16-bit transfers.

The contents of this register are not altered by drawing operations.

12.9.11. RASTER OPERATION REGISTER

This register contains the ROP code to be applied during processing a pixel, enables bitmap expansion, selects transparent modes, and controls the source operand. This is summarise in Table 12-9.

ROP Access = 8400000h Regoffset = 0x08Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S		P	D	DM	Rsv										
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D	D	S	S	P	E	P	B	R							
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-30	**S**	**Source.** These bits determine the SRC operand. Possible values are shown in Table 12-10. This field MUST be set to CONSTANT_FILL if the Raster Operation requires no SRC operand (such as in inverting the destination as well as constant fills). Failure to set this field correctly can result in a degradation of performance. Note that CPU writes to the Data Port will complete without error, and the data will be ignored unless the Source field is set to HOST.
Bit 29	**P**	**PAT present.** This bit is set to ‘1’ if a pattern data is to be used during the operation.
Bit 28	**D**	**DST present** This bit is set to ‘1’ if destination data is to be read during the operation.
Bit 27	**DM**	**Diagnostic Mode,** For normal operation this bit should be set to ‘0’. When set to ‘1’, GE register reads will be done from an alternative path for diagnostic verification.
Bits 26-16	**Rsv**	**Reserved.**
Bit 15	**D**	**DST transparency mode.** When this is set to ‘1’, pixels are selectively modified based upon a comparison of the DST data from the frame buffer vs. the DST transparency compare register. The results of the comparison are interpreted based upon the DST transparency match bit. Note that the DST present bit must also be set to ‘1’ when this bit is set. This mode is not valid when the pixel depth is 3 Bytes per pixel.
Bit 14	**D**	**DST transparency match.** This mode applies only when DST transparency mode is set. When this bit is set to ‘1’, pixels with DST data that match the DST transparency compare register will be modified. When it is set to ‘0’, pixels with DST data that do not match the DST transparency compare register will be modified.

Bit Number	Mnemonic	Description
Bit 13	S	**SRC transparency mode.** When this is set to ‘1’, pixels are selectively modified based upon a comparison of the SRC data vs. the SRC transparency compare register. The results of the comparison are interpreted based upon the SRC transparency match bit. This mode is only meaningful when using non-bitmap screen or host data as the source. Transparency for bitmap source data should not use this mode, but rather the SRC bitmap transparency mode. This mode is not valid when the pixel depth is 3 Bytes per pixel.
Bit 12	S	**SRC transparency match.** Applies only when SRC transparency mode is set. When this bit is set to ‘1’, pixels with SRC data that match the SRC transparency compare register will be modified. When this is set to ‘0’, pixels with SRC data that do not match the SRC transparency compare register will be modified.
Bit 11	P	**Packed.** If set to ‘1’, the source will be read in packed mode. Effectively, the source is viewed as a continuous stream of data. At the end of a destination scan line, any data remaining in the last-used source Dword is applied to the start of the next destination scan line. When this bit is set to ‘0’, any remaining source data is discarded at the end of a destination scan line. New source data is read from the next source scan line to apply to the start of the next destination scan line.
Bit 10	E	**Expand.** If set to ‘1’, the bitmap expansion will be enabled and source data from screen or host is assumed to be bitmap data. If set to ‘0’, source data is assumed to be color data with the depth specified in the Pixel_depth register.
Bit 9	P	**PAT transparency mode.** When this bit is set to ‘1’, pixels are selectively modified based upon the value of corresponding pattern data. Pattern Bytes that are set to zero are not modified. Note that the PAT present bit must also be set to ‘1’ when this bit is set.
Bit 8	B	**Bitmap transparency mode.** When this bit is set to ‘1’, pixels are selectively modified based upon the pre-expanded bitmap value. Pixels with corresponding bitmap values of zero are not modified. Pixels with corresponding bitmap values of one are written with the foreground value. Note that the expand bit must also be set when using this mode.
Bits 7-0	R	**ROP**, the raster operation used when computing a pixel result value.

Bits	Function
31:30	SRC operand type
29	Use PAT operand
28	Use DST operand
27	GE Diagnostic mode
26:16	Unused/Reserved
15	DST transparency mode
14	DST transparency match
13	SRC transparency mode
12	SRC transparency match
11	Packed SRC data
10	SRC bitmap expansion
9	PAT transparency mode
8	SRC bitmap transparency
7:0	Raster operation code

Table 12-9. Summary of ROP Functions

Bit 31	Bit 30	Function	Source
0	0	CONSTANT_FILL	Background color register
0	1	SCREEN	screen or frame buffer
1	0	HOST	host CPU
1	1	Reserved	

Table 12-10. Detail of SRC operand functions

Programming notes

The contents of this register are not altered by drawing operations.

12.9.12. SOURCE BASE ADDRESS OPERAND REGISTER

This register specifies the starting DRAM linear address of the source operand (aligned to a 32 Byte boundary).

Src_Base Access = 8400000h Regoffset = 0x098h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv										SB					
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SB											Rsv				
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.**
Bits 21-5	**SB**	**SrcOp_Base.** Base linear address of the source operand.
Bits 4-0	**Rsv**	**Reserved.** These bits are set to ‘0’.

Programming notes

The contents of this register are not altered by drawing operations.

12.9.13. SOURCE PITCH OPERAND REGISTER

This register specifies the number of Bytes needed to advance from a pixel in one scan line of the source to the corresponding pixel in the next scan line. This value is always positive. The Y_src_dir field of the Pixel_depth register controls how the operation advances to the next scan line.

Only a limited number of pitches are supported. The supported pitches (in Bytes) are described in Section 12.6. "VGA operand frame buffer adresses".

Src_Pitch Access = 8400000h Regoffset = 0x0ACh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv					S3		S2			S1			S0		
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-11	**Rsv**	**Reserved.**
Bits 10-9	**S3**	**Src_shift3.** These bits specify an amount to multiply Src_XY.Y, this result along with the other shift results, is added to the Src_base and Src_XY.X to compute the DRAM linear address of the source pixel. See the Table 12-11. below for the multiplication values that this field can specify.
Bits 8-6	**S2**	**Src_shift2.** See Src_shift3, above.
Bits 5-3	**S1**	**Src_shift1.** See Src_shift3, above.
Bits 2-0	**S0**	**Src_shift0.** See Src_shift3, above.

Value	Shift0	Shift1	Shift2	Shift3
000	0	0	0	0
001	0	0	32 * Y	1024 * Y
010	0	64 * Y	64 * Y	2048 * Y
011	128 * Y	128 * Y	128 * Y	4096 * Y
100	256 * Y	256 * Y	256 * Y	n/a
101	512 * Y	512 * Y	512 * Y	n/a
110	1024 * Y	1024 * Y	0	n/a
111	2048 * Y	0	0	n/a

Table 12-11. Scr_shift3 multiplication factors

Programming notes

This register can be accessed via 32-bit or 16-bit transfers.

The contents of this register are not altered by drawing operations.

12.9.14. SOURCE COORDINATE REGISTER

This register contains the coordinate address of the starting corner of the source operand.

Src_XY Access = 8400000h Regoffset = 0x0BDh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SY															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SX															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-16	**SY**	**Src_Y.** The source operand starting corner of the unsigned Y coordinate.
Bits 15-0	**SX**	**Src_X.** The source operand starting corner of the unsigned X location. When bitmap expansion is not enabled, this is a Byte address. When in bitmap expansion is enabled, this is a bit address.

Programming notes

The “starting” corner is controlled by the X_dir and Y_src_dir fields of the Pixel_depth register.

This register is double-buffered. While the GE is busy executing one operation, a new value may be safely written to this register for the next operation. If the Dst_XY double-buffered register is full, then the CPU will be held off.

The Y field and all but the lower 3 bits (5 when bitmap expansion is enabled) of the X field are ignored during Host-to-screen BitBlts.

The contents of this register are not altered by drawing operations.

12.9.15. STATUS REGISTER

Status Access = 8400000h Regoffset = 0x908h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GB	PB	Rsv													
0		Default value after reset = undefined													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bit 31	**GB**	**GE_Busy.** This read/write bit is set to ‘1’ when the GE is busy, GE register accesses that are done when this bit is set may result in the CPU being held for the duration of the current operation.
Bit 30	**PB**	**Pending Busy.** This read-only bit is set to ‘1’ when the Dst_XY pending register has data in it. If GE writes to that registers when this bit is set it may result in the CPU being held for the duration of the current operation. GE register reads always return data without holding the CPU, but the data returned from the read may not be valid. The Status register may be read at any time and the operation will return valid data. Note that Pending Busy implies Busy, that is the Pending Busy field can be set to ‘1’ only if the Busy field is also set to ‘1’.
Bits 29-0	**Rsv**	**Reserved.** These may read as one or zero.

12.9.16. WIDTH REGISTER

This register contains the one Byte width less than the destination operand.

This register can also be loaded by writing to the Destination register using one of the Text or Line Segment commands. For these cases, the Width register is loaded with the value in the Count field of the address described in Section 12.6.1., "Command Initiation".

This register is double-buffered. While the GE is busy executing one operation, a new value may be safely written to this register's Width field for the next operation. If the Dst_XY double buffered register is full, then the CPU will be held off.

Width Access = 8400000h Regoffset = 0x0C8h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-16	**Rsv**	**Reserved.**
Bits 15-0	**W**	**Width.** These bits should be set to one less than the number of Bytes across the destination area. This value should be a multiple of Pixel_depth, because only the number of Bytes specified in this field will be modified.

Programming notes

This register can be accessed via 32-bit or 16-bit transfers.

The contents of this register are not altered by drawing operations.

12.9.17. EXTRA USE REGISTER

This register contains 32 bits of data that software can read from and write to.

This register has no effect on any drawing operation or display.

Xtra Access = 8400000h Regoffset = 0x0D4h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-0	**D**	**Data for user software use.**

Programming notes

The contents of this register are not altered by drawing operations.

12.9.18. SRC TRANSPARACENCY COMPARE REGISTER

This 32-bit register contains the pixel value used for comparison in SRC transparency mode. For pixels depths of 1 Byte per pixel, the pixel value must be replicated in all four Bytes of this register. For pixel depths of 2 Bytes per pixel, the pixel value must be replicated in the upper and lower 16 bits of this register.

SRC_Transparency Access = 8400000h Regoffset = 0xECh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D								C							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B								A							

1 byte per pixel	**Colour replicated in A, B, C and D**
2 bytes per pixel	**Colour replicated across A, B and C, D**
3 bytes per pixel 24 bit colour	**Colour replicated across A, B and C**
4 bytes per pixel 32 bit colour	**Colour replicated across A, B, C and D**

12.9.19. DST TRANSPARENCY COMPARE REGISTER

This 32-bit register contains the pixel value used for comparison in DST transparency mode. For pixels depths of 1 Byte per pixel, the pixel value must be replicated in all four Bytes of this register. For pixel depths of 2 Bytes per pixel, the pixel value must be replicated in the upper and lower 16 bits of this register.

DST_Transparency Access = 8400000h Regoffset = 0xFCh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D								C							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B								A							

1 byte per pixel	**Colour replicated in A, B, C and D**
2 bytes per pixel	**Colour replicated across A, B and C, D**
3 bytes per pixel 24 bit colour	**Colour replicated across A, B and C**
4 bytes per pixel 32 bit colour	**Colour replicated across A, B, C and D**

12.9.20. NOTES ON: Interactions Between Blt Operations and VGA Framebuffer Accesses

The GE performs two major classes of operations: BitBlts and standard VGA Framebuffer accesses. These two types of operations share resources in the hardware. This imposes certain requirements on driver software.

The state of all standard VGA registers is unchanged by BitBlt and extended register reads/writes with the exception of the CR22 data latch. The state of this register is undefined after a BitBlt.

The state of all extended registers is unchanged by VGA read/write operations.

Between a BitBlt operation and a VGA read/write operation, the software must ensure that no BitBlt operation is in progress by means of the Status register.

Before performing any VGA read/write operations, the software must ensure the Foreground register has the value FFFFFFFFh and the background register has the value 00000000h. These are also the reset values of these registers.

Between a VGA write operation and a BitBlt operations, the software must ensure the VGA write pipeline is flushed by performing a VGA read operation.

12.10. GE OPERATIONS

12.10.1. PATTERN DATA

If the ROP register value specifies that pattern data is used in the computation of the destination results, then one row of the pattern data is read at the start of each scan line processed. This row of data is repeatedly applied to the result computation across scan line. The Pattern register points to the start of an 8-pixel-by-8-pixel color area that is aligned to the destination. The GE does not perform any horizontal alignment to the pattern data.

When the pixel depth is 3 Bytes, the least significant three bits of the Pattern register must indicate which Byte starts the pattern row. This field should be set to:

(Dst_X / 8) modulo 3

where Dst_X is the Byte address of the first 24-bit pixel in the destination row. (The same value that is written to the X field of the Dst_XY register).

Bitmap patterns are not directly supported. To use a bitmap pattern, first allocate off-screen frame buffer memory for a color version of the pattern. Then set up the GE to perform a Host-to-screen BitBlt with bitmap expansion into this allocated memory. The bitmap pattern is then written to the Data Port. The expanded pattern can now be used by pointing the Pattern register to the allocated memory.

12.10.2. BITMAT CONSIDERATIONS

Screen-to-screen and Host-to-screen operations can optionally expand single-bit-per-pixel bitmaps into color pixels. Each ‘1’ bit is replaced by the contents of the Foreground color register and each ‘0’ bit is replaced by the contents of the Background color register.

Bitmaps from the frame buffer (during Screen-to-screen BitBlts) must be aligned on a quad-word (64-bit) boundary. Bitmaps from the Host can be aligned on a double-word (32-bit) boundary. Leading bits of the bitmaps may be skipped by setting the least significant bits of the X field of the Src_XY register to the number of bits in the Byte to be ignored. When in bitmap expansion mode, the X field of the Src_XY address can be thought of as a bit address instead of a Byte address. For Host-to-screen bitmap expanded BitBlts only the least significant 5 bits of the Src_XY.X register are significant. The first bit after those skipped will then be aligned to the first destination pixel.

For bitmap expansion, the X_dir must be ‘0’. The result for a bitmap expansion BitBlt with X_dir set to ‘1’ is not defined.

With the X_dir field of the Pixel_depth register set to ‘0’, the bitmap is considered to start at the least significant end of the first quad-word and continues towards the most significant end of the quad-word and then to higher memory addresses. The first bit of a quad-word is bit 7 of Byte 0 and the last bit is bit 0 of Byte 7.

12.10.3. BITBIT OPERATIONS

Using the GE’s BitBlt commands it is possible to implement the following six operations:

1) Rectangular Fill

2) Screen-to-screen BitBlt

3) Host-to-screen BitBlt

4) Packed Text

5) Microsoft Font Text

6) Line Segments

12.10.4. RECTANGULAR FILL

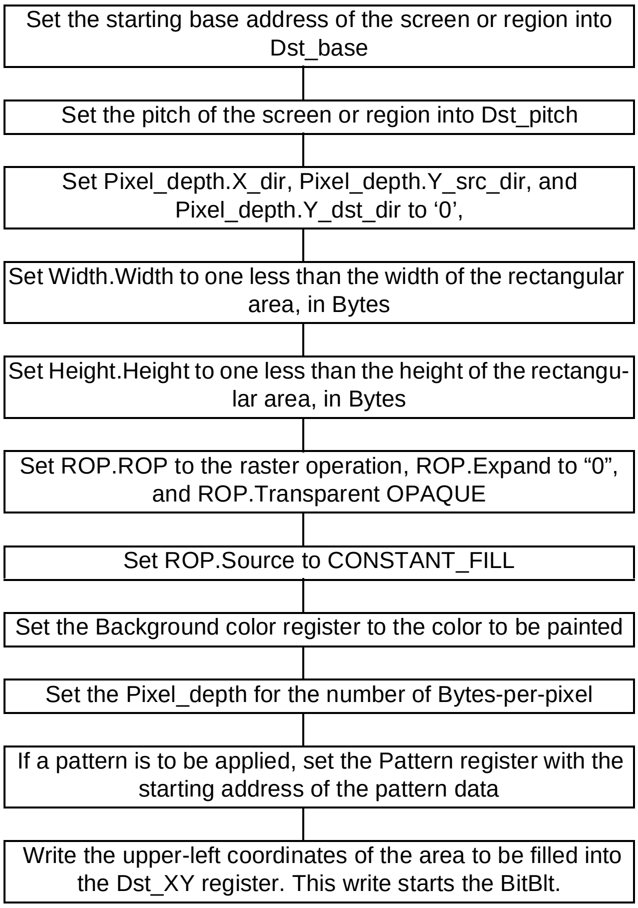
A rectangular fill operation is used to fill rectangular areas in the frame buffer with solid or patterned colors. The function performed during the fill operation is:

ROP ((Background), (Pattern), (Destination)) -> (Destination)

The specified rectangle is filled with the contents of the Background color register, the pattern data and the existing destination contents, as modified by the ROP. The CPU provides the rectangle’s upper-left (Dst_XY) coordinates, the width and the height of the rectangle. Destination and pattern data can be anywhere in the frame buffer. ROP may be any of the 256 standard raster operations.

The Rectangular Fill operation is optimized to run at the memory bandwidth.

To perform a rectangular fill (except for the last write, order is unimportant):



Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Background color, and Pixel_depth. These will not need to be written each time.

The implementation of this operation is performance optimized and can drive the DRAM buffer at its full bandwidth. Thus, result pixels are computed in groups of 32 bits, to allow one 64-bit result every 2 video domain clock cycles. As the memory subsystem supports separate Byte-write enables, the first and last 64-bit write in each scan line can be performed without requiring a read-modify-write cycle.

12.10.5. SCREEN-TO-SCREEN BITBIT

The Screen-to-screen BitBlt operation is used to copy data from one rectangle in the frame buffer (either on-screen or off-screen areas) to another with the identical geometry. The pixel depth of the source region must match that of the destination region, or it may be a bitmap (if bitmap expansion is specified by setting ROP.Expand).

The function performed during the BitBlt operation is:

ROP ((Source), (Pattern), (Destination)) -> (Destination)

If these rectangular areas are overlapping, then the direction of the BitBlt must be carefully selected:

* *Source region is below the destination:*

> BitBlt should be done from the upper-left-hand corner and progress downwards,

> the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register should be “0”.

* *Source region is above the destination:*

> BitBlt should be done from the lower-right-hand corner and progress upwards,

> the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register should be “1”.

The Source, destination and pattern data can be anywhere in the frame buffer. The ROP may be any one of the 256 standard raster operations.

To perform a Screen-to-screen BitBlt (except for the last write, order is unimportant):

1. Set the starting base address of the destination screen or region into Dst_base
2. Set the pitch of the destination screen or region into Dst_pitch
3. Set the starting base address of the source region into Src_base
4. Set the pitch of the source region into Src_pitch
5. Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to ‘0’ or to ‘1’, depending upon direction of the BitBlt
6. Set Width.Width to one less than the width of the destination rectangular area, in Bytes
7. Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values
8. Set ROP.Source to SCREEN
9. If Color Transparency mode is set, the set Background color register to the color to be made transparent
10. If a pattern is to be applied, set the Pattern register with the starting address of the pattern data
11. If ROP.Expand is set, then also set the Foreground and Background color registers
12. Write the starting coordinates of the source area to be copied into the Src_XY register
13. Write the starting coordinates of the destination area into the Dst_XY register. This write starts the BitBlt.

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

The implementation of this operation is performance optimized and can drive the DRAM buffer at its full bandwidth during constant fills. Thus, result pixels are computed in groups of 32 bits, to allow one 64-bit result every 2 graphics clock domain cycles. As the memory subsystem supports separate Byte-write enables, the first and last 64-bit write in each scan line can be performed without requiring a read-modify-write cycle.

12.10.6. HOST-TO-SCREEN BITBIT

The Host-to-screen BitBlt is used to copy data from the Host CPU to the frame buffer (either on-screen or off-screen areas). Note that if the CPU has built a rectangle in the frame buffer memory area with the Host data, then the Screen-to-Screen BitBlt operation can be used instead of this operation.

The pixel depth of the Host data must match that of the Destination region, unless it is a bitmap (if bitmap expansion is specified).

The function performed during the BitBlt is:

ROP((Host),(Pattern),(Destination))->(Destination).

The host area data is supplied by the CPU, which writes its data into the Data Port. The destination and pattern data can be anywhere in the frame buffer.

ROP may be any one of the 256 standard raster operations.

The CPU specifies the number of least significant Bytes of the first double-word that should be discarded, via the least significant 3 bits of the X field in the Src_XY register. The GE then merges Bytes of two double-words at a time, in order to build a double-word to operate on. For example, if the X field was set to 3, then the last Byte of the first double-word and the first three Bytes of the second double-word would be combined to form the first Host data double-word:

	Double Word 1				Double Word 0			
Byte:	7	6	5	4	3	2	1	0

Byte:	6	5	4	3

The CPU must provide the number of words required for Height * Width pixels. At the end of a scan line, the GE will discard the excess Host Bytes or bits that may be left in the last double-word and advance to the next scan line, unless Src_pitch is set to 0. In this case, data for adjacent scan lines are contiguous in the host data stream.

To perform a Host-to-screen BitBlt (except for the last write, order is unimportant):

Set the starting base address of the destination screen or region into Dst_base

Set the pitch of the destination screen or region into Dst_pitch

Set Src_pitch to 0 if packed source data or to a non-zero value, otherwise

Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to ‘0’

Set Width.Width to one less than the width of the destination rectangular area, in Bytes

Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values

Set ROP.Source to HOST

If Color Transparency mode is set, set the Background color register to the color to be made transparent

Set the Pixel_depth for the number of Bytes-per-pixel

If a pattern is to be applied, set the Pattern register with the starting address of the pattern data

If ROP.Expand is set, then also set the Foreground and Background color registers

Set Src_XY to indicate alignment of the Host data (only the least 3 significant bits are important)

Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.

The image data is then written to the GE’s Data Port, if the GE is unable to accept additional data the CPU will be held off (invisibly to the software)

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

12.10.7. PACKED TEXT

The Packed Text operation is used to efficiently expand packed bitmap fonts into full color representations in the frame buffer (either on-screen or off-screen areas). This operation is implemented as a Host-to-screen BitBlt with bitmap expansion and packed source data. The next section discusses how to handle Microsoft Font Text operations.

The function performed during the Packed Text operation is:

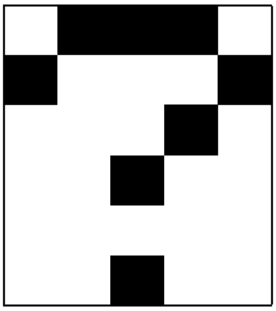
ROP ((Host), (Pattern), (Destination)) -> (Destination).

The Host packed bitmap data is supplied by the CPU via writes to the Data Port and is organized as double-words containing 32 bits of information. Each bit corresponds to a pixel. This data is expanded into Background and Foreground colors, unless the bitmap expansion transparent mode is on. If the transparent mode is set, then Host data bits of ‘0’ suppress any changes to the corresponding destination pixels. The Destination and Pattern can be anywhere in the frame buffer.

ROP may be any one of the 256 standard raster operations.

In a standard bitmap, the start of each scan line is aligned to a pitch-specified boundary. This is acceptable for wide bitmaps, however text font bitmaps are usually not very wide. To increase the amount of information provided to the GE per Host write, the Text operands are bit-packed. Each 32-bit write contains only useful font data, except possibly for the trailing bits of the last write.

For example, question mark character might appear in a fictitious font as:



Or in Binary form:

0	1	1	1	0
1	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	0	0	0
0	0	1	0	0

This would appear in memory as in Table 12-12.

In this example, the entire character bitmap fits into a single 32-bit double-word. This is a big savings over having to possibly send one 32-bit double-word for each font row. Note that 2 bits of don’t cares exist at the (top) end of the double word. Since this character is 5 bits wide and 6 lines high, it only needs 30 bits of storage. The remaining 2 bits will not be displayed.

After setting up the registers, the CPU writes the Host data, in 32-bit quantities, to the Data Port.

If a Pattern is applied to the text operation, a row of the pattern data will be read at the start of each character scan line.

top line					bottom line
01110	10001	00010	00100	00000	00100
Increasing memory addresses ------>					

Table 12-12. Bit representation

Breaking this up into Bytes see Table 12-13.

	01110100	01000100	01000000	000100XX
Byte	0	1	2	3

Table 12-13. Byte representation

Breaking this up into a double-word as in Table 12-14.

	000100XX	01000000	01000100	01110100
Byte	3	2	1	0
Word	0		1	

Table 12-14. Double word representation

To perform a Packed Text BitBlt (except for the last write, order is unimportant):

- Set the starting base address of the destination screen or region into Dst_base
- Set the pitch of the destination screen or region into Dst_pitch
- Set the ROP.Packed to 1 for packed source data
- Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, Pixel_depth.Y_dst_dir to ‘0’
- Set Height to the height of the destination character, in scan lines
- Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values
- Set ROP.Source to HOST
- Set the Pixel_depth for the number of Bytes-per-pixel
- If a pattern is to be applied, set the Pattern register with the starting address of the pattern data
- set the Foreground and Background color registers
- Set Src_XY to indicate alignment of the Host data (only the least 5 significant bits are important)
- Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.
- The character font data is then written to the GE’s Data Port, if the GE is unable to accept additional data the CPU will be held off (invisibly to the software)

To draw the next character, its starting address (taking into account inter-character spacing) is written to the Dst_XY register, along with that character’s width encoded into the address of the Dst_XY register. This write can be done even if the GE is busy, as the Destination and Width registers are double buffered. The CPU then writes all the bitmap data that corresponds to the second character, the third Dst_XY/ Width, the third bitmap data, etc.

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

12.10.8. MICROSOFT FONT TEXT

Microsoft fonts (consisting of 8-bit strips of a character) can be handled as simple 8-pixel-wide Host-to-screen BitBlts with bitmap expansion, but no packed data. The last strip of a character is handled in a different manner. The background color for the last strip is first filled into its rectangular area. Then the strip data is drawn in transparent mode with the unused bits filled with zeros.

12.10.9. LINE SEGMENTS

The line segment operations are used to draw horizontal or vertical line segments. The segments are runs of pixels that start from a specified coordinate address (via the Dst_XY register) and whose length is specified in the address used when writing to the Dst_XY register.

The function performed during the line draw is:

ROP ((Background), (Pattern), (Destination)) -> (Destination)

ROP may be any one of the 256 standard raster operations.

Simple and complex curves can be efficiently drawn. The software on the CPU must generate all points or scan lines to be drawn and then use the GE to draw the line segments.

Two different types of line segments are supported: horizontal and vertical. For horizontal line segments, the Height register should be programmed to ‘0’, to indicate a single pixel high line. The length of the line segment (in Bytes) will then be stored into the Width register when the Dst_XY register is written to. (The length is encoded into the Count field of the Dst_XY register’s address.) For vertical lines, the Width register should be programmed to one less than the number of Bytes per pixel, to indicate a single pixel wide line. The length of the line segment is stored into the Height register when the Dst_XY register is written to.

It is possible to draw thicker line segments, by programming the Height register (for horizontal segments) or the Width register (for vertical segments) to other values.

For horizontal line segments, the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register must be set to “0”.

The Background color register should be set to the color of the line segment to be drawn.

12.11. CURSOR SUPPORT

The GE supports a 64x64x2 cursor. The cursor is actually two 64x64x1 arrays: an AND array and an XOR array. For any given pixel that is within the cursor’s active region, the displayed pixel depends on the frame buffer’s pixel, the AND array value, the XOR array value and the Cursor_color0 and Cursor_color1 registers as shown in Table 12-15.

The AND array is stored in off screen memory, starting at Cursor. The XOR array is stored in off screen memory starting at (Cursor + 512). Two 64-bit on-chip registers hold one scan line of each of these arrays. Before a scan line that possibly includes a cursor is displayed, these two registers are loaded from the appropriate off-screen locations.

Note that for 8-bit and 16-bit pixel depths, the above cursor operation is performed AFTER the data has been expanded by the color look-up-table (LUT). Thus, the Inverted Frame Buffer Pixel, is the complement of the full-color pixel that would otherwise be displayed.

The cursor address (Cursor_XY) refers to the upper-left-hand corner of the cursor and specifies the distance, in pixels, from the upper-left-hand corner of the screen. So, if the cursor address were to be set to (0,0), then the entire cursor could be displayed in the upper-left-hand corner of the screen. The cursor's active region thus may extends from:

(Cursor_XY.X, Cursor_XY.Y)

to

(Cursor_XY.X + 63, Cursor_XY.Y + height - 1)

as controlled by the Cursor Height register (CR29).

Note: To suppress cursor display, enter one more than the number of display scan lines into the Y field.

AND Value	XOR Value	Displayed Pixel
0	0	Cursor_color0
0	1	Cursor_color1
1	0	Frame Buffer Pixel
1	1	Inverted Frame Buffer Pixel

Table 12-15. Cursor Arrays

12.11.1. CURSOR HEIGHT REGISTER (RW)

CR29 Access = 0x3X4h/0x3X5h Regoffset = 029h

7	6	5	4	3	2	1	0
C XOR	CH						
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**C XOR**	**Cursor XOR Pre/Post Look Up Table.** When this bit is set to one, the graphics cursor XOR operation is performed before the look up table. The default behavior, when this bit is set to zero, is for the XOR operation to happen after the look up table. This is correct for 15, 16, 24 bit per pixel modes but not 8bpp.
Bits 6-0	**CH**	**Cursor height.** This field represents the vertical extent of the graphics cursor in scan lines. Setting this to zero effectively turns the graphics cursor off. Values greater than 40h (decimal 64) are meaningless and produce unpredictable results.

Programming notes

Note: there is no cursor width register - the width is always 64 pixels. If a narrower cursor is required, pad the bitmap on the right with transparent cursor color (pad the AND plane with ‘1’s on the right and the XOR plane with ‘0’s).

12.11.2. CURSOR COLOR 0 REGISTER A (RW)

CR2A Access = 0x3X4h/0x3X5h Regoffset = 02Ah

7	6	5	4	3	2	1	0
CC 0 R							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CC 0 R**	**Cursor Color 0 Red.** These bits are the red component of cursor color 0.

12.11.3. CURSOR COLOR 0 REGISTER B (RW)

CR2B Access = 0x3X4h/0x3X5h Regoffset = 02Bh

7	6	5	4	3	2	1	0
CC 0 G							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CC 0 G**	**Cursor Color 0 Green.** These bits are the green component of cursor color 0.

12.11.4. CURSOR COLOR 0 REGISTER C (RW)

CR2C Access = 0x3X4h/0x3X5h Regoffset = 02Ch

7	6	5	4	3	2	1	0
CC 0 B							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CC 0 B**	**Cursor Color 0 Blue.** These bits are the blue component of cursor color 0.

12.11.5. CURSOR COLOR 1 REGISTER A (RW)

CR2D Access = 0x3X4h/0x3X5h Regoffset = 02Dh

7	6	5	4	3	2	1	0
CC 1 R							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CC 1 R**	**Cursor Color 1 Red.** These bits are the red component of cursor color 1.

12.11.6. CURSOR COLOR 1 REGISTER B (RW)

CR2E Access = 0x3X4h/0x3X5h Regoffset = 02Eh

7	6	5	4	3	2	1	0
CC 1 G							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CC 1 G**	**Cursor Color 1 Green.** These bits are the green component of cursor color 1.

12.11.7. CURSOR COLOR 1 REGISTER C (RW)

CR2F Access = 0x3X4h/0x3X5h Regoffset = 02Fh

7	6	5	4	3	2	1	0
CC 1 B							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CC 1 B**	**Cursor Color 1 Blue.** These bits are the blue component of cursor color 1.

12.11.8. GRAPHICS CURSOR ADDRESS REGISTER 0 (RW)

CR30				Access = 0x3X4h/0x3X5h			Regoffset = 030h
7	6	5	4	3	2	1	0
CAA							Rsv
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-1	**CAA**	**Cursor AND Address Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the cursor's AND mask. The cursor's XOR mask begins at this address + 512. This memory must be aligned on a 1 KByte boundary. For a discussion of DRAM linear addresses, see Section 12.6.
Bit 0	**Rsv**	**Reserved.** This bit should be written as zero.

Programming notes

Note that the cursor bitmap is ordered such that the top left hand comer of the cursor is represented by bit of the Byte addressed by this field (AND) and bit 7 of the Byte at 512 plus this address (XOR plane). The next pixel right is represented by bit 6 of pleses Bytes and so on until the bottom right hand pixel is represented by bit 0 of the Byte local ed at this address plus 511 (AND) and bit 0 of the Byte at 1023 plus this address.

12.11.9. GRAPHICS CURSOR ADDRESS REGISTER 1 (RW)

CR31 Access = 0x3X4h/0x3X5h Regoffset = 031h

7	6	5	4	3	2	1	0
CAA							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0	**CAA**	**Cursor AND Address Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the cursor's AND mask. For a discussion of DRAM linear addresses, see Section 12.6.

12.11.10. GRAPHICS CURSOR ADDRESS REGISTER 2 (RW)

CR32 Access = 0x3X4h/0x3X5h Regoffset = 032h

7	6	5	4	3	2	1	0
Rsv					CAA		
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-3	**Rsv**	**Reserved.** These bits should be written as zero.
Bits 2-0	**CAA**	**Cursor AND Address Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the cursor's AND mask. For a discussion of DRAM linear addresses, see Section 12.6.

12.12 GRAPHICS CLOCK REGISTERS

The GCLK is used to the Graphics Engine operations

12.12.1 GCLK CONTROL REGISTER 0

GLCK00 Access = 022h/023h Regoffset = 40h

7	6	5	4	3	2	1	0
Uns	4BM				8BN		
Default value after reset = 0x5Bh							

Bit Number	Mneumonic	Description
Bit 7	**Uns**	This is unused.
Bits 6-3	**4BM**	This the 4-bit M (divisor) value of the Graphics clock synthesizer.
Bits 2-0	**8BN**	These are bits 7-5 of the 8 bit N (mulitiplier) of the Graphics clock synthesizer.

Programming notes

This register defaults to 0x5B at reset. This value when combined with the default value of the other half of this pair results in a graphics clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

12.12.2 GCLK CONTROL REGISTER 1 LK01 INDEX 41

GLCK01 Access = 022h/023h Regoffset = 41h

7	6	5	4	3	2	1	0
3BP0	8BN					3BP1	
Default value after reset = 0xECh							

Bit Number	Mneumonic	Description
Bit 7	**3BP0**	This is the bit 0 of the 3-bit P (exponent) value of the Graphics clock synthesizer.
Bits 6-2	**8BN**	These are bits 4-0 of the 8-bit N (multiplier) value of the Graphics clock synthesizer.
Bits 1-0	**3BP1**	These are bits 2-1 of the 3-bit P (exponent) value of the Graphics clock synthesizer.

Programming notes

This register defaults to 0xEC at reset. This value when combined with the default value of the other half of this pair results in a graphics clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

12.13 UPDATE HISTORY FOR GRAPHICS ENGINE CHAPTER

The following changes have been made to the Graphics Engine Chapter.

Section	Change	Text
12.12	Added	GCLK register descriptions

The following changes have been made to the Graphics Engine Chapter.

Section	Change	Text
12.1.	Replaced	"~~above~~" With "between"
12.2.	Replaced	"~~displayed~~" With "displayed in order to be compatible with future versions of the GE."
12.2.	Removed	"~~Except for the VGA registers~~,"
12.2.	Removed	"~~VGA registers may be written to via byte, word or double-word accesses~~."
14.6	Removed	~~"The GE fully supports VGA operations. The standard two read modes and four write modes of the VGA frame buffer are implemented. While, there is no "GE mode" bit that places the drawing operations in a VGA mode, the Foreground and Background color registers must be set to all '1's and all '0's, respectively, before issuing standard VGA write operations."~~
12.5.	Removed	~~"Addressing for reads and writes~~ ~~Host addresses are mapped into display memory addresses according to the VGA addressing mode selected.~~ ~~Multiple byte reads and writes are broken into single byte operations by the GE. For multi-byte reads, all of the bytes are sequenced through by the GE which performs each read and store the resulting byte in correct byte of an VGA read register. After the last read is done, the GE will provide the CPU with the requested data.~~ ~~For multi-byte writes, all the GE sequences through the CPU data bytes, performing a 32-bit write to memory for each byte. No optimization is done by the GE to merge 32-bit Dword writes. (Such optimization would only be possible in the Chain-4 addressing modes.)"~~
12.5.1.	Removed	~~"Read Mode 0~~ ~~Read data is read into the GE. The byte selected by the read-map-select register is replicated into all four byte positions to form the resulting Dword, which is presented to the CPU."~~
12.5.2.	Removed	~~"Read Mode 1~~ ~~Read data is read into the GE. The color-compare assembler takes the appropriate 4 bit sets from the output and groups them into one byte. This byte is then replicated into all four byte positions to form the resulting Dword, which is presented to the CPU."~~
12.5.2.	Replaced	"~~pixels that~~" With "pixel which either matches the value or do not"
12.5.2.	Replaced	"~~pixels that~~" With "pixel that either matches the value or does not"

Update History for Graphics Engine chapter

Section	Change	Text
14.5.1.3	Removed	~~“Write Mode 0~~ ~~One of the bytes of the write data, depending upon the byte address, is selected and rotated 0-7 bits. The resulting byte is then output as all four bytes of the result. In parallel, the four set/reset mask bits are each expanded to a byte. The four set/reset enable bits select either the rotated write data or the expanded set/reset bytes, on a byte-by-byte basis. The VGA ROP is then applied to the selected data and the latched read data. A bit mask controls which bits of all bytes are altered and a map mask controls which bytes are written.”~~
14.5.1.4	Removed	~~“Write Mode 1~~ ~~Write mode 1 is a subset of Write Mode 0. No CPU-supplied write data is used. The read data latched from a previous read operation is written. The bit mask is disabled. The map-masks are implemented as they are for Write Mode 0.”~~
14.5.1.5	Removed	~~“Write Mode 2~~ ~~Write mode 2 is similar to write mode 0, except that only data from the bit-alignment is used. In the bit alignment unit, the four least significant bits of the current byte are selected. Since the bit ordering is different for this operation than Windows bitmaps, the stutterer/selector must reverse the order of the four bits. Bit-expander expands the bits into the FG or BG, which software will have set to all ‘1’s or ‘0’s, respectively.”~~
14.5.1.5	Removed	~~“The bit-mask, VGA logic-operation, and map-masks are implemented in the same way as Write Mode 0.”~~
14.5.1.6	Removed	~~“Write Mode 3~~ ~~Write mode 3 is similar to write mode 0, except that the CPU data byte is rotated and anded with the contents of the bit mask register to form the bit mask. The set/reset mask bits are expanded to one byte each and used regardless of the state of the set/reset enable bits.”~~
12.6.	Replaced	“~~A pixel’s X coordinate is usually expressed as an unsigned byte quantity, the number of bytes from the left edge of a scan line. If the X_dir field of the Pixel_depth register is ‘0’, advancing from left-to-right, then X points to the least-significant byte of the starting pixel. If the X_dir field of the Pixel_depth register is ‘1’, advancing from right-to-left, then X points to the most-significant byte of the starting pixel~~” With “A pixel’s X coordinate is usually expressed as an unsigned Byte quantity, the number of Bytes from the left edge of a scan line. If the X_dir field of the Pixel_depth register is ‘0’, advancing from left-to-right, then X points to the least-significant Byte of the starting pixel. If the X_dir field of the Pixel_depth register is ‘1’, advancing from right-to-left, then X points to the most-significant Byte of the starting pixel.”
12.6.1.	Replaced	“~~Cmd is one of the following~~:” With “ This is shown in Table 12-4 CMD operations:”
12.7.	Removed	~~“Except for the VGA registers, Writes, must be done using double-word (32-bit) transfers. There are a few extended registers for which only the lower 16 bits are defined. For these, 16-bit writes are acceptable. VGA registers may be written to via byte, word or double-word accesses.”~~
12.8.	Replaced	“~~accepting the short possible periods of being held~~.” With “reducinghold period as much as possible.”
12.8.1.	Replaced	“~~enough~~” With “exact amount of”
12.8.1.	Added	“command”
12.9.3.	Replaced	"**~~Top of Data FIFO~~”** With **“Data_Port”**

Section	Change	Text
12.9.3.	Added	“Destination Operand Base Address Register Regoffset = 018h (DST_Base)Destination Operand Base Address Register Regoffset = 018h (DST_Base)”
12.9.6.	Added	“implemented”
12.9.6.	Added	The complete addressing of this register is described in Section 9.5.3.1. Command Initiation.
12.9.6.	Replaced	“~~The address of this register is described in~~ Section 9.5.5.” With “The contents of this register are not altered by drawing operations.”
12.9.7.	Added	The contents of this register are defined to be FFFFFFFFh.
12.9.10.	Replaced	“~~to ‘1’~~ if “ With “when set to ‘1’”
12.9.14.	Removed	“~~corner which is the~~ “
12.9.14.	Replaced	“~~Bits 32-16 **Src_Y.** The unsigned Y coordinate of the starting corner of the source operand~~.” With “Bits 32-16 **Src_Y.** The source operand starting corner of the unsigned Y coordinate.”
12.9.14.	Replaced	“Bits 15-0 **Src_X.** The unsigned X location of the starting corner of the source operand. When bitmap expansion is not enabled, this is a Byte address. When in bitmap expansion is enabled, this is a bit address.” With “Bits 15-0 **Src_X.** The source operand starting corner of the unsigned X location. When bitmap expansion is not enabled, this is a Byte address. When in bitmap expansion is enabled, this is a bit address.”
12.10.4.	Added	“4) Set Width.Width to one less than the width of the rectangular area, in Bytes”
12.10.4.	Replaced	“~~5) Set Width. Width to one less than the width of the rectangular area, in bytesBytes~~” With “5) Set WidthHeight.Width Height to one less than the width height of the rectangular area, in bytesBytes”
12.10.4.	Replaced	“~~10) Write the upper-left coordinates of the area to be filled into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address~~.” With “10) Write the upper-left coordinates of the area to be filled into the Dst_XY register.”
12.10.5.	Replaced	“~~14) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address~~.” With “14) Write the starting coordinates of the destination area into the Dst_XY register.”
12.10.6.	Replaced	“~~13) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address, starting the BitBlt~~.” With “13) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.”

Section	Change	Text
12.10.7.	Replaced	"~~12) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Width-specified address alias and the width of the destination character (in bytes) in the Count field of the address, starting the BitBlt~~." With "12) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt."
12.11.	Replaced	"~~(Cursor_XY.X + 63, Cursor_XY.Y + 63)~~" With "(Cursor_XY.X + 63, Cursor_XY.Y + height - 1)"
12.11.1.	Added	"Cursor Height Register CR29 3X5h Index 29 (RW) Bit 7 **Cursor XOR Pre/Post Look Up Table.** When this bit is set to one, the graphics cursor XOR operation is performed before the look up table. The default behavior, when this bit is set to zero, is for the XOR operation to happen after the look up table. This is correct for 15, 16, 24 bit per pixel modes but not 8bpp. Bits 6-0 **Cursor height.** This field represents the vertical extent of the graphics cursor in scan lines. Setting this to zero effectively turns the graphics cursor off. Values greater than 40h (decimal 64) are meaningless and produce unpredictable results. Note: there is no cursor width register - the width is always 64 pixels. If a narrower cursor is required, pad the bitmap on the right with transparent cursor color (pad the AND plane with '1's on the right and the XOR plane with '0's). This register is set to 00h after reset."
12.11.2.	Added	Cursor Color 0 Register A CR2A 3X5h Index 2A (RW) Bits 7-0 **Cursor Color 0 Red.** These bits are the red component of cursor color 0. This register is undefined after reset.
12.11.3.	Added	Cursor Color 0 Register B CR2B 3X5h Index 2B (RW) Bits 7-0 **Cursor Color 0 Green.** These bits are the green component of cursor color 0. This register is undefined after reset.
12.11.4.	Added	Cursor Color 0 Register C CR2C 3X5h Index 2C (RW) Bits 7-0 **Cursor Color 0 Blue.** These bits are the blue component of cursor color 0. This register is undefined after reset.

Section	Change	Text
12.11.5.	Added	Cursor Color 1 Register A CR2D 3X5h Index 2D (RW) Bits 7-0 **Cursor Color 1 Red.** These bits are the red component of cursor color 1. This register is undefined after reset.
12.11.6.	Added	Cursor Color 1 Register B CR2E 3X5h Index 2E (RW) Bits 7-0 **Cursor Color 1 Green.** These bits are the green component of cursor color 1. This register is undefined after reset.
12.11.7.	Added	Cursor Color 1 Register C CR2F 3X5h Index 2F (RW) Bits 7-0 **Cursor Color 1 Blue.**These bits are the blue component of cursor color 1. This register is undefined after reset.
12.11.8.	Added	Graphics Cursor Address Register 0 CR30 3X5h Index 30 (RW) Bits 7-1 **Cursor AND Address Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the cursor's AND mask. The cursor's XOR mask begins at this address + 512. This memory must be aligned on a 1 KByte boundary. For a discussion of DRAM linear addresses, see section tbc. Note that the cursor bitmap is ordered such that the top left hand corner of the cursor is represented by bit 7 of the Byte addressed by this field (AND) and bit 7 of the Byte at 512 plus this address (XOR plane). The next pixel right is represented by bit 6 of these Bytes and so on until the bottom right hand pixel is represented by bit 0 of the Byte locat ed at this address plus 511 (AND) and bit 0 of the Byte at 1023 plus this address. Bit 0 *Reserved.* This bit should be written as zero. This register is undefined after reset.

Update History for Graphics Engine chapter

Section	Change	Text
12.11.9.	Added	Graphics Cursor Address Register 1 CR31 3X5h Index 31 (RW) Bits 7-0 **Cursor AND Address Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the cursor’s AND mask. For a discussion of DRAM linear addresses, see section 5.4.5. This register is undefined after reset.
12.11.10.	Added	Graphics Cursor Address Register 2 CR32 3X5h Index 32 (RW) Bits 7-3 *Reserved.* These bits should be written as zero. Bits 2-0 **Cursor AND Address Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the cursor’s AND mask. For a discussion of DRAM linear addresses, see section tbc. This register is undefined after reset.

13. VIDEO CONTROLLER

13.1. INTRODUCTION

The STPC Client controls video signal input, buffering and output through the Video Controller. The Video Input and buffering is controlled by the Video Input Port, while the Video Output Port controls the video output in several standards.

13.1.1. THE VIDEO INPUT PORT

The Video Input Port interfaces to external video in several digital formats.

The Video Input Port includes a fully functional VIP Host Master Port with hardware polling, programmable time-out period and programmable time-slice arbitration logic. This interface implements the full VIP Host Port Protocol - burst mode, master or slave-terminated transfers, wait-states and time-out transfers.

The channel has a 32-Byte FIFO to optimize transfers to system memory while insuring adequate bandwidth for VIP transfers.

Both channels support Hardware Polling with a programmable delay period to reduce polling when the selected target FIFO is not ready. Hardware polling minimizes transfer startup time for the DMA operations.

Arbitration between DMA Channels and host accesses can be round-robin or priority based. Round-robin arbitration and maximum burst length controls allow the maximum latency to be calculated and controlled. Priority based arbitration insures maximum bandwidth for critical tasks.
The time-out period is programmable to accommodate devices with long access times.

13.1.2. THE VIDEO PIPELINE REGISTERS

The video input and display is controlled through the Video Pipeline registers. These registers provide the settings for the display buffer areas, filter control, color mixing and color space mixing.

13.2. VIDEO INPUT PORT OVERVIEW

The purpose of the Video Input Port is to accept an encoded digital video stream in one of a number of industry standard formats, decode it, optionally decimate it 2:1, and deposit it into an offscreen area of the frame buffer. An interrupt request can be generated when an entire field or frame has been captured.

13.3. DIGITAL VIDEO INPUT FORMATS

The video input port can be programmed to decode one of several video formats. The following sections discuss this functionality in more detail.

Existing video input formats include:

Input Format	Description
VIP 1.0 (ITU-R 656)	Lock internal timing generator to EAV codes
UTIR-601	8-bit multiplexed CCIR 601
Passthrough	Video stream is passed directly through the Video Ports without processing

13.3.1. VIP 1.0 Compatible Video

The Video Input Port supports the simplified SAV (Start of Active Video) and EAV (End of Active Video) codes as defined in the VIP 1.0 Specification.

In this mode, the Video Timing Generator cannot be used to specify the horizontal or vertical active periods. The capture of video must be based solely on the SAV and EAV codes embedded in the video stream.

This implementation includes:

– The Video Timing Generator may be used independently in this mode, to generate the system timing signals, HSYNC# and B/T#.
– Horizontal and vertical active window based on SAV and EAV codes only
– Byte swapping may be disabled based on the Task Bit from the SAV code.
– Invalid pixel detection when the value is 0x00. When a pixel data value of 0x00 is encountered during an active line, the data is not written to the frame buffer and the pointer is not incremented. This allows re-sampled video to be output without changing the PIXCLK frequency.

13.3.2. 8-bit multiplexed ITU-R 601

This mode provides a glueless video interface to the STi3520A MPEG-2 decoder chip. The video data interface consists of 8 data pins, 2 control pins and a pixel clock. The UTIR-601 outputs video data in 4:2:2 format, multiplexed to 8 bit data words in Cb, Y, Cr, Y format. The UTIR-601 uses input signals field (B/T#) and horizontal sync (HSYNC#) to generate video timing. The STPC Client Video Controller can be configured to generate video timing (driving HSYNC# and B/T#) or lock to these signals when generated by an external video timing source. The OSD signal is not supported.

13.3.3. Video Pass-through Mode

Note: VIP 1.0 video may not be compatible with passthrough mode if the video encoder depends on the SAV and EAV signals for video timing.

Video data can be accumulated in the frame buffer and / or simply passed through the Video Controller to a video encoder. Since there is no buffering in his mode, all components (in the video path) must be running at the same clock speed. In addition, an external chip and the video encoder must be genlocked.

The incoming video stream and control signals are not decoded in any way, they are simply passed directly to the video output port as a constant, uninterrupted stream. Since there is no knowledge of video timing, the decimator must be disabled while using this mode as it would corrupt the incoming data stream. Any ITU-R-656 extensions (SAV, EAV, ancillary data) present in the video stream are passed to the output unchanged.

This mode is included to allow raw video timing and data to be sent to the Video Output Port. The video port is disabled when pass through mode is selected. The timing of the B/T# and HSYNC# signals reflects the timing seen by an external device.

13.4. VIP Specifications Not Supported

13.4.1. Ancillary Data

The Video Input Port does not support the capture of Ancillary Data. The VIP specification allows this method to be used for capturing sliced VBI and digital audio PCM data through the Video Input Port.

Digital Audio PCM data cannot be transferred to the frame buffer. The specification for audio data transfer as Ancillary Data is still being developed by an ITU task force.

13.4.2. DMA Channel Restrictions

The DMA controllers are only capable of accessing Host FIFO Space. They are not capable of accessing Host Register Space.

13.4.3. Chroma Mask

There is no support provided for chroma key mixing since the STPC CRTC supports both color and chroma based video mixing.

13.5. VIDEO INPUT MODULE ADDRESS SPACE

VIP Target devices are memory mapped into Graphics Register Space in the 4 MBytes section allocated to memory mapped graphics and video registers.

256 KBytes of this space are allocated to the Video Input Module. VIP Host Target Address Space occupies 32 KBytes of this region. Host Port and DMA registers occupy another 32 KBytes of this region. The Video Input Module Address Map is shown in Table 13-1.

AD[31:28]	[27:24]	[23:20]	[19:18]	[17:15]	Description
0000	GBASE	0110	00	100	Video Input Port Registers - PIXCLK domain

Table 13-1. Video Input Module Address Space

13.6. VIP INPUT PORT REGISTERS

The video input port registers are all initialized to 0 at power up. Writes to registers marked reserved are ignored, reads always return 0.

13.6.1. FRAME BUFFER ADDRESS READBACK REGISTER

The frame buffer address register is loaded from vid_ad0 or vid_ad1. A read path is provided at this address for testing purposes. The value is unsynchronized and should not be read during active video.

Fb1_Adr Access = 8400000h Regoffset = 0x00h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv										FBA					
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBA															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns ‘0’s.
Bits 21-0	**FBA**	**Frame Buffer Address.** Read-only static readback for frame buffer address register (for test). Lower address bits 2-0 are reserved and when read return a value of ‘0’.

13.6.2. VIDEO INPUT PORT CONFIGURATION REGISTER

The top Byte of vin_cfg is reserved for enabling and disabling interrupts.

Bits 31-28 are used to reset interrupt enables. Individual interrupts are disabled by writing a ‘1’ to the associated Reset IRQ enable field. Writing a zero to the Reset IRQ enable field preserves the existing enable status. Read values for these fields are undefined and should be masked off before comparing.

Bits 27 to 24 are the interrupt enables. Individual interrupts are enabled by writing a ‘1’ to the associated interrupt enable field. Writing a zero preserves the existing value.

Writing a ‘1’ to both the enable and reset enable field at the same time produces undefined results.

Vin_Cfg Access = 8400000h Regoffset = 0x04h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rst_BFIEn	Rst_FIEn	Rst_VBIEn	Rst_BOEn	BF_IEn	F_IEn	VB_IEn	BO_IEn.	VCLK		ST	DE	AU	VIF		
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv	BS	FB1		FDC		FCC			DBE		EVC	FB1			IPE
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bit 31	**Rst_BFIEn**	**Reset Buffer Full IRQ Enable** (Write-only, see table below: Table 13-2).
Bit 30	**Rst_FIEn**	**Reset Field IRQ Enable** (Write-only, see table below:Table 13-3).
Bit 29	**Rst_VBIEn**	**Reset Vertical Blank IRQ Enable** (Write-only, see table below: Table 13-4).
Bit 28	**Rst_BOEn**	**Reset Buffer Overflow IRQ Enable** (Write-only, see table below: Table 13-5).
Bit 27	**BF_IEn**	**Buffer Full IRQ Enable** (see table below: Table 13-6).
Bit 26	**F_IEn**	**Field Change IRQ Enable** (see table below: Table 13-7).
Bit 25	**VB_IEn**	**Vertical Blank IRQ Enable** (see table below: Table 13-8).
Bit 24	**BO_IEn.**	**Video Input Buffer Overflow Enable** (see table below: Table 13-9).

Bit Number	Mnemonic	Description
Bits 23-22	VCLK	**VCLK source.** VCLK source determines the clock source for the Video Input Port. A clock is required for the Video Input Port to respond to host accesses. The power on default is GCLK. VCLK is only an output when DCLK is the enabled source and the video port clock and timing signals are being generated by the CRTC. The following sequence is recommended when enabling the external VCLK. 1. Set vin_cfg[23:22] to '00' (Select GCLK for the internal timing). This insures that the VCLK pin is not being driven by the Video controller. 2. Enable the external VCLK driver. 3. Set vin_cfg[23:22] to '01' (Select VCLK for internal timing). This resets the time-out counter and selects the VCLK input. If the time-out counter (~16 GCLK periods) expires without detecting a valid VCLK input, the clock source will be changed back to GCLK. 4. Check vin_stat[9] to make sure that VCLK is present. Note: If video is not being captured correctly, vin_stat[9] should be checked to be sure that a valid VCLK is being provided (see table below: Table 13-10).
Bit 21	ST	**Start Buffer.** Controls which video input buffer will be filled first. (see table below: Table 13-11).
Bit 20	DE	**Decimator Enable** (see table below: Table 13-12).
Bit 19	AU	**Auto Update.** This bit enables automatic updating of the displayed video buffer (see table below: Table 13-13).
Bits 18-16	VIF	**Video Input Format.** This field controls how the video stream will be decoded. In VIP 1.0 Video Mode the timing information is recovered from SAV and EAV codes embedded in the video stream. The video timing generator may be used to generate system timing In multiplexed CCIR-601 mode, the video input port can generate video timing or lock to an external source. Modes 0 and 1 are STi3520A compatible. Video timing generation is enabled for formats 0-3 and disabled for modes 4-7 (see table below: Table 13-14).
Bit 15	Rsv	**Reserved.** This Read-Only bit is reserved. When read it returns undefined data (see table below: Table 13-15).
Bit 14	BS	**Byte Swap.**
Bits 13-12	FB1	**FB1 High Water Mark.** HIGH ORDER 2 BITS of the frame buffer FIFO high water mark. Vin_cfg[13-12] and vin_cfg[3-1] are concatenated to form the video frame buffer FIFO high water mark. Video data is buffered between the video input port and the frame buffer in a FIFO (FB1). FB1 High Water Mark is used to optimize frame buffer accesses by specifying the point where FB1 makes a request to the frame buffer memory controller. When the frame buffer FIFO contains this number of QWORDs it will request access to the frame buffer.
Bits 11-10	FDC	**Frame Drop Control.** Frame Drop Control determines how often frames are captured. A Frame period consists of an odd and even field sequence, even when only one of the fields is captured. Frame dropping can be used to reduce the input video stream bandwidth when bottlenecks prevent capture and/or transmission at full video rates (see table below: Table 13-16).

Bit Number	Mnemonic	Description
Bits 9-7	FCC	**Field Capture Control.** Field Capture Control determines what fields are used to generate a frame. In progressive scan mode, fields are de-interlaced by merging odd and even fields into a single video buffer. This method of de-interlacing provides the highest vertical resolution but can cause motion artifacts where there are areas of movement. When capturing interlaced video in double buffer mode, the buffer is switched at the end of each enabled field (see table below: Table 13-17).
Bit 6	DBE	**Double Buffer Enable.** Double Buffer Enable allows the amount of frame buffer memory to be reduced when capturing at less than full video rates. When single buffering is selected, all captured fields are written to the frame buffer using the buffer selected by the start buffer field (see table below: Table 13-18).
Bit 4	EVC	**Enable Video Capture.** Enable Video Capture starts or stops video capture operation. Video capture starts at the first enabled field of the next frame when video is being captured based on the field bit. When both fields are enabled, frame capture starts with field 1 (the odd field) as defined by ITU-R 656 (see table below: Table 13-19).
Bits 3-1	FB1	**FB1 High Water Mark.** LOW ORDER 3 BITS of the frame buffer FIFO high water mark.. See vin_cfg[13:12] for the high order bits.
Bit 0	IPE	**Input Port Enable.** The Input Port Enable allows the port to be reset by software. This bit should not be asserted during normal operation as it unconditionally resets the port to the default values (see table below: Table 13-20).

Bit 31	Reset Buffer Full IRQ Enable
0	Preserve Buffer Full IRQ enable
1	Reset Buffer Full IRQ enable

Table 13-2. Reset Buffer Full IRQ Enable

Bit 30	Reset Field IRQ Enable
0	Preserve Field IRQ enable
1	Reset Field IRQ enable

Table 13-3. Reset Field IRQ Enable

Bit 29	Reset Vertical Blank IRQ Enable
0	Preserve Vertical Blank IRQ enable
1	Reset Vertical Blank IRQ enable

Table 13-4. Reset Vertical Blank IRQ Enable

Bit 28	Reset Buffer Overflow IRQ Enable
0	Preserve Buffer Overflow IRQ enable
1	Reset Buffer Overflow IRQ enable

Table 13-5. Reset Buffer Overflow IRQ Enable

Bit 27	Buffer Full IRQ Enable
0	Preserve existing BF_IEn value
1	IRQ is generated when either video input buffer goes full

Table 13-6. Buffer Full IRQ Enable

Bit 26	Field Change IRQ Enable
0	Preserve existing F_IEn value
1	IRQ is generated when the internal Field bit changes

Table 13-7. Field Change IRQ Enable

Bit 25	Vertical Blank IRQ Enable
0	Preserve existing VB_IEn value
1	IRQ is generated at the end of the current field after flushing the frame buffer FIFO

Table 13-8. Vertical Blank IRQ Enable

Bit 24	Video Input Buffer Overflow Enable
0	Preserve existing BO_IEn value
1	IRQ is generated when either video input buffer overflows (see vin_stat bit 24)

Table 13-9. Video Input Buffer Overflow Enable

Bit 23	Bit 22	VCLK source
0	0	Use GCLK for video timing generator and interface clock (default)
0	1	Use input VCLK for video interface clock
1	0	Use DCLK for video interface clock
1	1	*Reserved*

Table 13-10. VCLK Souce

Bit 21	Start Buffer
0	video input buffer 0 filled first
1	video input buffer 1 filled first

Table 13-11. Start Buffer

Bit 20	Decimator Enable
0	no decimation of input pixels
1	Enable 2:1 video decimator

Table 13-12. Decimator Enable

Bit 19	Auto Update
0	Buffer must be updated by the driver
1	Buffer automatically switched to the most recently completed display buffer

Table 13-13. Auto Update

Bit 18	Bit 17	Bit 16	Video Input Format
0	0	0	VIP 1.0 Video Mode
0	0	1	Reserved
0	1	0	Pass through
0	1	1	STi3520A Compatibility mode (Multiplexed CCIR-601)
1	X	X	Reserved

Table 13-14. Video Input Format

Bit 15	Description
0	Bytes within words are swapped on input to match the format expected by the video display, Y Cb Y Cr (default setting)
1	Bytes within words are passed directly through the input port. This setting should be selected in pass through mode to maintain Cb Y Cr Y format

Table 13-15. Reserved

Bit 11	Bit 10	Frame Capture/Drop
0	0	Capture all frames
0	1	Capture first frame, drop one, repeat
1	0	Capture first frame, drop two, repeat
1	1	Capture first frame, drop three, repeat

Table 13-16. Frame Drop Control

Bit 9	Bit 8	Bit 7	Video Input Format
0	0	0	Reserved
0	0	1	Interlaced mode, capture only odd fields
0	1	0	Interlaced mode, capture only even fields
0	1	1	Interlaced mode, capture both even and odd fields
1	X	X	Reserved

Table 13-17. Field Capture Control

Bit 6	Double Buffer Enable
0	single buffer
1	double buffer

Table 13-18. Double Buffer Enable

Bit 4	Enable Video Capture
0	Video capture will end after current frame
1	Video capture will begin at start of next frame or task

Table 13-19. Enable Video Capture

Bit 0	Input Port Enable
0	Video input port disabled, counters/state machines initialized, capture of video stopped
1	Video input port enabled

Table 13-20. Input Port Enable

13.6.3. VIDEO INPUT PORT STATUS REGISTER

Status bits that are latched are cleared by writing to vin_stat with a bit pattern that contains a ‘1’ in the locations that are being reset and ‘0’ in the locations that are to be preserved.

Vin_Stat Access = 8400000h Regoffset = 0x08h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv				Ch2Int	Ch1Int	VCLK	VB	CP	F IRQ	VB IRQ	OEF	AB	BO IRQ	BF	B 0 F
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bit 11	**Ch2Int**	**Channel 2 Interrupt Pending Copy.** This bit is a read-only copy of the interrupt request bit in the DMA channel status register. It is provided here for convenience. See the description under DMA operation for more information.
Bit 10	**Ch1Int**	**Channel 1 Interrupt Pending Copy.** This bit is a read-only copy of the interrupt request bit in the DMA channel status register. It is provided here for convenience. See the description under DMA operation for more information.
Bit 9	**VCLK**	**VCLK present.** This read-only bit reflects the prescence of the VCLK signal (see table below: Table 13-21).
Bit 8	**VB**	**Vblank.** This read-only bit reflects the value of the internal vertical blank signal (see table below: Table 13-22).
Bit 7	**CP**	**Capture in Progress.** This read-only bit is set at the start of the first video frame after Enable Capture of Video is set. It is cleared at the end of the first frame after Enable Capture of Video is cleared. This bit is controlled by hardware.
Bit 6	**F IRQ**	**Field IRQ.** This bit is set and latched when the digital field bit changes and the Field IRQ enable bit (vin_cfg) is set to 1. It is cleared by writing a value of 1 to vin_stat[6].
Bit 5	**VB IRQ**	**VBlank IRQ.** This bit is set and latched when the last line of enabled video has been written to the frame buffer and the vblank IRQ enable bit (vin_cfg) is set to ‘1’. It is cleared by writing a value of ‘1’ to vin_stat[5].
Bit 4	**OEF**	**Odd/Even Field.** This is a read only bit that reflects the value of the internal field flag (as defined by ITU-R 656). This status bit is not affected by the B/T# inversion control.

Bit Number	Mnemonic	Description
Bit 3	AB	**Active Buffer.** This is a read only bit that reflects the value of an internal flag. It indicates which video buffer is currently being filled (see table below: Table 13-23).
Bit 2	BO IRQ	**Buffer Overrun IRQ.** This bit is set and latched when either of the buffers receives a write from the pixel packer and the corresponding buffer full flag is set, indicating that a buffer overrun has occurred. If the corresponding interrupt enable is asserted, an interrupt is generated. This bit is cleared by writing a value of '1' to vin_stat[2].
Bit 1	BF	**Buffer 1 Full.** This bit is set and latched when buffer 1 receives the last pixel of a captured frame. This condition can, if enabled, generate an IRQ This bit is cleared by writing a value of '1' to vin_stat[1].
Bit 0	B 0 F	**Buffer 0 Full.** This bit is set and latched when buffer 0 receives the last pixel of a captured frame. This condition can, if enabled, generate an IRQ. This bit is cleared by writing a value of '1' to vin_stat[0].

Bit 9	VCLK Signal
0	VCLK is not present
1	VCLK is present

Table 13-21. VCLK Signal

Bit 8	VBlank
0	Active video region
1	Vertical blanking region

Table 13-22. VBlank

Bit 3	Active Buffer
0	Field 1 (Top field) is currently being processed.
1	Field 2 (Bottom field) is currently being processed.

Table 13-23. Active Buffer

Programming notes

Read only bits are unaffected by write cycles. Reserved bits are undefined and must be masked off before making comparisons.

13.6.4. VIDEO INPUT BUFFER ADDR 0

Vin_Ad0 Access = 8400000h Regoffset = 0x0Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv										VBA 0					
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBA 0															
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns ‘0’s.
Bits 21-0	**VBA 0**	**Video Buffer Addr 0.** Quad word frame buffer start address for video input buffer 0. Lower address bits 2-0 are reserved and when read return a value of ‘0’.

Programming notes

Lines of video always start at a quad word boundary in the frame buffer. When the display window size is not a multiple of 8, any remaining Bytes in the last quad word will be unused (and undefined).
The LS 3 bits of this register are hardwired to zero to force QWORD alignment.

13.6.5. VIDEO INPUT BUFFER ADDR 1

The LS 3 bits of this register are hardwired to zero to force QWORD alignment.

Vin_Ad1 Access = 8400000h Regoffset = 0x10h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv										VBA 1					
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBA 1															
Default value after reset =undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns ‘0’s.
Bits 21-0	**VBA 1**	**Video Buffer Addr 1.** Quad word frame buffer start address for video input buffer 1. Lower address bits 2-0 are reserved and when read return a value of ‘0’.

13.6.6. VIDEO INPUT DESINATION PITCH

Vin_Dp Access = 8400000h Regoffset = 0x14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv		DP													
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-14	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns ‘0’s.
Bits 13-0	**DP**	**Destination Pitch.** This register holds the number of Bytes in the frame buffer the beginning of one video scan line to the next. Lower address bits 2-0 are reserved and when read return a value of ‘0’.

Programming notes

When the Field Capture Control selects one of the interlaced modes, the destination pitch is set to the number of quad words required to hold a line of video data. When de-interlacing by merging odd and even fields is selected, the destination pitch should be set to twice the number of quad words required to hold a line of video data. The LS 3 bits of this register are hardwired to zero to force QWORD alignment.

13.6.7. EXTERNAL TIMING GENERATOR 1

Vtg_Ext1 Access = 8400000h Regoffset = 0x28h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VGT	Rsv	Bt_oe	HSYNC_OE	Rsv	Bt_pol	HSYNC_POL	Rsv	GM			Rsv			HS_St	
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS_St				Rsv			HS_End								HS_Odd
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bit 31	**VGT**	**VTG enable** (see table below: Table 13-24).
Bit 30	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bit 29	**Bt_oe**	**Output enable for the B/T# video timing signal.** Set to '1' when the Video Controller is generating the system video timing signals (see table below: Table 13-25).
Bit 28	**HSYNC_OE**	**Output enable for the HSYNC- VIDEO TIMING SIGNAL.** Set to '1' when the Video Controller is generating the system video timing signals (see table below: Table 13-26).
Bit 27	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bit 26	**Bt_pol**	**B/T# polarity, .** This bit defines the active edge for the B/T# signal as input and output. The setting of this bit sets the polarity of the external signal to be high or low true. The polarity of the internal field (F) and horizontal blank (H) bits are not affected (see table below: Table 13-27).
Bit 25	**HSYNC_POL**	**HSYNC# POLARITY.** This bit defines the active edge for the HSYNC# signal as input and output. The setting of this bit sets the polarity of the external signal to be high or low true. The polarity of the internal field (F) and horizontal blank (H) bits are not affected by these bits (see table below: Table 13-28).
Bit 24	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bit 23-21	**GM**	**Genlock Mode.** Defines the method for genlocking to an external source (see table below: Table 13-29).
Bit 20-18	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.

Bit Number	Mnemonic	Description
Bit 17-12	HS_St	**Leading edge of HSYNC# in pixels.** Allows the HSYNC# trailing edge to be shifted relative to the start of the horizontal line in pixels, referenced to the horizontal counter. Since the video clock runs at twice the pixel rate, this value must be multiplied by 2 (shifted left 1) before being compared to the horizontal count. HS_Odd is used to generate the least significant bit, insuring that HSYNC# can be generated at any point during the horizontal scan line.
Bit 11-9	Rsv	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bit 8-1	HS_End	**Trailing edge of HSYNC# in pixels.** Allows the HSYNC# trailing edge to be shifted relative to the end of a horizontal line in pixels, referenced to the horizontal counter. Since the video clock runs at twice the pixel rate, this value must be multiplied by 2 (shifted left 1) before being compared to the horizontal count. HS_Odd is used to generate the least significant bit, insuring that HSYNC# can be generated at any point during the horizontal scan line.
Bit 0	HS_Odd	**HSYNC# Odd compensation.** This bit allows the leading and trailing edges of HSYNC# to be shifted by 1 VCLK to compensate for an odd number of pipe stages between the video input port and an external device. This bit is used as the least significant bit of HS_End and HS_St when being compared to the horizontal counter.

Bit 31	Video Timing Generator Enable
0	Video timing is reset to the start of field 1
1	Video timing generator is enabled

Table 13-24. Video Timing Generator Enable

Bit 29	Output Enable for the video timing signal
0	B/T# is an input
1	B/T# is an output

Table 13-25. Output Enable for the video timing signal

Bit 28	Output Enable for the HSYNC- VIDEO TIMING SIGNAL
0	HSYNC# is an input
1	HSYNC# is an output

Table 13-26. Output Enable for the HSYNC- VIDEO TIMING SIGNAL

Bit 26	B/T# polarity
0	B/T# is low for field 1
1	B/T# is high for field 1

Table 13-27. B/T# polarity

Bit 25	HSYNC# POLARITY
0	HSYNC# is low true
1	HSYNC# is high true

Table 13-28. HSYNC# POLARITY

Bit 23	Bit 22	Bit 21	Genlock Mode
0	0	0	No genlocking. Video timing generator resets horizontal and vertical counters based on H_Total and V_Total. (default)
0	0	1	Genlock to B/T# and HSYNC#
0	1	0	Genlock to SAV/EAV codes
0	1	1	Reserved
1	X	X	Reserved

Table 13-29. Genlock Mode

Programming notes

When the video timing generator is in master mode, the leading edge of the external HSYNC occurs on the clock edge following when the horizontal counter matches the HSSt value. The trailing edge occurs on the clock edge following when the horizontal counter matches the HSEnd value.

When the video timing generator is in slave mode, the horizontal counter is set to HSSt value on the second VCLK edge following HSYNC# assertion. In slave mode, the horizontal timing is independent of the trailing edge of HSYNC# and HSEnd is ignored. The default values are specified to match ITU-R 656 (525 line) timing in slave mode.

13.6.8. EXTERNAL TIMING GENERATOR 2

Vtg_Ext2 Access = 8400000h Regoffset = 0x2Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv											BT_Dly1				
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BT_Dly1					BT_Dly2										BT_Odd
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-21	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bit 20-11	**BT_Dly1**	**B/T# delay for field 1 in pixels.** This field determines the location relative to the horizontal counter that B/T# switches to indicate field 2 of an interlaced frame.
Bits 10-1	**BT_Dly2**	**B/T# delay for field 2 in pixels.** This field determines the location relative to the horizontal counter that B/T# switches to indicate field 2 of an interlaced frame.
Bit 0	**BT_Odd**	**B/T## Odd compensation, .** This bit allows the leading and trailing edges of B/T# to be shifted by 1 VCLK to compensate for an odd number of pipe stages between the video input port and an external device. This bit is used as the least significant bit of BT_Dly1 and BT_Dly2 when being compared to the horizontal counter.

13.6.9. HORIZONTAL TIMING GENERATOR

Vtg_HT Access = 8400000h Regoffset = 0x30h

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="10">H_Start</td><td>Rsv</td><td colspan="5">H_End</td></tr>
<tr><td colspan="16">Default value after reset = undefined</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">H_End</td><td>Rsv</td><td colspan="10">H_Total</td></tr>
<tr><td colspan="16">Default value after reset = undefined</td></tr>
</table>

Bit Number	Mnemonic	Description
Bits 31-22	**H_Start**	**Horizontal start of active video in pixels.** When video capture is based on the video timing generator, The H_Start and H_End values are used to determine when video is captured within the vertical display window.
Bit 21	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bits 20-11	**H_End**	**Horizontal end of active video in pixels.**
Bit 10	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bits 9-0	**H_Total**	**Total number of horizontal pixels.** This field contains the total number of pixels per line.

13.6.10. VIDEO TIMING GENERATOR

V_Start specifies the first line of active video in each field.

V_End specifies the last line of active video in each field.

Vtg_VT Access = 8400000h Regoffset = 0x34h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
V_Start										Rsv	V_ End				
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V_End					Rsv	V_Total									
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-22	**V_Start**	**Vertical Field Start.** Line number of the last line of blanked video in each field. The first line of active video is V_Start + 1.
Bit 21	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bits 20-11	**V_End**	**Vertical Field End,.** Line number of the last line of active video in each field.
Bit 10	**Rsv**	**Reserved.** This Read-Only field is reserved. When read it returns undefined data.
Bits 9-0	**V_Total**	**Vertical Total, V_Total.** V_Total contains the total number of lines in a field. When the internal vertical counter reaches the value contained in V_Total it restarts from either count zero or count one, depending on the field. It gets reset to 1 at the beginning of field 1, making the number of lines in field 1 equal to V_Total. At the beginning of field 2 it gets reset to zero making the number of lines V_Total + 1. The internal field bit (F) gets inverted coincident with the resetting of the vertical and horizontal counters.

13.7. VIDEO ACCELERATOR REGISTERS

The Video Pipeline registers, similar to the extended graphics (non-VGA) registers, are located in the 4-MByte memory-mapped registers region of the 16-MByte memory space occupied by the Graphics Controller. The Video Pipeline registers are located at the 256-KByte wide sub-region 2. The Figure 13-1 below shows the address format for the Video Pipeline registers.

All registers can be read with accesses of any width. The CPU can read any register via Byte (8-bit), word (16-bit), or double-word (32-bit) accesses. Writes must be done using double-word (32-bit) transfers.

MBytes Region 1

KBytes-region2

31 27	26 24	23 22	21 18	17 16	15
00001	<GBASE>	01	0010	00	0

Figure 13-1. Address format for the Video Pipeline registers

13.8. SOURCE SPECIFICATION REGISTERS

13.8.1. VIDEO SOURCE BASE REGISTER

This register specifies the DRAM linear starting address of the video source image, aligned to an 8 Byte boundary.

This register is double buffered, the active register is only updated during vertical blanking.

Video_Src_Base Access = X600000h Regoffset = 0x0h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv										VSI					
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSI															
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-22	**Rsv**	**Reserved.**
Bits 21-0	**VSI**	Base linear address of the **Video Source Image**. Lower address bits 2-0 are reserved and when read return a value of ‘0’.

Programming noted

This address may specify either the top left corner or bottom left corner, depending on the state of the Y_Vid_Src_dir bit in the Video_Src_Pitch register.

13.8.2. VIDEO SOURCE PITCH REGISTER

This register contains the Video_Src_pitch field, which specifies the number of Bytes which must be added to the address of a pixel on one line of the video source image to compute the address of the corresponding pixel on the line below.

Video_Src_Pitch Access = X600000h Regoffset = 0x4h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv		VSI		Y_V_S_D	Current_Ad										
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-14	**Rsv**	**Reserved.**
Bits 13-12	**VSI**	**Video Source Image color format:** 00 - RGB 555, 01 - RGB 565, 10 - YUV 422.

Bit 11	**Y_V_S_D**	**Y_Vid_Src_dir.** Specifies the Y direction in which the Video Source Image should be read. This bit controls the translation of XY addresses to linear DRAM addresses.
		If(Y_Vid_Src_dir == 0)
		DRAM linear address = Video_Src_Base + (YDIFF * Vid_Src_Pitch);
		Else
		DRAM linear address = Video_Src_Base - YDIFF * Vid_Src_Pitch);
		Where YDIFF is value that varies.

Bits 10-0	**Current_Ad**	Specifies the amount to add to the current address to get to the address of the corresponding pixel in the next line. Lower bits 2-0 are reserved and when read return a value of ’0’.

Programming notes

This register also contains the Y_Vid_Src_dir bit which specifies the Y direction in which the Source Image should be read, and the Video_Color_fmt field which defines the the Color format of the Source Image.

13.8.3. VIDEO SOURCE DIMENSION REGISTER

This register contains the dimensions of the Video Source Image relative the the starting corner.

Video_Src_Dim Access = X600000h Regoffset = 0x8h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv						dY									
Default value after reset = undefined															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv						dX									
Default value after reset = undefined															

Bit Number	Mnemonic	Description
Bits 31-26	**Rsv**	**Reserved**.
Bits 25-16	**dY**	**dY**, the height of the Video Source image - 1, in lines, from the starting corner to the end of the image (dependent on Y_Vid_Src_dir).
Bits 15-10	**Rsv**	**Reserved.**
Bits 9-0	**dX**	**dX**, the width in pixels, of a line.

Programming notes

This register is double buffered. The active register is only updated during vertical sync.

13.8.4. CRTC BURST LENGTH REGISTER

This register contains the CRTC low water mark and burst length.

CRTC_Burst_Length Access = X600000h Regoffset = 0xCh

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Crtc_Dlwm								Crtc_lwm							
Default value after reset = 00h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Crtc_dt					Rsv	Crtc_Bl									
Default value after reset = 00h															

Bit Number	Mnemonic	Description
Bits 31-24	**Crtc_Dlwm**	**Delta low water mark.** Together with crtc_dt and crtc_lwm, this field defines a variable low water mark. When the video window starts, the CRTC low water mark is set to crtc_lwm. After that time, for every crtc_dt*8 pixels' time elapsed, the low water mark will be incremented by crtc_dlwm Bytes. Since this field is represented as a 2's complement number, setting bit 31 results in a low water mark which is a decreasing function of time. A decreasing or constant function will be the normal mode of operation of the CRTC low water mark during the video window. Note that this CRTC low water mark is distinct from the one described in CR1B. This one is valid during the video windows only. For normal CRTC operation (scanlines or pixels outside the video window), the pertinent CRTC low water mark is specified by CR1B. Guarantee of the CRTC ownership can be achieved by the Setting of this field to zero. This causes the CRTC low water mark to remain at a constant value of crtc_lwm.
Bits 23-16	**Crtc_lwm**	**Crtc_lwm,** the (initial) low water mark for the CRTC FIFO in Bytes. During the video window, if the CRTC FIFO occupancy rises above the low water mark (defined as a function of time by crtc_dlwm and crtc_dt) and the video occupancy rises above the video low water mark then ownership of the system DRAM can be given back to the CPU. The value of this register field can only be a multiple of eight (bits 18-16 are not writable and read as zeroes).
Bits 15-11	**Crtc_dt**	**Crtc_dt**, delta t. See the description of crtc_dlwm above.
Bit 10	**Rsv**	**Reserved.**
Bits 9-0	**Crtc_Bl**	**Minimum CRTC burst length.** This is the minimum number of Bytes that will be sent in one transfer to fill the CRTC FIFO (during the active video window only). This value can only be a multiple of eight (bits 2-0 are not writable and read as zeroes) since pixels are fetched 8 Bytes at a time.

13.8.5. VIDEO BURST LENGTH REGISTER

This register is the video counterpart of the previous register. It specifies the video low water mark and burst length.

Video_Burst_Length Access = X600000h Regoffset = 0x10h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Vid_Dlwm								Vid_lwm							
Default value after reset = 000000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vid_Dt					Rsv	Vid_Bl									
Default value after reset = 00000000h															

Bit Number	Mncemonic	Description
Bits 31-24	**Vid_Dlwm**	**Delta low water mark.** Together with vid_dt and vid_lwm, this field defines a variable low water mark. When the video window starts, the video low water mark is set to vid_lwm. After that time, for every vid_dt*8 pixels' time elapsed, the low water mark will be incremented by vid_dlwm Bytes. As with crtc_dlwm, above, This field is a 2's complement number. Setting this field to zero causes the video low water mark to remain at a constant value of vid_lwm.
Bits 23-16	**Vid_lwm**	**Vid_lwm,** the (initial) low water mark for the video FIFO in Bytes. During the video window, if the video FIFO occupancy rises above the low water mark (defined as a function of time by vid_dlwm and vid_dt) and the crtc occupancy rises above crtc_lwm then ownership of the system DRAM can be given back to the CPU. The value of this register field can only be a multiple of eight (bits 18-16 are not writable and read as zeroes).
Bits 15-11	**Vid_Dt**	**Vid_dt, delta t.** See the description of vid_dlwm above.
Bit 10	**Rsv**	**Reserved.**
Bits 9-0	**Vid_Bl**	**Minimum video burst length.** This is the minimum number of Bytes that will be sent in one transfer to fill the video FIFO. This value can only be a multiple of eight (bits 2-0 are not writable and read as zeroes) since pixels are fetched 8 Bytes at a time.

13.9. DESTINATION SPECIFICATION REGISTERS

13.9.1. VIDEO DESTINATION REGISTER

This register contains the coordinates of the top left corner of the video window.

This register is double buffered. The active register is only updated during vertical sync.

Video_Dst_XY Access = X600000h Regoffset = 0x14h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv						Y									
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv					X										
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-26	**Rsv**	**Reserved.**
Bits 25-16	**Y**	**Y,** the Y coordinate of the top edge of the video window, relative to the display. The first display line is line 0.
Bits 15-11	**Rsv**	**Reserved.**
Bits 10-0	**X**	**X,** the X coordinate of the left edge of the video window, relative to the display. The first pixel of each display line is pixel 0.

13.9.2. VIDEO DESTINATION DIMENSION REGISTER

This register contains the dimensions of the video window.

Vid_Dst_Dim Access = X600000h Regoffset = 0x18h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv						dY									
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv					dX										
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-26	**Rsv**	**Reserved.**
Bits 25-16	**dY**	**dY,** the height of the video window in screen lines - 1 is entered in this field.
Bits 15-11	**Rsv**	**Reserved.**
Bits 10-0	**dX**	**dX,** the width of the video window in screen pixels.

Programming notes

This register is double buffered. The active register is only updated during vertical sync.

13.10. FILTER CONTROL REGISTERS

13.10.1. HORIZONTAL SCALING AND DECIMATION REGISTER

This register contains the control for horizontal scaling and decimation.

Horiz_Scl Access = X600000h Regoffset = 0x20h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv											F0E	F1E	Rsv		
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv			HPI												
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-21	**Rsv**	**Reserved.**
Bit 20	**F0E**	**Filter Enable 0.** When in YUV modes, this bit enables the interpolator for the Y channel, and in RGB modes enables all three interpolators. If this bit is 0, pixel replication is used.
Bit 19	**F1E**	**Filter Enable 1.** When in YUV modes, this bit enables the interpolators for the U and V channels. If this bit is 0, pixel replication is used. This bit controls the RGB components in RGB modes and should be the same as F0E.
Bits 18-13	**Rsv**	**Reserved.**
Bits 12-0	**HPI**	**Horizontal Phase Increment, hpi.** Defines the horizontal scale factor. hpi is calculated from source width and destination width: hpi = ((source_width/hdf_tmp) * 4096) / dest_width Note that the maximum value for hpi is 4096.

Programming notes

This register is double buffered. The active register is only updated during vertical sync.

13.10.2. VERTICAL CONTROL AND DECIMATION REGISTER

This register contains the control for vertical scaling and decimation.

Vert_Scl Access = X600000h Regoffset = 0x28h

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="11">Rsv</td><td>VF0E</td><td>VF1E</td><td colspan="3">Rsv</td></tr>
<tr><td colspan="16">Default value after reset = 00000000h</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">Rsv</td><td colspan="13">VPI</td></tr>
<tr><td colspan="16">Default value after reset = 00000000h</td></tr>
</table>

Bit Number	Mncemonic	Description
Bits 31-21	**Rsv**	**Reserved.**
Bit 20	**VF0E**	**Vertical Filter Enable 0.** When in YUV modes, this bit enables the interpolator for the Y channel, and in RGB modes enables all three interpolators. If this bit is 0, pixel replication is used.
Bit 19	**VF1E**	**Vertical Filter Enable 1.** When in YUV modes, this bit enables the interpolators for the U and V channels. If this bit is 0, pixel replication is used. This bit controls the RGB components in RGB modes and should be the same as F0E.
Bits 18-13	**Rsv**	**Reserved.**
Bits 12-0	**VPI**	**Vertical Phase Increment, vpi.** Defines the vertical scale factor. vpi is calculated from source height and destination height: vpi = ((source_height) * 4096) / dest_height Note that the maximum value for vpi is 4096.

Programming notes

This register is double buffered. The active register is only updated during vertical sync.

13.10.3. COLOR SPACE CONVERTER SPECIFICATION REGISTER

This register contains the control for the Color Space Converter.

Clr_Con_Spec Access = X600000h Regoffset = 0x2Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv															CSCE
Default value after reset = 00000000h															

Bit Number	Mnemocnic	Description
Bits 31-1	**Rsv**	**Reserved.**
Bit 0	**CSCE**	**Color Space Converter Enable.** When set, YUV data is converted to RGB according to the formula:- R = 1.164(Y - 16) + 1.591(V - 128) G = 1.164(Y - 16) - 0.700(V - 128) - 0.336(U - 128) B = 1.164(Y - 16) + 1.733(U - 128) When clear, pixels are passed through unchanged.

13.11. VIDEO AND GRAPHICS MIXING CONTROL REGISTERS

13.11.1. MIX MODE REGISTER

This register contains the Mix_Mode field which defines the method used to mix graphics and video.

Mix_Mode Access = X600000h Regoffset = 0x30h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv															
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv														MM	
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-2	**Rsv**	**Reserved.**
Bit 1-0	**MM**	**Mix_Mode,** controls the way in which graphics and video are mixed (see table below: Table 13-30).

Bit 1	Bit 0	Mix Mode
0	0	**Video Window only.** The video always appears in a rectangular window which is defined by the Destination Specification registers.
0	1	**Video Window with Color Key.** The Destination specification is further qualified by the Color Key register. Within the specified video window, if the graphics pixel (pre color palette) is equal to the value specified by the Color Key register, then the corresponding video pixel is displayed, otherwise the graphics pixel is displayed. Note that in 8-bit graphics modes, only Color_Key[7:0] are used in the comparison and in 16-bit graphics modes, Color_Key[15:0] are used.
1	0	**Video Window with Chroma Key.** The destination specification is qualified by the Chroma key registers. Chroma key compares each of the pixel components to independent ‘high’ and ‘low’ values (between limits compare). If all the selected components are between their limits, then the corresponding graphics pixel is displayed, otherwise the video pixel is displayed. Note that the video pixel can be compared either before or after the Color Space Converter. Note also that the chrom key can be programmed to ignore any or all component values.
1	1	**Reserved.**

Table 13-30. Mix Mode

13.11.2. COLOR KEY REGISTER

This register contains the color key value used in color keying mixing.

CLR_Key Access = X600000h Regoffset = 0x34h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv								CK							
Default value after reset = 0000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CK															
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-24	**Rsv**	**Reserved.**
Bit 23-0	**CK**	**Color_Key,** this value is compared to the graphics pixel to determine whether to display the video pixel in color key mode. When the graphics is operating in 8-bit per pixel mode, Color_key[7:0] is compared, when the graphics is operating in 16-bits per pixel, Color_Key[15:0] is compared and when the graphics is operating in 24-bits per pixel, Color_Key[23:0] is compared.

13.11.3. CHROMA KEY LOW REGISTER

This register contains the chroma key low limits, the component ignore bits and the color mode bit used in chroma keying mixing.

CKL Access = X600000h Regoffset = 0x38h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv				CK	IC2	IC1	IC0	CH2L							
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1L								CH0L							
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-28	**Rsv**	**Reserved.**
Bit 27	**CK**	**Chroma key mode** (see table below: Table 13-31).
Bit 26	**IC2**	**Ignore component 2**. If set, component 2 (V or B) is ignored in the chroma key comparison.
Bit 25	**IC1**	**Ignore component 1.** If set, component 1 (U or G) is ignored in the chroma key comparison.
Bit 24	**ICO**	**Ignore component 0.** If set, component 0 (Y or R) is ignored in the chroma key comparison.
Bits 23-16	**CH2L**	**Ch2low,** the low limit against which component 2 is compared during chroma key operations.
Bits 15-8	**CH1L**	**Ch1low,** the low limit against which component 1 is compared during chroma key operations.
Bits 7-0	**CH0L**	**Ch0low**, the low limit against which component 0 is compared during chroma key operations.

Bit 27	Chroma key mode
0	components examined at input to color space converter (YUV mode)
1	components examined at output of color space converter (RGB mode)

Table 13-31. Chroma key mode

13.11.4. CHROMA KEY HIGH REGISTER

This register contains the chroma key high limits used in chroma keying mixing.

CKH Access = X600000h Regoffset = 0x3Ch

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rsv								CH2H							
Default value after reset = 00000000h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1H								CH0H							
Default value after reset = 00000000h															

Bit Number	Mnemonic	Description
Bits 31-24	**Rsv**	**Reserved.**
Bits 23-16	**CH2H**	**Ch2high,** the high limit against which component 2 is compared during chroma key operations.
Bits 15-8	**CH1H**	**Ch1high,** the high limit against which component 1 is compared during chroma key operations.
Bits 7-0	**CH0H**	**Ch0high,** the high limit against which component 0 is compared during chroma key operations.

The operation of the chroma key can be summarized as follows:-

Let Cn represent component n, n = 0..2

Let Chnlow represent Chlow for component n, n = 0..2

Let Chnhigh represent Chigh for component n, n = 0..2

Kn be the result of the compare for component n, n = 0..2

```
for(n = 0; n < 3; n++)
if((Cn >= Chnlow) && (Cn <= chmhigh))
Kn = 1
```

13.11.5. Status Register

FILTER_STAT	Access = GBase +480000h	Regoffset = 0x40h

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
V_En	Rsv														
Default value after reset = 00h															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rsv															
Default value after reset = 00h															

This register contains enable bit for the video scaler.

Bit Number	Mneumonic	Description
Bit 31	**V_En**	**vid_enable.** Setting the enable bit turns on the video scaler.
Bits 30-0	**Rsv**	**Reserved.**

Programming notes

This register defaults to 00h after reset

13.12 UPDATE HISTORY FOR VIDEO CONTROLLER CHAPTER

The following changes have been made to the Video Controller Chapter on the 15/09/99.

Section	Change	Text
13.8.2.	Changed	"~~Bits 10-3 Specifies the amount to add to the current address to get to the address of the corresponding pixel in the next line.~~" with "Bits 10-0 Specifies the amount to add to the current address to get to the address of the corresponding pixel in the next line. Lower address bits 2-0 are reserved and when read return a value of '0'."

The following changes have been made to the Video Controller Chapter.

Correction of typographical errors

Update History for Video Controller chapter

14. POWER MANAGEMENT

14.1. INTRODUCTION

For full information on the action of the SMM, please refer to the STMicroelectronics manual titled "Programming manual for STPC x86 CPU Core”. This chapter presents the control registers for SMM of the STPC.

The STPC provides the following hardware structures to assist the software in managing the power consumption by the system:

– System Activity detection,
– Three power down timers,
– Doze timer for detecting lack system activity for short durations,
– Standby timer for detecting lack of system activity for medium durations,
– Suspend timer for detecting lack of system activity for long durations,
– House-keeping activity detection,
– House-keeping timer to cope with short bursts of house-keeping activity while dozing or in standby state,
– Peripheral Activity detection,
– Peripheral timer for detecting lack of peripheral activity,
– STPCLK# modulation to adjust the system performance in various power down states of the system including full power on state.

Lack of system activity for progressively longer period of times is detected by the three power down timers. These timers can generate SMI interrupts to CPU so that the SMM software can put the system in decreasing states of power consumption. System activity in a power down state can generate an SMI interrupt to allow the software to bring the system back up to the full power on state. The chip-set supports up to 3 power down states: Doze state, Standby state and Suspend state. These correspond to increasing levels of power savings.

The chip-set can detect presence/absence of the following System activities:

– DMA Request (DRQ) activity,
– Interrupt Request (INTR) activity,
– Parallel IO (PIO) activity,
– Serial IO (SIO) activity,
– Keyboard (KBD) activity,
– Floppy Disk Controller (FDC) activity,
– Hard Disk Controller (HDC) activity,
– PCI master device activity,
– A programmable address range.

Each of these can be individually enabled. The presence of an enabled system activity resets the power down timers. The chip-set generates the SMI interrupt when no system activity is detected for the delay period programmed in the power-down timers. The software can then put appropriate sub-systems in power down mode, request STPCLK# assertion and stop CPU and other system clocks, program the current power-down state in the chip set and set up the next timer.

Presence of an enabled system activity, when the STPC is in a power down state will first enable any stopped clocks, wait for a a programmable delay to allow any internal PLLs to stabilize and then deassert STPCLK# to enable CPU execution. The device can optionally generate SMI interrupt to allow the SMM to bring the system back to power-on state.

The current revision of the STPC does not implement support for stopping CPU and other system clocks.

In Doze or Standby state, a house-keeping activity can bring the system back to full speed for a short period of time before returning back to Doze or Standby state. The chip-set can detect following house-keeping activities:

– DMA Request (DRQ) activity,
– Interrupt Request (INTR) activity,
– Keyboard (KBD) activity,
– PCI master device activity.

The house-keeping timer determines the length of time the system will be on before returning to the original power-down state. An activity can be either a system activity or a house-keeping activity but not both at the same time. Further, the Suspend state can not make use of this feature.

The absence of the following peripheral activities can be enabled to cause a SMI interrupt and thus allowing the software to put the unused peripherals in power down state while the remainder of the system is still in full power on state:

– Parallel IO (PIO) activity,
– Serial IO (SIO) activity ,
– Keyboard (KBD) activity,
– Floppy Disk Controller (FDC) activity,
– Hard Disk Controller (HDC) activity,
– A programmable address range.

Each of these can be individually enabled for inactivity detection. The presence of a peripheral activity does not reset the peripheral timer. It always times out after the programmed delay period. An SMI interrupt is generated if any of the enabled peripheral was not active for this time period. The device provides IO access trapping to detect access to a powered-down peripheral so that the software can bring the peripheral to power on state before the access is completed.

The STPC can also do software transparent power management if so enabled. In this mode of operation, doze and standby time-outs will change the CPU clock without generating an SMI interrupt. The state transitions from fully-on to doze or standby and back to fully-on will take place automatically. Also note that the suspend state can never be entered automatically and always requires software assist.

The STPC decodes the following to detect activities of various kind, see table below: Table 14-1.

Activity	Detected via
ISA DMA masters	Low to high transition of hold request of 206
PCI masters	High to low transition of any of PCIRQ2-0#
Parallel port	IO read/write at 378h-37Fh, 278h-27Fh and 3BCh-3BFh
Serial port	IO read/write at 3F8h-3FFh, 2F8h-2FFh, 3E8h-3EFh and 2E8h-2EFh
Keyboard	IO read/write at 60h, 62h, 64h and 66h
Floppy disk	IO read/write at 3F2h, 3F4h, 3F5h and 3F7h
Hard disk	IO read/write in 170h-177h, 376h,1F0h-1F7h and 3F6h address range as well as any bus master activity by the internal IDE controller.

Table 14-1. Activity Detected

14.2. POWER MANAGEMENT CONTROLLER REGISTERS

14.2.1. TIMER REGISTER 0

This register controls the timer selection for the length of timeout for doze, standby, and suspend modes.

Timer0 Access = 0022h/0023h Regoffset = 060h

7	6	5	4	3	2	1	0
SUTT			STT			Rsv	
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-5	**SUTT**	**Suspend Timeout Timer**, when set to any value other than the disable value (000), this timer will generate SMI interrupt on time out. Once enabled this timer counts down from the programmed valued. If any of the enabled system activities are detected before time out, the timer will reset and start again. These bits are encoded as follows in Table 14-2. The suspend timer will count whenever it is not disabled and the suspend time-out bit in the SMI status register 0 is not set to a 1.
Bits 4-2	**STT**	**Standby Timeout Timer,** when set to any value other than the disable value (000) this timer, on expiration, can either generate the SMI interrupt to the CPU or if programmed for auto-power saving (software transparent power management) mode, change the power-down state to Standby state (refer to auto-power saving mode for details of the power saving features are enabled with standby state). Similar to the Suspend timer, presence of an enabled system activity will reset the timer to start counting again. These bits are encoded as follows in Table 14-3. The standby timer will count whenever it is not disabled and the standby time-out bit in the SMI status register 0 is not set to a 1.
Bits 1-0	**Rsv**	**Reserved.**

Bit 7	Bit 6	Bit 5	Suspend Timer reset
0	0	0	disabled
0	0	1	4 minutes
0	1	0	8 minutes
0	1	1	12 minutes
1	0	0	16 minutes
1	0	1	32 minutes
1	1	0	48 minutes
1	1	1	64 minutes

Table 14-2. Suspend Timer reset

Bit 4	Bit 3	Bit 2	Standby Timer reset
0	0	0	disabled
0	0	1	Reserved
0	1	0	2 minutes
0	1	1	4 minutes
1	0	0	6 minutes
1	0	1	8 minutes
1	1	0	12 minutes
1	1	1	16 minutes

Table 14-3. Standby Timer reset

14.2.2. TIMER REGISTER 1

Timer1 Access = 0022h/0023h Regoffset = 061h

7	6	5	4	3	2	1	0
Rsv	HKT			PTT			Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.**
Bits 6-4	**HKT**	**House-keeping Timer.** This timer determines how long the PMU will be in Doze house-keeping state when an enabled house-keeping activity is detected while in doze or standby power-down states. It is encoded as follows in Table 14-4. The house-keeping counts only when the PMU is in one of the house-keeping states. Another house-keeping activity while the controller is in house_keeping state will reset the house-keeping timer to start counting again. A system activity detection in the house_keeping state will have the same effect as if the controller was in Doze or Standby state. Either a SMI interrupt will be generated to allow the software to bring the system to power-on state or the controller will automatically transition to power-on state. The house-keeping timer and function can be disabled by masking out all activity detection via House-keeping Enable registers.
Bits 3-1	**PTT**	**Peripheral Timeout Timer.** When set to a value other than (000) this timer on expiration, will generate SMI if any of the enabled peripherals remained inactive during the entire period. Unlike the power-down timers, the peripheral timer does not reset due to an enabled peripheral activity. It always times out after the programmed delay. A SMI interrupt is generated only if any of the enabled peripherals were inactive during this period. This field is encoded as follows in Table 14-5. The peripheral timer counts whenever it is enabled.
Bit 0	**Rsv**	**Reserved.**

Bit 6	Bit 5	Bit 4	House-keeping Timer reset
0	0	0	disabled
0	0	1	64 micro-seconds
0	1	0	128 micro-seconds
0	1	1	256 micro-seconds
1	0	0	Reserved
1	0	1	4 milli-seconds
1	1	0	16 milli-seconds
1	1	1	32 milli-seconds

Table 14-4. House-keeping Timer reset

Bit 3	Bit 2	Bit 1	Peripheral Timer reset
0	0	0	disabled
0	0	1	8 seconds
0	1	0	16 seconds
0	1	1	32 seconds
1	0	0	64 seconds
1	0	1	128 seconds
1	1	0	256 seconds
1	1	1	512 seconds

Table 14-5. Peripheral Timer reset

14.2.3. TIMER REGISTER 2

Timer 2 Access = 0022h/0023h Regoffset = 08Dh

7	6	5	4	3	2	1	0
DTT			Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7- 5	DTT	**Doze Timeout Timer.** When set to any value other than the disable value (00), this timer, on expiration, can either generate the SMI interrupt to the CPU or if programmed for auto-power saving (software transparent power management) mode, change the power-down state to Doze state (refer to auto-power saving mode for details of the power saving features that are enabled with Doze state). Similar to the suspend timer, presence of an enabled system activity will reset the timer to start counting again. This 3-bit field is encoded as follows in Table 14-6. The doze timer will count whenever it is not disabled and the doze time-out bit in the SMI status register 0 is not set to a ‘1’.
Bits 4-2	Rsv	**Reserved.**

Bit 7	Bit 6	Bit 5	Doze Timer reset
0	0	0	disabled
0	0	1	50 milli-seconds
0	1	0	100 milli-seconds
0	1	1	500 milli-seconds
1	0	0	Reserved
1	0	1	4 seconds
1	1	0	8 seconds
1	1	1	16 seconds

Table 14-6. Doze Timer reset

14.2.4. SYSTEM ACTIVITY ENABLE REGISTER 0

This is the first of the three registers that control which system activity to detect.

Sys_Activ_en0 Access = 0022h/0023h Regoffset = 062h

7	6	5	4	3	2	1	0
DRQ	PCIM	PIO	SIO	KBD	FDC	HDC	Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**DRQ**	**DMA Request (DRQ).**
Bit 6	**PCIM**	**PCI master device (PCIM).**
Bit 5	**PIO**	**Parallel IO (PIO).**
Bit 4	**SIO**	**Serial IO (SIO).**
Bit 3	**KBD**	**Keyboard (KBD).**
Bit 2	**FDC**	**Floppy Disk Controller (FDC).**
Bit 1	**HDC**	**Hard Disk Controller (HDC).**
Bit 0	**Rsv**	**Reserved.**

Programming notes

When detected, the power-down timers will reload with their initial time values or if enabled via SMI control register, a SMI interrupt will be generated or if programmed for auto-power down mode and in Doze or Standby power-down states, transition to power-on state will take place. Set the following bits to ‘1’ to detect the associated activity, and to ‘0’ to ignore the associated activity.

14.2.5. SYSTEM ACTIVITY ENABLE REGISTER 1

This is the second of the three registers that control which system activity to detect.

Sys_Activ_en1 Access = 0022h/0023h Regoffset = 063h

7	6	5	4	3	2	1	0
Rsv		AR0	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be programmed to ‘0’.
Bit 5	**AR0**	**Address range 0.**
Bits 4-0	**Rsv**	**Reserved.** Must be programmed to ‘0’.

14.2.6. SYSTEM ACTIVITY ENABLE REGISTER 2

This is the third of the three registers that control which system activity to detect.

Sys_Activ_en2 Access = 0022h/0023h Regoffset = 064h

7	6	5	4	3	2	1	0
IRQ15-1	IRQ0	NMI	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**IRQ15-1**	**IRQ15-1** detection enabled.
Bit 6	**IRQ0**	**IRQ0** detection enabled.
Bit 5	**NMI**	**NMI** detection enable.
Bits 4-0	**Rsv**	**Reserved.**

14.2.7. HOUSE-KEEPING ACTIVITY ENABLE REGISTER 0

This register controls which house-keeping activity to detect. House-keeping activities are detected only in Doze and Standby states. If enabled, a house-keeping activity reverts the system back to power-on state for a short period of time programmed in the house-keeping timer. Set the following bits to a ‘1’ to enable activity detection and a ‘0’ to ignore the associated activity.

HK_Activ_en0 Access = 0022h/0023h Regoffset = 065h

7	6	5	4	3	2	1	0
DRQ	PCI MD	KBD	IRQ15-1	IRQ0	NMI	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**DRQ**	**DMA Request (DRQ)** activity
Bit 6	**PCI MD**	**PCI master** device activity
Bit 5	**KBD**	**Keyboards** (KBD) activity
Bit 4	**IRQ15-1**	**IRQ15-1** activity
Bit 3	**IRQ0**	**IRQ0** activity
Bit 2	**NMI**	**NMI** activity
Bits 1-0	**Rsv**	**Reserved.**

14.2.8. HOUSE-KEEPING ACTIVITY ENABLE REGISTER 1

This is the second house-keeping activity detection enable register.

HK_Activ_en1 Access = 0022h/0023h Regoffset = 066h

7	6	5	4	3	2	1	0
Rsv		AR0	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be programmed to ‘0’.
Bit 5	**AR0**	**Address range 0**.
Bits 4-0	**Rsv**	**Reserved.** Must be programmed to ‘0’.

14.2.9. PERIPHERAL INACTIVITY DETECTION REGISTER 0

This register controls which peripheral inactivity is enabled for generating a SMI interrupt on a peripheral time-out.

Perif_Inact0			Access = 0022h/0023h				Regoffset = 067h
7	6	5	4	3	2	1	0
PIO	SIO	KBD	FDC	HDC	ARO	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**PIO**	**Parallel IO (PIO)** activity.
Bit 6	**SIO**	**Serial IO (SIO)** activity.
Bit 5	**KBD**	**Keyboard (KBD)** activity.
Bit 4	**FDC**	**Floppy Disk Controller (FDC)** activity.
Bit 3	**HDC**	**Hard Disk Controller (HDC)** activity.
Bit 2	**AR0**	**Address range 0.**
Bits 1-0	**Rsv**	**Reserved.** Must be programmed to ‘0’.

Programming notes

Lack of peripheral activity for an enabled peripheral for one peripheral time-out period generates SMI interrupt. A ‘1’ in a bit position enables the SMI generation for associated peripheral and a ‘0’ disables it. Software can use Peripheral Inactivity status register to determine which peripheral should be powered down.

14.2.10. PERIPHERAL ACTIVITY DETECTION REGISTER 0

This register controls which peripheral accesses will cause a SMI.

Perif_Act0 Access = 0022h/0023h Regoffset = 069h

7	6	5	4	3	2	1	0
PIO	SIO	KBD	FDC	HDC	ARO	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**PIO**	**Parallel port (PIO)** access
Bit 6	**SIO**	**Serial port (SIO)** access
Bit 5	**KBD**	**Keyboard (KBD)** access
Bit 4	**FDC**	**Floppy Disk Controller (FDC)** access
Bit 3	**HDC**	**Hard Disk Controller (HDC)** access
Bit 2	**AR0**	**Address range 0**
Bits 1-0	**Rsv**	**Reserved.** Must be programmed to ‘0’

Programming notes

Typically the power management software will detect non-usage of a peripheral device via Peripheral inactivity status registers, bring the peripheral into power down state and then enable trapping access to that peripheral via this register.

Thus when an application attempts to make use of a powered down peripheral, the access is trapped and a SMI interrupt is generated to allow software to re-power the peripheral device before allowing the access to complete. This is register is first of the two such registers.

A ‘1’ in a bit position enables SMI generation for the associate peripheral and a ‘0’ disables.

14.2.11. PERIPHERAL ACTIVITY DETECTION REGISTER 1

This is the second register that controls which peripheral accesses will cause a SMI interrupt. This register is similar in functionality to Peripheral Activity detection register 0.

Perif_Act1 Access = 0022h/0023h Regoffset = 06Ah

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be programmed to ‘0’.

14.2.12. ADDRESS RANGE 0 REGISTER 0

This register contains bits which are compared with PCI address bits 31-24 if range compare is enabled for memory cycle or compared against bits 15-8 if range compare is enabled for IO cycles.

Add_Rang0-0 Access = 0022h/0023h Regoffset = 06Bh

7	6	5	4	3	2	1	0
Default value after reset = 00h							

14.2.13. ADDRESS RANGE 0 REGISTER 1

Add_Rang0-1 Access = 0022h/0023h Regoffset = 06Ch

7	6	5	4	3	2	1	0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-3		These bits are compared with PCI address bits 23-19 if range compare is enabled for memory cycle or compared against bits 7-3 if range compare is enabled for IO cycles.
Bit 2		This bit is compared with PCI address bit 18 if range compare is enabled for memory cycle or compared with address bit 2 if range compare is enabled for IO cycles and range is 4-Bytes. Otherwise this bit when 1 specifies that the range of IO address to be compared is 16-Bytes and when 0, the range is 8-Bytes.
Bit 1		This bit is compared with PCI address bit 17 if range compare is enabled for memory cycle. Otherwise if range compare is enabled for IO cycles, this bit if 1 specifies that the range of IO address to be compared is 8/16-Bytes and when 0 the range is 4-Bytes.
Bit 0		This bit when ‘1’ specifies that range compare should be done for memory cycles and when ‘0’, for IO cycles.

14.2.14. SMI CONTROL REGISTER 0

This register controls the generation of SMI interrupt as follows:

SMI_Cont0 Access = 0022h/0023h Regoffset = 071h

7	6	5	4	3	2	1	0
							Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7		If ‘1’ then generate SMI on Doze time-out. Otherwise if set to a ‘0’, the hardware will transition to Doze state automatically on Doze time-out.
Bit 6		If ‘1’ then generate SMI on Standby time-out. Otherwise if set to a ‘0’, the hardware will transition to Standby state automatically on Standby time-out.
Bit 5		If ‘1’ then generate SMI on Suspend time-out. Otherwise if set to a ‘0’, SMI is not generated. The hardware never transitions into Suspend state by itself.
Bit 4		If ‘1’ then generate SMI on House-keeping time-out. Otherwise if set to a ‘0’, the hardware will automatically transition back to the doze or standby state (which ever state it was in before entering house-keeping state).
Bit 3		If ‘1’ then generate SMI on detecting a house-keeping activity. Otherwise if set to a ‘0’, and if in Doze or Standby state, the hardware will automatically transition to the associated house_keeping states for the duration programmed in the house-keeping timer.
Bit 2		If ‘1’ then generate SMI on detecting a system activity. Otherwise if set to a ‘0’, and if in Doze or Standby state, the hardware will automatically transition to Power-on state on detecting a unmasked system activity. This bit will typically be set to a 1’ by software on entering a power-down state so that a system activity can wake up the system.
Bit 1		This is a write only bit. Setting this bit to a ‘1’ sets bit-7 of the SMI status register 1 and generates a SMI interrupt. This bit however will always read back as ‘0’.
Bit 0	**Rsv**	**Reserved.**

14.2.15. SMI STATUS REGISTER 0

This register contains the status information pertaining to the SMI interrupt.

SMI_Stat0 Access = 0022h/0023h Regoffset = 073h

7	6	5	4	3	2	1	0
DTO	STO	STO	HKT	HKA	SAD	PID	PAD
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**DTO**	**Doze time-out.** This bit is set to a ‘1’ when Doze time-out occurs. An SMI interrupt will be generated if associated SMI enable bit in SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#. If SMI generation has been disabled then the controller will automatically transition to Doze state. This bit will then be cleared on transition from Doze or Standby to Power-on state.
Bit 6	**STO**	**Standby time-out.** This bit will be set to a ‘1’ when Standby time-out occurs. A SMI interrupt will be generated if the associated SMI enable bit in SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#. If SMI generation has been disabled then the hardware will automatically transition to Standby state. This bit will then be cleared on transition Standby to Power-on state.
Bit 5	**STO**	**Suspend time-out.** This bit will be set to a ‘1’ when Suspend time-out occurs. A SMI interrupt will be generated if the associated SMI enable bit in the SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#.
Bit 4	**HKT**	**House-keeping timeout detected.** This bit will be set to a ‘1’ if the controller is in one of the house-keeping states and the house-keeping timer expires. A SMI interrupt will be generated if the associated SMI enable bit in the SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#. If the SMI generation has been disabled, the hardware will automatically transition to doze or standby state. This bit then will be cleared on transition from Doze or Standby states to any other state.
Bit 3	**HKA**	**House-keeping activity detected.** This is a read-only bit and represents the OR of the System activity status registers masked (ANDed) with the corresponding bits in the House-keeping Activity enable registers. A SMI interrupt will be generated when this bit is a 1’ and if the associated SMI enable bit in the SMI Control register is set to a ‘1’. The software can refer to Activity status register to determine the cause of the interrupt. The software must clear the corresponding bits of the Activity Status register to deassert SMI#. If SMI generation has been disabled and if the controller in Doze or Standby state, it will automatically transition to House-keeping state.

Bit Number	Mnemonic	Description
Bit 2	SAD	**System Activity detected**. This is a read-only bit and represents the OR of the System activity Status registers masked (ANDed) with the corresponding bits of the System Activity enable registers. A SMI interrupt will be generated if this bit is a ‘1’ and if the associated SMI enable bit in SMI Control register is set to a ‘1’. The software can refer to Activity status register to determine the cause of this interrupt. The software must clear the System Activity Status registers bits for the enabled system activities to deassert SMI#. If SMI generation has been disabled and if the controller is in Doze or Standby state, it will automatically transition to Power-on state.
Bit 1	PID	**Peripheral Inactivity detected.** This is a read-only bit and represents the OR of Peripheral Inactivity Status register bits masked (ANDed) with the associated Peripheral inactivity detection register bit. A SMI# interrupt will be generated when this bit is a ‘1’. The software can refer to Peripheral Inactivity status registers to determine which peripheral should be powered down. The software must clear the corresponding bits of the Peripheral Inactivity detection register to deassert SMI#.
Bit 0	PAD	**Peripheral Activity Detected.** This is a read-only bit and represents the OR of the System activity Status register masked (ANDed) with the corresponding bits of the Peripheral Activity detection registers. A SMI interrupt will be generated when this bit is a ‘1’. The software can refer to the System Activity status register to determine which peripheral caused the interrupt. The software must clear the corresponding bits of the System activity register to deassert SMI#.

Programming notes

The SMI# output is a logical OR of all the bits (ANDed with their respective SMI generation enable bits) in this register. SMI# output will be deasserted within 3 PCI clocks after the cause of the SMI# interrupt is cleared.
This register defaults to 00h after reset deasserting SMI# output.

14.2.16. SMI STATUS REGISTER 1

This register is similar to SMI Status register 0 in that it reports the cause of the SMI interrupt to the software.

SMI_Stat1 Access = 0022h/0023h Regoffset = 074h

7	6	5	4	3	2	1	0
S SMI	Rsv						
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**S SMI**	**Software SMI.** This bit is set to a ‘1’ by write writing a ‘1’ in bit-1 of the SMI Control register. The software must clear this bit to deassert SMI#.
Bits 6-0	**Rsv**	**Reserved.**

14.2.17. PERIPHERAL INACTIVITY STATUS REGISTER 0

This register contains a ‘1’ in a bit position if the associated peripheral was inactive for the entire duration of the last peripheral time-out period.

Perif_Stat0 Access = 0022h/0023h Regoffset = 075h

7	6	5	4	3	2	1	0
PIO	SIO	KBD	FDC	HDC	ARO	Rsv	
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**PIO**	**Parallel port (PIO)** activity.
Bit 6	**SIO**	**Serial port (SIO)** activity.
Bit 5	**KBD**	**Keyboard (KBD)** activity.
Bit 4	**FDC**	**Floppy Disk Controller (FDC)** activity.
Bit 3	**HDC**	**Hard Disk Controller (HDC)** activity.
Bit 2	**AR0**	**Address range 0**
Bits 1-0	**Rsv**	**Reserved.**

It can also be cleared by software by writing a ‘1’ in the bit which is set to ‘1’.

Programming notes

A bit in this register is set to a ‘1’ only at peripheral timer time-out. It is set to a ‘0’, as soon as an activity from the associated peripheral is detected.

The status reflected in this register is not conditioned by whether the peripheral was enabled for inactivity detection through the Peripheral Inactivity Detection registers or not. The SMI interrupt however will be generated only if any of the enabled peripherals (via Peripheral Inactivity Enable register) were inactive for the entire duration of the peripheral time out.

14.2.18. ACTIVITY STATUS REGISTER 0

This register records presence of activity.

Activ_Stat0 Access = 0022h/0023h Regoffset = 077h

7	6	5	4	3	2	1	0
DRQ	PCIM	PIO	SIO	KBD	FDC	HDC	Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**DRQ**	**DMA Request (DRQ)** activity.
Bit 6	**PCIM**	**PCI master device (PCIM)** activity.
Bit 5	**PIO**	**Parallel IO (PIO)** activity.
Bit 4	**SIO**	**Serial IO (SIO)** activity.
Bit 3	**KBD**	**Keyboard (KBD)** activity.
Bit 2	**FDC**	**Floppy Disk Controller (FDC)** activity.
Bit 1	**HDC**	**Hard Disk Controller (HDC)** activity.
Bit 0	**Rsv**	**Reserved.**

Programming notes

A ‘1’ in a bit position indicates that presence of the associated activity since the bit was last cleared. Once set, a bit of this register can only be cleared by software writing a ‘1’ to it or by reset or if auto power management is enabled then any transition to Doze or Standby state (including the ones from house-keeping states) will clear all enabled System and House-keeping activities.

The status reflected in this register is not conditioned by the settings of System Activity Enable, House-keeping Activity Enable, Peripheral Inactivity or Peripheral Activity Detection registers.

14.2.19. ACTIVITY STATUS REGISTER 1

This register is similar to Activity Status register 0. It contains the status for the following bits.

Activ_Stat1 Access = 0022h/0023h Regoffset = 078h

7	6	5	4	3	2	1	0
Rsv		AR0	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-6	**Rsv**	**Reserved.** Must be programmed to ‘0’
Bit 5	**AR0**	**Address range 0**
Bits 4-0	**Rsv**	**Reserved.** Must be programmed to ‘0’

14.2.20. ACTIVITY STATUS REGISTER 2

This register is similar to Activity Status registers 0 and 1.

Activiv_Stat2 Access = 0022h/0023h Regoffset = 079h

7	6	5	4	3	2	1	0
IRQ15-1	IRQ0	NMI	Rsv				
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**IRQ15-1**	**IRQ15-1** activity.
Bit 6	**IRQ0**	**IRQ0** activity.
Bit 5	**NMI**	**NMI** activity.
Bits 4-0	**Rsv**	**Reserved.**

14.2.21. PMU STATUS REGISTER

This register contains the state the power management controller currently is in.

PMU Access = 0022h/0023h Regoffset = 07Ah

7	6	5	4	3	2	1	0
Rsv	PMU	PMU	PMU	PMU	PMU	PMU	PMU
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved.**
Bit 6	**PMU**	**PMU microsecond clock test mode.** This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the micro-second clock to tick at oscillator clock frequency instead of every microsecond.
Bit 5	**PMU**	**PMU millisecond clock test mode.** This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the millisecond clock to tick at oscillator clock frequency instead of every millisecond.
Bit 4	**PMU**	**PMU second clock test mode.** This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the second clock to tick at oscillator clock frequency instead of every second.
Bit 3	**PMU**	**PMU minute clock test mode.** This bit is for factory use only and must be set to a '0' by the software. Setting this bit to a 1' causes the minute clock to tick at oscillator clock frequency instead of every minute.
Bit 2	**PMU**	**PMU state** (see table below: Table 14-7).
Bit 1	**PMU**	**PMU state** (see table below: Table 14-7).
Bit 0	**PMU**	**PMU state** (see table below: Table 14-7).

Bit 2	Bit 1	Bit 0	PMU state
0	0	0	Power-on
0	0	1	Doze
0	1	0	Standby
0	1	1	Suspend
1	0	1	Doze_house_keeping
1	1	0	Standby_house_keeping
1	1	1	Reserved

Table 14-7. PMU state

The architecture allows for either the software to explicitly program the power-down state the controller should be in or the controller can change states automatically (auto-power down mode of operation) or a mix of the two. Some power-down states are entered and exited automatically by the hardware while the others require software assist. This is based on the SMI Control register settings as follows:

Transition from Power-on to Doze state will take place automatically on Doze time-out, if bit-7 of the SMI control register is set to a ‘0’. Otherwise if bit-7 is programmed to be a ‘1’, an SMI will be generated instead and the software can change the state to Doze.

Transition from Doze to Power-on will take place automatically in presence of an enabled system activity if bit-2 of the SMI control register is programmed to be a 0’. Otherwise if bit-2 is programmed to a ‘1’, an SMI interrupt will be generated instead and software can change the state to Power-on.

Transition from Doze to Doze_house_keeping state will take place automatically if an enabled house_keeping activity is detected and bit-4 of the SMI control register is set to a ‘0. Otherwise if bit-7 is programmed to be a ‘1’, an SMI interrupt will be generated instead.

Transition from Doze_house_keeping state to Doze will take place automatically on house-keeping time-out if bit-3 of the SMI control register is set to a ‘0’. Otherwise an SMI interrupt will be generated instead.

Transition from Doze_house_keeping state to Power-on will take place on detecting an enabled system activity automatically if bit-2 of the SMI control register is programmed to ‘0’. Otherwise an SMI interrupt will be generated instead.

Transitions from Doze or Power-on state to Standby will take place automatically on standby time-out if bit-6 of the SMI control register is set to a ‘0’. Otherwise an SMI interrupt will be generated instead.

Transition from Standby to Power-on will take place automatically in presence of an enabled system activity if bit-2 of the SMI control register is programmed to be a ‘0’. Otherwise if bit-2 is programmed to a ‘1’, SMI interrupt will be generated instead and software can change the state to Power-on.

Transition from Standby to Standby_house_ keeping state will take place automatically if an enabled house_keeping activity is detected and bit-4 of the SMI control register is set to a ‘0’. Otherwise if bit-7 is programmed to be a ‘1’, a SMI interrupt will be generated instead.

Transition from Standby_house_keeping state to Standby will take place automatically on house-keeping time-out if bit-3 of the SMI control register is set to a ‘0’. Otherwise SMI interrupt will be generated instead.

Transition from Standby_house_keeping state to Power-on will take place on detecting an enabled system activity automatically if bit-2 of the SMI control register is programmed to ‘0’. Otherwise SMI interrupt will be generated instead.

The hardware never transitions to Suspend state automatically.

The power saving features associated with each power-down state are independent of how the state was entered.

14.2.22. GENERAL PURPOSE REGISTER

This is a read/write IO register that can be used by software.

GP Access = 0022h/0023h Regoffset = 07Bh

7	6	5	4	3	2	1	0
GP	GP	GP	GP	GP	GP	GP	GP
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bit 7	**GP**	**General Purpose Register Bit 7.**
Bit 6	**GP**	**General Purpose Register Bit 6.**
Bit 5	**GP**	**General Purpose Register Bit 5.**
Bit 4	**GP**	**General Purpose Register Bit 4.**
Bit 3	**GP**	**General Purpose Register Bit 3.**
Bit 2	**GP**	**General Purpose Register Bit 2.**
Bit 1	**GP**	**General Purpose Register Bit 1.**
Bit 0	**GP**	**General Purpose Register Bit 0.**

Programming notes

Writing to this register also updates the external ‘373 latch that can be used to control external devices for power-down purposes. Reads of this register return the value of this internal register.

The GPIOCS# signal will be asserted when writing to this register to latch the data on the ISA data bus.

14.2.23. CLOCK CONTROL REGISTER 0

This register allows control over power saving via stop clock modulation. The power-saving can be tuned to the power-management state the PMU is in.

Clk_Cont0 Access = 0022h/0023h Regoffset = 07Ch

7	6	5	4	3	2	1	0
STPCLK			DSSS			STPCLK	Rsv
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-5	**STPCLK**	**Power-on and housekeeping states** STPCLK# modulation control. These bits control the duty cycle of STPCLK# deassertion when the PMU is in Power-on or one of the house-keeping states as follows in Table 14-8. The STPCLK# is deasserted and the duty-cycle control ignored if a SMI interrupt is pending.
Bit 4-2	**DSSS**	**Doze/Standby/Suspend states** STPCLK# modulation control. These bits control the duty cycle of the STPCLK# deassertion when PMU is in one of the power-down states as follows in Table 14-9. The STPCLK# is deasserted and the duty-cycle control ignored if a SMI interrupt is pending.
Bit 1	**STPCLK**	**STPCLK# modulation period.** If ‘1’ then the period is 64ms else, if ‘0’, then the period is 64ms.
Bit 0	**Rsv**	**Reserved.**

Bit 7	Bit 6	Bit 5	Ratio	Power-on STPCLK# Modulation
0	0	0	1	STPCLK# is never asserted
0	0	1	1/2	1 half period
0	1	0	1/4	1 quarter period
0	1	1	1/8	one-eighth period
1	0	0	1/16	one-sixteenth period
1	0	1	1/32	1/32 period
1	1	0	1/64	1/64 period
1	1	1		*Reserved.*

Table 14-8. Power-on and housekeeping states

Bit 4	Bit 3	Bit 2	Ratio	Doze STPCLK# Modulation
0	0	0	1	$\overline{\text{STPCLK}}$ is never asserted
0	0	1	1/2	1 half period
0	1	0	1/4	1 quarter period
0	1	1	1/8	one-eighth period
1	0	0	1/16	one-sixteenth period
1	0	1	1/32	1/32 period
1	1	0	1/64	1/64 period
1	1	1	0	The entire period

Table 14-9. Doze/Standby/Suspend states

14.2.24. DOZE TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit doze timer.

Doze Access = 0022h/0023h Regoffset = 088h

7	6	5	4	3	2	1	0
Default value after reset = 00h							

Bit Number	Mnemonic	Description
Bits 7-0		Bits 8-1 of the current value of the doze timer.

Programming notes

This register should not be used by the software.

Note that bit 0 of the current value of the doze timer is not readable.

14.2.25. STANDBY TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of 5-bit standby timer.

Standby Access = 0022h/0023h Regoffset = 089h

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-5	**Rsv**	**Reserved**.
Bits 4-0		Bits 4-0 of the current value of the standby timer.

Programming notes

This register should not be used by software.

14.2.26. SUSPEND TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the 7-bit Suspend timer.

Suspend Access = 0022h/0023h Regoffset = 08Ah

7	6	5	4	3	2	1	0
Rsv							
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bit 7	**Rsv**	**Reserved**.
Bits 6-0		Bits 6-0 of the current value of the suspend timer.

Programming notes

This register should not be used by software.

14.2.27. HOUSE-KEEPING TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit house-keeping timer.

HK_Timer Access = 0022h/0023h Regoffset = 08Bh

7	6	5	4	3	2	1	0
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0		Bits 8-1 of the house-keeping timer.

Programming notes

This register should not be used by software.

14.2.28. PERIPHERAL TIMER READ BACK REGISTER

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit Peripheral timer.

Perif_Timer Access = 0022h/0023h Regoffset = 08Ch

7	6	5	4	3	2	1	0
Default value after reset = undefined							

Bit Number	Mnemonic	Description
Bits 7-0		Bits 8-1 of the Peripheral timer.

Programming notes

This register should not be used by software.

14.3 UPDATE HISTORY FOR POWER MANAGEMENT CHAPTER

The following changes have been made to the Power Management Chapter.

Section	Change	Text
14.2.6.	Added	“This register defaults to 00h disabling all activity detection.”
14.2.17.	Removed	This register is not implemented in this revision.