iDragon SCX501 Programming Manual

Release A

May 9, 2000

RiSE Technology Company

Information provided is believed to be accurate and reliable. However, RiSE Technology assumes no responsibility for the consequences of use of such information nor for any infringements of patents or other rights of third parties which may result from its use. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied.

1 TABLE OF CONTENTS

2 LIST OF TABLES

LIST OF TABLES

3 LIST OF FIGURES

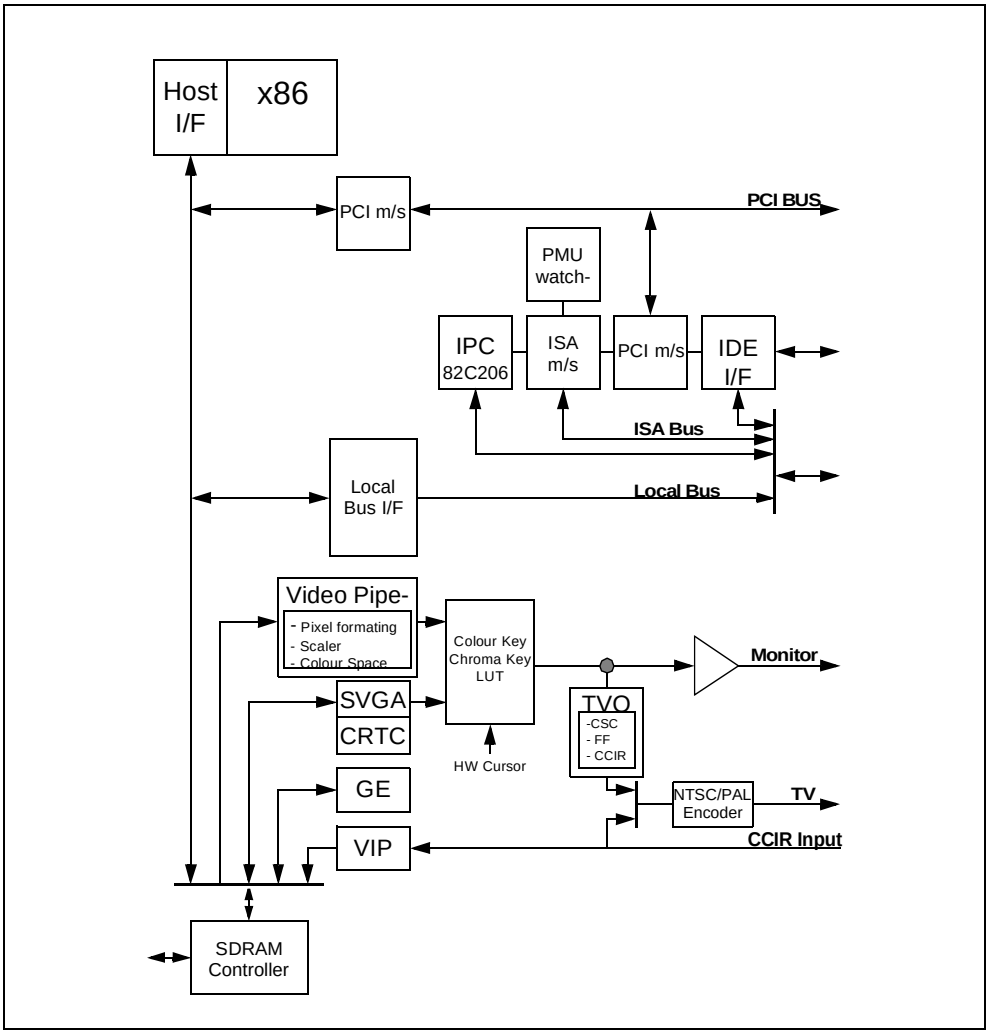
4. INTRODUCTION

This programming manual details the register sets for the iDragon SCX501 Device. The manual is split into chapters each dedicated to a function.

This documents contains all the information required to program and configure the iDragon SCX501.

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 iDragondevice. 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 iDragondevice, 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 hexadeci mal 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+6	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 external software.

Section	Register Name	Mnemonic	Purpose	Address	Access type
3.	**Power on strap Registers**			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 Strap register	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_cont0	Configuration		Index 025h
7.6.3	Shadow Control Register 1	Shadow_cont1	Configuration		Index 026h
7.6.4	Shadow Control Register 2	Shadow_cont2	Configuration		Index 027h
7.6.5	Shadow Control Register 3	Shadow_cont3	Configuration		Index 028h
7.6.6	VGA Decode Register	VGA_dec	Configuration		Index 029h
7.7	**Host SDRAM controller registers**			0022h	
7.7.1	SDRAM Bank 0 Register	SDRAM_bank0	Configuration	0023h	Index 030h
7.7.2	SDRAM Bank 1 Register	SDRAM_bank1	Configuration		Index 031h
7.7.3	SDRAM Bank 2 Register	SDRAM_bank2	Configuration		Index 032h
7.7.4	SDRAM Bank 3 Register	SDRAM_bank3	Configuration		Index 033h
7.7.5	Memory Bank Width Register	Mem_width	Configuration		Index 034h
7.7.6	Graphics Memory Size Register	Graph_mem	Configuration		Index 036h
7.7.7	Presents Detect Register	Pres_dect	Configuration		Index 097h
7.	**Memory Interface**			GBase+4C6000h	
7.8.1	Register 1	MEM_REG1	Configuration		000h
7.8.2	Register 2	MEM_REG2	Configuration		004h
7.9	**MCLK Control Registers**			22h	
7.9.1	MCLK Control Register 0	MCLK00		23h	Index 0x40h
7.9.2	MCLK Control Register 1	MCLK01			Index 0x41h
10.5.	**North Bridge Config Registers**				
10.3.	Configuration Address Register	Config_ address	IO	0xCF8h	
10.4.	Configuration Data Register	Config_data	IO	0xCFC - CFFh	
10.5.1.	North Bridge Vendor Identification Register	NB_Vend_ID	PCI Config		Index 0x0h

Section	Register Name	Mnemonic	Purpose	Address	Access type
10.5.2.	North Bridge Device Identification Register	NB_Device_ID	PCI Config		Index 0x2h
10.5.3.	North Bridge PCI Command Register	NB_Command	PCI Config		Index 0x4h
10.5.4.	North Bridge PCI Status Register	NB_Status	PCI Config		Index 0x6h
10.5.5.	North Bridge PCI Revision Id Register	NB_Rev_ID	PCI Config		Index 0x8h
10.5.6.	North Bridge Device Class Code Register	NB_Class_ Code	PCI Config		Index 0x9h
10.5.7.	North Bridge Header Type Register	NB_HEAD	PCI Config		Index 0x0Eh
10.5.8.	North Bridge Control Register	NB_Control	PCI Config		Index 0x50h
10.5.9.	North Bridge PCI Error Status Register	NB_Error_ Status	PCI Config		Index 0x54h
10.7.	**South Bridge PCI Function 0 Configuration Registers**			0xCF8h	
10.7.1.	South BridgeVendor Identification Register	SB_Vend_ID0	PCI config F#0	0xCFCh	Index 0x0h
10.7.2.	South Bridge Device Identification Register	SB_Device_ ID0	PCI Config F#0		Index 0x2h
10.7.3.	South Bridge PCI Command Register	SB_ Command0	PCI Config F#0		Index 0x4h
10.7.4.	South Bridge PCI Status Register	SB_Status0	PCI Config F#0		Index 0x6h
10.7.5.	South Bridge PCI Revision Id Register	SB_Rev_ID0	PCI Config F#0		Index 0x8h
10.7.6.	South Bridge Device Class Code Register	SB_Class_ Code0	PCI Config F#0		Index 0x9h
10.7.7.	South Bridge Header Type Register	SB_Header0	PCI Config F#0		Index 0xEh
10.7.8.	South Bridge Miscellaneous Register	SB_Misc0			Index 040h
10.8.	**South Bridge PCI Function 1 Configuration Registers**			0xCF8h	
10.8.1.	South Bridge Vendor Identification Register	SB_Vend_ID1	PCI config F#1	0xCFCh	Index 0x0h
10.8.2.	South Bridge Device Identification Register	SB_Device_ ID1	PCI Config F#1		Index 0x2h
10.8.3.	South Bridge PCI Command Register	SB_ Command1	PCI Config F#1		Index 0x4h
10.8.4.	South Bridge PCI Status Register	SB_Status1	PCI Config F#1		Index 0x6h
10.8.5.	South Bridge Revision ID Register	SB_Rev_ID1	PCI Config F#1		Index 0x8h
10.8.6.	South Bridge Programming Interface Register	Prog_Interf	PCI Config F#1		Index 0x9h
10.8.7.	South Bridge Sub-Class Code Register	Sub_Class	PCI Config F#1		Index 0xAh
10.8.8.	South Bridge Base-Class code Register	Base_Class	PCI Config F#1		Index 0xBh
10.8.9.	South Bridge Latency Timer control Register	Latency_Timer	PCI Config F#1		Index 0xDh

Section	Register Name	Mnemonic	Purpose	Address	Access type
10.8.10.	South Bridge Header Type Register	Header_Time	PCI Config F#1		Index 0xEh
10.8.11.	South Bridge Base Address 0 Register	Base0	PCI Config F#1		Index 0x10h
10.8.12.	South Bridge Base Address 1 Register	Base1	PCI Config F#1		Index 0x14h
10.8.13.	South Bridge Base Address 2 Register	Base2	PCI Config F#1		Index 0x18h
10.8.14.	South Bridge Base Address 3 Register	Base3	PCI Config F#1		Index 0x1Ch
10.8.15.	South Bridge Base Address 4 Register	Base4	PCI Config F#1		Index 0x20h
10.8.16.	South Bridge Primary IDE Timing Register	Prime_IDE_Time	PCI Config F#1		Index 0x40h
10.8.17.	South Bridge Secondary IDE Timing Register	Second_IDE_Time	PCI Config F#1		Index 0x44h
10.8.18.	South Bridge Miscellaneous Register	SB_Misc1	PCI Config F#1		Index 0x48h
9.5	**ISA standard Registers**				
9.5.1	DMA 1 Controller Registers	DMA_1	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	
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	

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.5.3	Interval Timer Registers	IT_1	IO	0040h	
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 B Register	Port _B	IO	0061h	
9.5.5	Port 60h Register	Port_60	IO	0060h	
9.5.5	Port 64h Register	Port_64	IO	0064h	
9.5.6	Port 70 Register	Port_70	IO	0070h	
9.5.7	Interrupt Controller 2 Registers	IC_2	IO	00A0h	
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	
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	

Section	Register Name	Mnemonic	Purpose	Address	Access type
9.5.9	DMA Page Register Port 84h	Port_84	IO	0084h	(Reserved)
9.5.9	DMA Page Register Port 85h	Port_85	IO	0085h	(Reserved)
9.5.9	DMA Page Register Port 86h	Port_86	IO	0086h	(Reserved)
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	Port_8B	IO	008Ch	(Reserved)
9.5.9	DMA Page Register Port 8Ch	Port_8C	IO	008Dh	(Reserved)
9.5.9	DMA Page Register Port 8Dh	Port_8D	IO	008Eh	(Reserved)
9.5.9	DMA Page Register Port 8Eh	Port_8E	IO	008Fh	(Reserved)
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
9.6.6	PIRQC Routing control Register 0	PCR_Cont0	Configuration		Index 054h
9.6.7	PIRQD Routing control Register 0	PDR_Cont0	Configuration		Index 055h
9.6.8	Interrupt Level Control Register 0	IRQ_Lev_Cont 0	Configuration		Index 056h
9.6.9	Interrupt Level Control Register 1	IRQ_Lev_Cont 1	Configuration		Index 057h
9.6.10	IPC Configuration Register	IPC_Conf	Configuration		Index 001h
9.6.11	VMI IRQ Routing Control Register	VIR_Cont	Configuration		Index 058h
9.6.12	ISA Syncronizer Bypass Register	ISA_Sync	Configuration		Index 059h
11.3	**VGA registers**				
11.4	**General Registers**				
11.4.1	Motherboard Enable Register	MBEN		0x094h	
11.4.2	Add-in VGA Enable Register	ADDEN		0x46E8h	
11.4.3	Video Subsystem Enable 1 Register	VSE1		0x102h	
11.4.4	Video Subsystem Enable 2 Register	VSE2		0x3C3h	
11.4.5	Miscellaneous Output Register	MISC		0x3CC/ 0x3C2h	
11.4.6	Input Status Register #0	INP0		0x3C2h	
11.4.7	Input Status Register #1	INP1		0x3XAh	
11.5	Sequencer Registers				
11.5.1	Sequencer Index Register	SRX		0x03C4h	
11.5.2	Sequencer Reset Register	SR0		0x03C5h	Index 000h
11.5.3	Sequencer Clocking Mode Register	SR1			Index 001h
11.5.4	Sequencer Plane Mask Register	SR2			Index 002h
11.5.5	Sequencer Character Map Register	SR3			Index 003h
11.5.6	Sequencer Memory Mode Register	SR4			Index 004h

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.5.7	Extended Register Lock/Unlock Register	SR6			Index 006h
11.6	Graphics Controller Registers				
11.6.1	Graphics Controller Index Register	GRX		0x03CEh	
11.6.2	Graphics Set/Reset Register	GR0		0x03CFh	Index 000h
11.6.3	Graphics Enable Set/Reset Register	GR1			Index 001h
11.6.4	Graphics Color Compare Register	GR2			Index 002h
11.6.5	Raster Op/Rotate Count Register	GR3			Index 003h
11.6.6	Graphics Read Map Select Register	GR4			Index 004h
11.6.7	Graphics Mode Register	GR5			Index 005h
11.6.8	Graphics Miscellaneous Register	GR6			Index 006h
11.6.9	Graphics Color Don't Care Register	GR7			Index 007h
11.6.10	Graphics Bit Mask Register	GR8			Index 008h
11.7	Attribute Controller Registers				
11.7.1	Attribute Controller Index Register	ARX		0x3C0h	
11.7.2	Attribute Palette Registers	AR0 - ARF		0x3C1/ 0x3C0h	
11.7.3	Attribute Ctrl Mode Register	AR10		0x3C1/ 0x3C0h	
11.7.4	Attribute Ctrl Overscan Color Register	AR11		0x3C1/ 0x3C0h	
11.7.5	Attribute Color Plane Enable Register	AR12		0x3C1/ 0x3C0h	
11.7.6	Attribute Horz Pixel Panning Register	AR13		0x3C1/ 0x3C0h	
11.7.7	Attribute Color Select Register	AR14		0x3C1/ 0x3C0h	
11.8	**CRT Controller Registers**				
11.8.1	Index Register	CRX	*see Note 1*	0x3X4h	
11.8.2	Horizontal Total Register	CR0	*see Note 1*	0x3X5h	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
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

Section	Register Name	Mnemonic	Purpose	Address	Access type
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*	0x3X4h	
11.9.1	Repaint Control Register 0	CR19	*see Note 1*	0x3X5h	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
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	SDRAM Timing Parameter Register	CR3C			Index 03Ch

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.9.30	SDRAM 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**				
11.12.1	Palette Pixel Mask Register	Pixel_Mask		0x3C6h	
11.12.2	Palette Read index Register	Read_index		0x3C7h	
11.12.3	Palette State Register	Palette_State		0x3C7h	
11.12.4	Palette Write Index Register	Write_index		0x3C8h	
11.12.5	Palette Data Register	Palette_data		0x3C9h	
11.13	DCLK Control Rgisters			22h	
11.13.1	DCLK Control Register 00	DCLK00	PCI Config	23h	Index 0x42h
11.13.2	DCLK Control Register 01	DCLK01	PCI Config		Index 0x43h
11.13.3	DCLK Control Register 10	DCKL10	PCI Config		Index 0x44h
11.13.4	DCLK Control Register 11	DCLK11	PCI Config		Index 0x45h
11.13.5	DCLK Control Register 20	DCLK20	PCI Config		Index 0x46h
11.13.6	DCLK Control Register 21	DCLK21	PCI Config		Index 0x47h
11.13.7	DCLK Control Register 30	DCLK30	PCI Config		Index 0x48h
11.13.8	DCLK Control Register 31	DCLK31	PCI Config		Index 0x49h
12.5	**VGA Operand Source**				
12.9.1	Back Ground Color Register	Background		8400000h	Index 0x004h
12.9.2	Cursor Coordinate Register	Cursor_XY			Index 0x11Ch
12.9.3	Top of Data FIFO Register	Data_Port			Index 0x804h
12.9.4	Destination Operand Base Address Register	Dst_base			Index 0x018h
12.9.5	Destination Pitch Register	Dst_pitch			Index 0x028h
12.9.6	Destination Operand Coordinate Register	Dst_XY		8410000h	
12.9.7	Foreground Color Register	Foreground		8400000h	Index 0x034h
12.9.8	Height Register	Height			Index 0x048h
12.9.9	Pattern Base Address Operand Register	Pattern			Index 0x058h
12.9.10	Pixel Depth Operand Register	Pixel_depth			Index 0x07Ch
12.9.11	Raster Operation Register	ROP			Index 0x08Ch
12.9.12	Source Base Address Operand Register	Src_base			Index 0x098h
12.9.13	Source Pitch Operand Register	Src_pitch			Index 0x0ACh
12.9.14	Source Coordinate Register	Src_XY			Index 0x0BDh
12.9.15	Status Register	Status			Index 0x908
12.9.16	Width Register	Width			Index 0x0C8h
12.9.17	Extra Use Register	Xtra			Index 0x0D4h

Section	Register Name	Mnemonic	Purpose	Address	Access type
12.9.18	SRC Transparency Compare Register	SRC_ transparency			Index 0xECh
12.9.19	DST Transparency Compare Register	DST_ transparency			Index 0xFCh
13.6	**Video Input Port Registers**				
13.6.1	Frame Buffer Address Readback Register	Fb1_adr		GBase +600000h	Index 0x00h
13.6.2	Video Input Port Configuration Register	Vin_cfg			Index 0x04h
13.6.3	Video Input Port Status Register	Vin_stat			Index 0x08h
13.6.4	Video Input Buffer Addr 0	Vin_ad0			Index 0x0Ch
13.6.5	Video Input Buffer Addr 1	Vin_ad1			Index 0x10h
13.6.6	Video Input Desination Pitch	Vin_dp			Index 0x14h
13.6.7	External Timing Generator 1	Vtg_ext1			Index 0x28h
13.6.8	External Timing Generator 2	Vtg_ext2			Index 0x2Ch
13.6.9	Horizontal Timing Generator	Vtg_ht			Index 0x30h
13.6.10	Video Timing Generator	Vtg_vg			Index 0x34h
13.10	**Video Output Port Register**			GBase+4CA000h	
13.10.1	Register_0.Configuration0	Config0			Index 000h
13.10.2	Register_1.Configuration1	Config1			Index 001h
13.10.3	Register_2.Configuration2	Config2			Index 002h
13.10.4	Register_3.Configuration3	Config3			Index 003h
13.10.5	Register_4.Configuration4	Config4			Index 004h
13.10.6	Register_5.Configuration5	Config5			Index 005h
13.10.7	Register_6.Configuration6	Config6			Index 006h
13.10.9	Register_9.Status	Stat9			Index 009h
13.10.10	Register_10.Increment_dfs	Incr_10			Index 010h
13.10.10	Register_11.Increment_dfs	Incr_11			Index 011h
13.10.10	Register_12.Increment_dfs	Incr_12			Index 012h
13.10.11	Register_13.Phase_dfs	Phase_13			Index 013h
13.10.11	Register_14.Phase_dfs	Phase_14			Index 014h
13.10.18	Register_21:line_reg=Itarg[8:1]	Reg21			Index 021h
13.10.18	Register_22:line_reg=Itarg[0] and Iref[8:2]	Reg22			Index 022h
13.10.18	Register_23:line_reg=Iref[1:0]	Reg23			Index 023h
13.10.19	Register_31.cgms_bit [1:4]	Reg31			Index 031h
13.10.19	Register_32.cgms_bit [5:12]	Reg32			Index 032h
13.10.19	Register_33.cgms_bit [13:20]	Reg33			Index 033h
13.10.21	Register_39.cccf1	Reg39			Index 039h
13.10.21	Register_40.cccf1	Reg40			Index 040h
13.10.22	Register_41.cccf2	Reg41			Index 041h
13.10.22	Register_42.cccf2	Reg42			Index 042h
13.10.23	Register_43.cclif1	Reg43			Index 043h
13.11	**Video Pipeline Registers**				
13.11.1	Source Specification Registers		*see Note 2*	X600000h	
13.11.1.1	Video Source Base Register	Video_Src_ Base			Index 0x0h

Section	Register Name	Mnemonic	Purpose	Address	Access type
13.11.1.2	Video Source Pitch Register	Video_Src_ Pitch			Index 0x4h
13.11.1.3	Video Source Dimension Register	Video_Src_ dim			Index 0x8h
13.11.1.4	CRTC Burst Length Register	CRTC_Burst_ Length			Index 0xCh
13.11.1.5	Video Burst Length Register	Video_Burst_ Length			Index 0x10h
13.11.2	Destination Specification Registers				
13.11.2.1	Video Destination Register	Video_Dst_XY			Index 0x14h
13.11.2.2	Video Destination Dimension Register	Video_Dst_ dim			Index 0x18h
13.11.3	**Filter Control Register**				
13.11.3.1	Horizontal Scaling and Decimation Register	Horiz_Scale			Index 0x20h
13.11.3.2	Vertical Control and Decimation Register	Vert_Scale			Index 0x28h
13.11.3.3	Color Space Converter Specification Register	Color_Con_ Spec			Index 0x2Ch
13.11.4	Video and Graphics mixing control Registers				
13.11.4.1	Mix Mode Register	Mix_Mode			Index 0x30h
13.11.4.2	Color Key Register	Color_Key			Index 0x34h
13.11.4.3	Chroma Key Low Register	Chroma_Key_ Low			Index 0x38h
13.11.4.4	Chroma Key High Register	Chroma_Key_ High			Index 0x3Ch
13.11.4.5	Status Register	Filter_Status			Index 0x40h
14.2	**Local Bus Registers**		16 bit access	22h	
14.3	**Local Bus Address Decode Registers**			23h	
14.3.1	I/O Slot Base Register 0	IOAREG0			Index 0x00h
14.3.1	I/O Slot Base Register 1	IOAREG1			Index 0x01h
14.3.1	I/O Slot Base Register 2	IOAREG2			Index 0x02h
14.3.1	I/O Slot Base Register 3	IOAREG3			Index 0x03h
14.3.2	I/O Slot Mask Register 0	IOMREG0			Index 0x04h
14.3.2	I/O Slot Mask Register 1	IOMREG1			Index 0x05h
14.4	**Local Bus Timing Registers**				
14.4.1	Memory Timing Template 0	TIMEBANK0			Index 0x06h
14.4.1	Memory Timing Template 1	TIMEBANK1			Index 0x07h
14.4.2	I/O Timing Template 0	TIMEIO0			Index 0x08h
14.4.2	I/O Timing Template 1	TIMEIO1			Index 0x09h
14.4.2	I/O Timing Template 2	TIMEIO2			Index 0x0Ah
14.4.2	I/O Timing Template 3	TIMEIO3			Index 0x0Bh
14.5.1	**Local Bus Control Register**				
14.5.1	Control Register	CONTROL			Index 0x0Ch
14.5.2	I/O Width Register	IOWIDTH			Index 0x0Dh

Section	Register Name	Mnemonic	Purpose	Address	Access type
15.2	**Power Management Controller Registers:**			0022h	
15.2.1	Timer Register 0	Timer0	Configuration	0023h	Index 060h
15.2.2	Timer Register 1	Timer1	Configuration		Index 061h
15.2.3	Timer Register 2	Timer2	Configuration		Index 08dh
15.2.4	System Activity Enable Register 0	Sys_activ_en0	Configuration		Index 062h
15.2.5	System Activity Enable Register 1	Sys_activ_en1	Configuration		Index 063h
15.2.6	System Activity Enable Register 2	Sys_activ_en2	Configuration		Index 064h
15.2.7	House-Keeping Activity Enable Register 0	HK_activ_en0	Configuration		Index 065h
15.2.8	House-Keeping Activity Enable Register 1	HK_activ_en1	Configuration		Index 066h
15.2.9	Peripheral Inactivity Detection Register 0	Perif_inactiv0	Configuration		Index 067h
15.2.10	Peripheral Activity Detection Register 0	Perif_activ0	Configuration		Index 069h
15.2.11	Peripheral Activity Detection Register 1	Perif_activ1	Configuration		Index 06Ah
15.2.12	Address Range 0 Register 0	Address_ range0-0	Configuration		Index 06Bh
15.2.13	Address Range 0 Register 1	Address_ range0-1	Configuration		Index 06Ch
15.2.14	SMI Control Register 0	SMI_cont0	Configuration		Index 071h
15.2.15	SMI Status Register 0	SMI_stat0	Configuration		Index 073h
15.2.16	SMI Status Register 1	SMI_stat1	Configuration		Index 074h
15.2.17	Peripheral Inactivity Status Register 0	Perif_status0	Configuration		Index 075h
15.2.18	Activity Status Register 0	Activ_status0	Configuration		Index 077h
15.2.19	Activity Status Register 1	Activ_status1	Configuration		Index 078h
15.2.20	Activity Status Register 2	Activ_status2	Configuration		Index 079h
15.2.21	PMU State Register	PMU	Configuration		Index 07Ah
15.2.22	General Purpose Register	GP	Configuration		Index 07Bh
15.2.23	Clock Control Register 0	Clock_cont0	Configuration		Index 07Ch
15.2.24	Doze Timer Read Back Register	Doze	Configuration		Index 088h
15.2.25	Standby Timer Read Back Register	Standby	Configuration		Index 089h
15.2.26	Suspend Timer Read Back Register	Suspend	Configuration		Index 08Ah
15.2.27	House-Keeping Timer Read Back Register	HK_timer	Configuration		Index 08Bh
15.2.28	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: X is the value of the G_Base and can range from 8 to ...
Note 3: These registers are note described in this publication.

Table 6-1. CPU Registers located in the ST 486 Manual

Register Name	Mnemonic	Purpose	Address	Access type
Configuration Registers			22h	
Configuration Control 1	CCR1	IO	23h	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. SDRAM CONTROLLER

7.1 INTRODUCTION

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

The iDragon 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 SDRAM 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.

This main memory is supported using 4 Memory Banks. *Note that these memory banks are those viewed by the iDragon and not internal banks of the chips themselves.*

Each Memory Bank is accessed by a Chip Select signal (CS#[3:0]).

7.2 MEMORY CONTROLLER

The iDragon handles the memory data (DATA) bus directly, controlling from 2 to 128 MBytes. The SDRAM controller supports accesses to the Memory Banks to/from the CPU (via the host), from the VMI, to/from the CRTC, to the VIDEO & to/from the GE. (Banks 0 to 3) which can be populated with either single or double sided 72-bit (4 bit parity) DIMMs. Parity is not supported.

The SDRAM controller only supports 64 bit wide Memory Banks.

Four Memory Banks (if DIMMS are used; Single sided or two double-sided DIMMs) are supported in the following configurations (see Table 7-1):

Table 7-1. Supported Memory Configs

Memory Bank size	Number	Organisation	Device size
1Mx64	4	1Mx16	16Mbit
2Mx64	8	2Mx8	
4Mx64	16	4Mx4	

The SDRAM Controller supports buffered or un buffered SDRAM but not EDO or FPM modes. SDRAMs must support Full Page Mode Type access.

The iDragon Memory Controller provides various programmable SDRAM parameters to allow the SDRAM interface to be optimized for different processor bus speeds SDRAM speed grades and CAS Latency.

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 iDragon relies on subtractive decode before initiating an internal memory cycle. ISA master cycles in this range maps to main memory. The iDragon will negate IOCHRDY if necessary.

The DMA master cycles in this range maps to main memory. The iDragon 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 iDragon 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-cache able.

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 con trolled via Shadow Control register 3. This ad dress 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 con trolled via Shadow Control register 3. This ad dress 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 SDRAM 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 SDRAM memory can be different from the popu lated 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 ad dress range has the same characteristics as the 0 640K (compatible DOS memory) range.

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

With the exception of memory space allocated to the Extended Graphics (described later), all cycles above the addressable SDRAM 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 SDRAM 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 SDRAM 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 iDragon 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 iDragon 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 SDRAM 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 iDragon uses the physical memory behind the CPU address range A0000h - B0000h for the SMM memory. The SMM base address registe 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 mem ory can only be accessed when SMIACT# is ac tive. The cacheability of this segment is hardwired to 0.

7.3.16 ADDRESSABLE SDRAM MEMORY

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

TOPM = The size of total physical SDRAM is de fined by SDRAM 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 shad ow, or 0KB, if any of the 8 x 16KB-segments of D0000h-EFFFFh is enabled for shadow.

The addressable SDRAM memory =

TOPM - TOGM + MHOLE_SIZE + REMAP_SIZE

7.3.17 CPU ADDRESS TO SDRAM ADDRESS MAPPING

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

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

The lower range of the SDRAM, starting from the SDRAM address 00h, is allocated to frame buffer. The rest of the memory is used by the system. The CPU address is mapped to the SDRAM 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 iDragon configuration registers).

iDragon 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 SDRAM = 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 1128 MBytes above this is mapped 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 SDRAM space that no device accesses. The total CPU memory then becomes 4 MBytes plus 896 KBytes.

7.4 IO ADDRESS MAP .

Table 7-2. IO map space

IO address	Description	Notes
0000h-000Fh	8237 DMA controller 1 registers.	1
0020h-0021h	8259 Interrupt controller 1 registers.	
0022h	iDragon specific configuration registers index port	
0023h	iDragon 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 03D4h,03D5h,03DAh 03C0h-03CFh	VGA registers.	
0CF8h	PCI configuration Address register.	
0CFCh-0CFFh	PCI configuration Data register.	
46E8h	VGA add-in mode enable register.	2

The iDragon implements a number of regisers in IO address space.

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

Notes:

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

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

7.4.1 PCI CONFIGURATION ADDRESS MAP:

The iDragon occupies Device number 0 slot on the PCI bus and implements a number of registers in PCI configuration address space. These regis ters occupy the following map (seeTable 7-3) :

Table 7-3. PCI configuration address space

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

7.5 CACHE RELATED REGISTERS

7.5.1 CACHE ARCHITECTURE REGISTER 0 C.I. 20H (CASH_ARC0)

This register controls various attributes of the L2 cache.

Bit 7 **CPU pipeline access support**

Bit 7	CPU pipelined access
0	not supported
1	supported

Bit 6 **Burst addressing order**

Bit 6	Burst order
0	Intel
1	linear

Bit 5 **L1 write back indication**

Bit 5	L1 write back
0	Not supported
1	Supported

Bits 4-3 **SRAM type.** These bits control the type of SRAMs used to construct L2 cache.

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

Bit 2 **Number of L2 banks.**

Bit 2	L2 Banks
0	One bank
1	Two banks

When programmed to 2 banks, L2 interleaving is enabled.

Bit 1 **L2 write back control.**

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

Bit 0 **L2 cache enable.**

Bit 1	L2 cache
0	disabled
1	enabled

This register defaults to 00h at reset.

7.5.2 CACHE ARCHITECTURE REGISTER 1 C.I. 21H (CASH_ARC1)

This register controls various attributes of L2 cache.

Bits 7-5 **L2 cache size.**.

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

Bit 4 **IO NA# Enable..**

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

Bits 3-2 **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 al ways work.

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

Bit 1 **Read around write enable.**

Bit 4	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

Bit 0 *Reserved.*

This register defaults to 00h at reset.

7.5.3 CACHE ARCHITECTURE REGISTER 2 C.I. 22H (CASH_ARC2)

Bit 7 *Reserved.*

Bit 6 **Slow host data driver.**

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

Bit 5 **Cache write enable pulse width.**

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

Applicable to asynchronous SRAMs only. Must be '0' for synchronous SRAMs.

Bit 4 **Cache data hold after write enable.**

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

Must be a '1' if 1.5 clocks wide write enable pulse width is selected via bit 5 above.

Bits 3-2 **Burst access wait states.**

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

Bits 1-0 **Tag access wait states.**

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

This register defaults to FFh at reset.

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 - C.I. 24H (MEM_HOLE)

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.

Bit 7 **Memory Hole Enable.** This bit controls the enable of memory hole function.

0 = disabled

1 = enabled

Bits 6-4 **Memory Hole Size.** These bits control the size of memory hole.

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

Bits 3-0 **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.

This register defaults to 00h at reset.

Programming notes;

This memory hole is also non-cacheable.

7.6.2 SHADOW CONTROL REGISTER 0 - C.I. 25H (SHADOW_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 fo read and one for write.

Bit 7 **Read Control CC000h-CFFFFh.** This bit controls the read attribute of the CC000h-CFFFFh memory.

0 = shadow disabled for read cycle
1 = shadow enabled for read cycle

Bit 6 **Write Control CC000h-CFFFFh.** This bit controls the write attribute of the CC000h-CFFFFh memory.

0 = shadow disabled for write cycle
1 = shadow enabled for write cycle

Bit 5 **Read Control C8000h-CBFFFh.** This bit controls the read attribute of the C8000h-CBFFFh memory.

0 = shadow disabled for read cycle
1 = shadow enabled for read cycle

Bit 4 **Write Control C8000h-CBFFFFh.** This bit controls the write attribute of the C8000h-CBFFFh memory.

0 = shadow disabled for write cycle
1 = shadow enabled for write cycle

Bit 3 **Read Control C4000h-C7FFFh.** This bit controls the read attribute of the C4000h-C7FFFh memory.

0 = shadow disabled for read cycle
1 = shadow enabled for read cycle

Bit 2 **Write Control C4000h-C7FFFFh.** This bit controls the write attribute of the C4000h-C7FFFh memory.

0 = shadow disabled for write cycle
1 = shadow enabled for write cycle

Bit 1 **Read Control C0000h-C3FFFh.** This bit controls the read attribute of the C0000h-C3FFFh memory.

0 = shadow disabled for read cycle
1 = shadow enabled for read cycle

Bit 0 **Write Control C0000h-C3FFFFh.** This bit controls the write attribute of the C0000h-C3FFFFh memory.

0 = shadow disabled for write cycle
1 = shadow enabled for write cycle

This register defaults to 00h at reset.

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 - C.I. 26H (SHADOW_1

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

Bit 7 **Shadow Read Control DC000h-DFFFFh.** This bit controls the read attribute of the DC000h DFFFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 6 **Shadow Write Control DC000h-DFFFFh.** This bit controls the write attribute of the DC000h DFFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

Bit 5 **Shadow Write Control D8000h-DBFFFh.** This bit controls the read attribute of the D8000h DBFFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 4 **Shadow Write Control D8000h-DBFFFh.** This bit controls the write attribute of the D8000h DBFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

Bit 3 **Shadow Read Control D4000h-D7FFFh.** This bit controls the read attribute of the D4000h D7FFFh memory

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 2 **Shadow Write Control D4000h-D7FFFh.** This bit controls the write attribute of the D4000h D7FFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

Bit 1 **Shadow Read Control D0000h-D3FFFh.** This bit controls the read attribute of the D0000h D3FFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 0 **Shadow Write Control D0000h-DFFFFh.** This bit controls the write attribute of the D0000h-DFFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

This register defaults to 00h at reset.

Programming Notes

This entire 64K segment has the cacheability bit hardwired to '0' (disabled)

7.6.4 SHADOW CONTROL REGISTER 2 - C.I. 27H (SHADOW_2)

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

Bit 7 **Read Control EC000h-EFFFFh.** This bit controls the read attribute of the EC000h-EFFFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 6 **Write Control EC000h-EFFFFh.** This bit controls the write attribute of the EC000h-EFFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

Bit 5 **Read Control E8000h-EBFFFh.** This bit controls the read attribute of the E8000h-EBFFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 4 **Write Control E8000h-EBFFFh.** This bit controls the write attribute of the E8000h-EBFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

Bit 3 **Read Control E4000h-E7FFFh.** This bit controls the read attribute of the E4000h-E7FFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 2 **Write Control E4000h-E7FFFh.** This bit controls the write attribute of the E4000h-E7FFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

Bit 1 **Read Control E0000h-E3FFFh.** This bit controls the read attribute of the E0000h-E3FFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 0 **Write Control E0000h-EFFFFh.** This bit controls the write attribute of the E0000h-EFFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

This register defaults to 00h at reset

Programming Notes

This entire 64K segment has the cacheability bit hardwired to '0' (disabled)

7.6.5 SHADOW CONTROL REGISTER 3 - C.I. 28H (SHADOW_3)

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

Bit 7 **SDRAM Initialization Enable.** This bit controls whether CPU accesses in A0000h-BFFFFh address range are decoded as VGA frame buffer access or SMRAM access.

0 = A0000h-BFFFFh is interpreted as VGA frame buffer access

1= A0000h-BFFFFh is interpreted as SMRAM access.

The iDragon 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.

Bit 6 **Cache Control F0000h-FFFFFh.** This bit controls the cacheability of F0000h-FFFFFh block when the shadow function is enabled.

0 = cacheability disabled

1 = cacheability enabled

Bit 5 **Cache Control C0000h-C7FFFh.** This bit controls the cacheability of C0000h-C7FFFh block when the shadow function is enabled.

0 = cacheability disabled

1 = cacheability enabled

Bits 4-2 *Reserved*

Bit 1 **Read Control F0000h-FFFFFh.** This bit controls the read attribute of F0000h-FFFFFh memory.

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

Bit 0 **Write Control F0000h-FFFFFh.** This bit controls the write attribute of F0000h-FFFFFh memory.

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

This register defaults to 00h at reset.

Programming notes

T*he 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 - C.I. 29H (VGA_DEC)

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

Bits 7-6 *Reserved*

Bit 5 stop g-clock (Graphics clock).

Bit 4 stop p-clock (PCI Clock).

Bit 3 stop d-clock (Dot clock).

Bit 2 **Palette Snoop Enable:**

1 =Palette write cycles are propagated to PCI bus in addition to updating the internal palette.

0 = Palette write cycles are terminated internally and are not propagated to PCI.

Bit 1 **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 **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.

This register defaults to 03h after reset.

7.7 HOST SDRAM CONTROLLER REGISTERS

The iDragon manages 4 Memory Banks (if DIMM sockets are used they can be populated with either single or double sided 64-bit data DIMMs). For SDRAM densities are supported see Section 7.2 .

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 SDRAM BANK 0 REGISTER - C.I. 30H (SDRAM__Bank0)

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

Bank 0 Top Address = Memory Bank0 size in MBytes -1.

Bank 1 Top Address = Memory Bank0 + Memory Bank1 size in MBytes -1

This register defaults to 07h.

Example 1:

Memory Bank0 = 4MB
Memory Bank1 = 4MB

Bank 0 Top Address = 4 -1= 3= 03h

Bank 1 Top Address = 3 + 4 - 1 = 07h

Bank 2, 3 Top Address = 07h

Example 2: for use with double sided DIMMs

Memory Bank0 = 32MBytes (dbl. sided DIMMS)
Memory Bank1 = 32MBytes (dbl. sided DIMMS)

Bank 0 Top Address = 16 - 1 = 15 = 0Fh

Bank 1 Top Address = 16 + 16 - 1 = 31 = 1Fh

Bank 2 Top Address = 32 + 16 - 1 = 47 = 2Fh

Bank 3 Top address = 48 + 16 - 1 = 63 = 3Fh

7.7.2 SDRAM BANK 1 REGISTER - C.I. 31H (SDRAM_Bank1)

This register controls the top address of SDRAM bank 1.

7.7.3 SDRAM BANK 2 REGISTER - C.I. 32H (SDRAM_Bank2)

This register controls the top address of SDRAM bank 2.

7.7.4 SDRAM BANK 3 REGISTER - C.I. 33H (SDRAM_Bank3)

This register controls the top address of SDRAM bank 3.

7.7.5 MEMORY BANK WIDTH - C.I. 34H (MEM_WIDTH)

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

Bit 7-4 *Unused*

Bit 3 **Memory Bank 3 Width Code**

Bit 2 **Memory Bank 2 Width Code**

Bit 1 **Memory Bank 1 Width Code**

Bit 0 **Memory Bank 0 Width Code**

Code	Memory B ank W idth
0	64 bits (default)
1	32 bits

This register defaults to 00h at reset.

7.7.6 GRAPHICS MEMORY SIZE REGISTER - C.I. 36H (GRAPH_MEM)

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

Bit 7 **Graphics RAS Active.** This bit controls if RAS is kept active after the current framebuffe SDRAM access.

0 = keep RAS# active
1 = deassert RAS#

Bit 6 *Reserved*.

Bits 5-0 **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 neces sary.

This register defaults to 04h.

7.7.7 Presents Detect Register - C.I. 97h

This register is read through the DDC register.

7.8 SDRAM CONTROLER REGISTER ACCESS

These registers are used to configure the SDRAM controller.

7.8.1 REGISTER 0 (MEM_REG0) GBASE + 4C6000h

This is a 31 bit configuration register for the SDRAM controller block:

Bit 30-28 **Read CAS Latency** should be equal to CAS Latency (Left programmable for debug purpose).

Bit 27 **Registered Memory Bank**, Indicate if we use registered Memory Banks '1' or not '0'.

Bit 26 **Mode Register Set Request,** If set to1, the SDRAM chips are updated with the corresponding value programmed in bits [16:0].

Bit 25 **Keep RAS Active**, If set to 1, RAS# is kept active after the current SDRAM access.

Bit 24 **Latch_Host_Data_In** for host input data. If set to 0, it selects MD[63:0] directly from SDRAM, if set to 1, it selects latched MD[63:0] by RDCLK (see bits 3-0 in Register 1).

Bit 23 **Latch_GE_Data_In** for GE input data. If set to 0 it selects MD[63:0] directly from SDRAM , if set to 1 it selects latched MD[63:0] by RDCLK (see bits 3-0 in Register 1).

Bit 22 **Latch_CRTC_Data_In** for CRTC input data. If set to 0, it select MD[63:0] directly from SDRAM, if set to 1 it selects latched MD[63:0] by RDCLK (see bits 3-0 in Register 1).

Bit 21-19 **CAS Latency** (Called CL in datasheet s). Number of controller cycles.

Bit 18-17 **CONFIGURATION** of a Memory Bank :

Bit 18	Bit 17	Memory Bank Configuration
0	0	[4Mx4]x16
0	1	[2Mx8]x8
1	0	[1Mx16]x4
1	1	*Reserved*

These bits are used to determine the maximum burst length (full page burst).

Maximum burst length :

[4Mx4]x16: 1024,

[2Mx8]x8 : 512,

[1Mx16]x4: 256.

If the Memory Banks are populated with a different kind of memory, the 2 bits are programmed to maximise the burst length using the minimum amount of the memory.

Bit 16-15 **RAS Precharge** (called tRP in datasheets)**.** Number of controller cycles .

Bit 14-11 *Reserved.* Should be set to '0000'.

Bit 10-8 **RAS to CAS Delay** (Called tRCD in datasheets). Number of controller cycles.

Bit 7 **Burst Type**.Should be set to '0'.

Bit 6-4 **Burst Length**. Should be set to '111'

Bit 3-0 **RAS Cycle Time** (Called tRC in datasheets). Number of controller cycles.

Reset values. :31x32198376h

7.8.2 REGISTER 1 (MEM_REG1) GBASE + 4C6004h

This 5 bits register is used for the read clock scheme. See Chapter 6.3 "Clock considerations for more details. This delay can be set up to 3.5 ns beyond th 15ns required from the previous MCLKI edge.

Bit 4 **MEM16_OE_**, This bit is set to '1' to get 16mA output enabled, set to '0' to get 8mA output enabled

Bit 3-0 **Read Clock Delay Programming,** These should be set to 00h.

7.9 MEMORY CLOCK REGISTERS

The MCLK is used to the Memory operations

7.9.1 MCLK CONTROL REGISTER 0 MCLK00 INDEX 40

Bit 7 This is unused.

Bits 6-3 This the 4-bit M (divisor) value of the Memory synthesizer.

Bits 2-0 These are bits 7-5 of the 8 bit N (mulitipli er) of the Memory clock synthesizer.

This register defaults to 0x5B at reset. This value when combined with the default value of the othe half of this pair results in a memory clock of 80.05

MHz assuming 14.318 MHz oscillator clock as the reference input.

For MCLK frequency programming values see Table 7-4

7.9.2 MCLK CONTROL REGISTER 1 MCLK01 INDEX 41

Bit 7 This is the bit 0 of the 3-bit P (exponent) value of the Memory clock synthesizer.

Bits 6-2 These are bits 4-0 of the 8-bit N (multiplier) value of the Memory clock synthesizer.

Bits 1-0 These are bits 2-1 of the 3-bit P (exponent) value of the Memory clock synthesizer.

This register defaults to 0xEC at reset. This value when combined with the default value of the othe half of this pair results in a memory clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

For MCLK frequency programming values see Table 7-4

Table 7-4. MCLK Control Register Address 22 Index 40h, 41h

MHz	Reg1,Index 41h	Reg0, Index 40h	Actual Freq.	m	n	p
100	E8h	5Ch	100.227260	Bh	9Ah	1
95	C8h	5Ch	95.020649	Bh	92h	1
90	C0h	75h	89.999989	Eh	B0h	1
85	FCh	42h	85.014194	8h	5Fh	1
80	ECh	5Bh	80.051643	Bh	7Bh	1
75	A0h	6Ch	74.895095	Dh	88h	1
66	84h	74h	65.965901	Eh	81h	1
60	69h	6Eh	60.026216	13h	218h	2h
55	25h	5Dh	54.994827	11h	169h	2h
50	69h	5Ch	50.113630	Bh	9Ah	2
45	41h	75h	44.999994	Eh	B0h	2
8	22h	42h	8.053978	8h	75h	4h

7.10 SDRAM INTERFACE

The iDragon provides MA, RAS#, CAS#, WE#, CS#, DQM#, BA0 (MA[11])and MD for SDRAM control. From 2 to 128 MBytes of main memory are supported in 1 to 4 banks. All Banks must be 64 bits wide.

The following memory devices are supported:

4Mbit x 4, 8Mbit x 2 & 16Mbit x 1 or if in the case of two internal bank chips, 2Mbit x 4 x 2, 4Mbit x 2 x 2 & 8Mbit x 1 x 2 .

The following Figure 7-1 and Figure 7-2, shows two possible SDRAM organizations based on one or two bank configurations.

Notes for Figure 7-1 and Figure 7-2;

All buffers must be low skew clock buffers

One clock driver can operate upto four memory chips.

All the clock lines must follow the rules below;

MCLKI = MCLK0 + MCLK0A

= .....

= MCLK0 + MCLK0

= MCLK1 + MCLK1A

= .....

= MCLK1 + MCLK1

This means that all line lengths must from the buff er to the memory chips (MCLK1 or MCLK0 or ... and from the buffer to the iDragon (MCLKI) must be identical.

Figure 7-1. One memory Bank with eight chips (8-Bit)

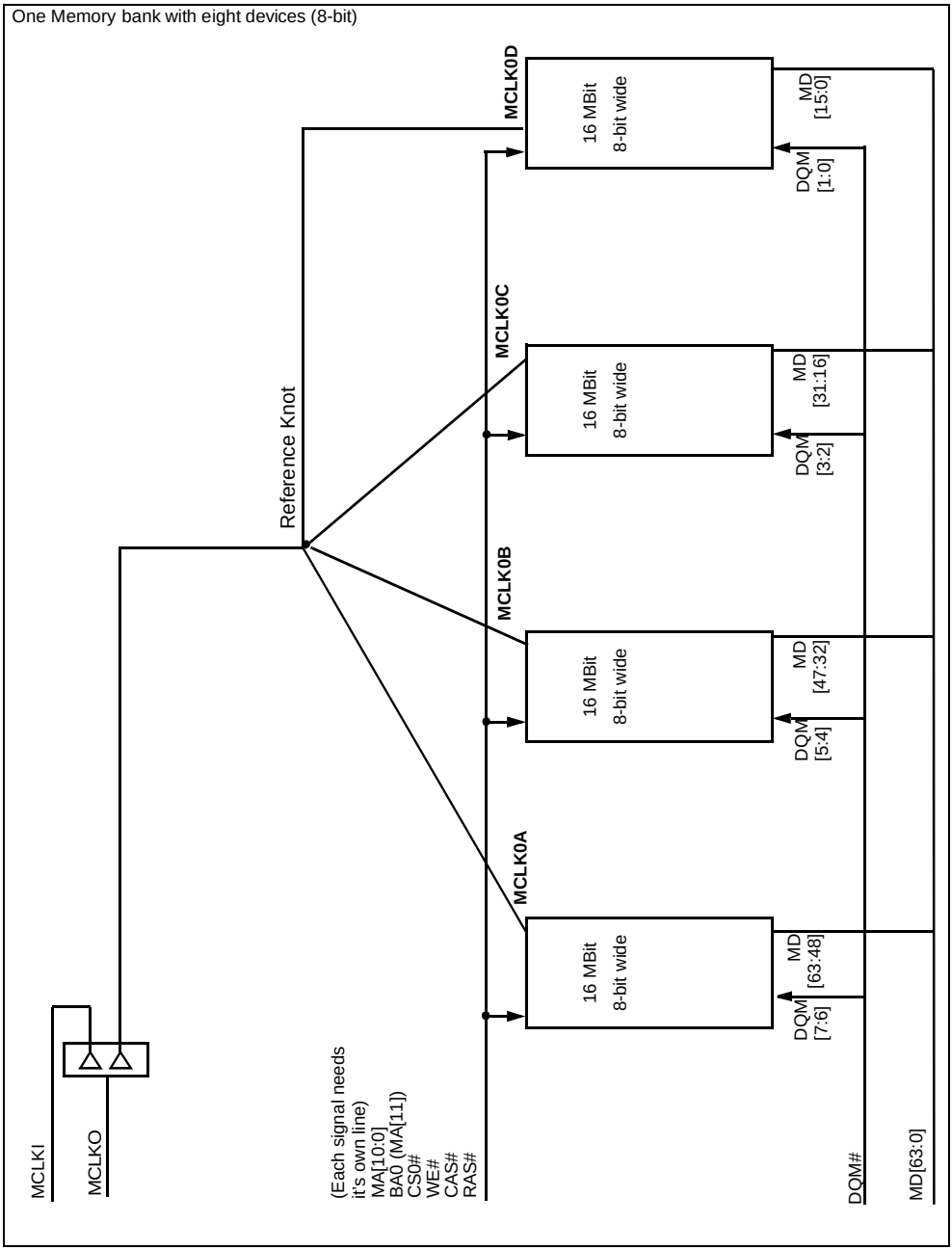
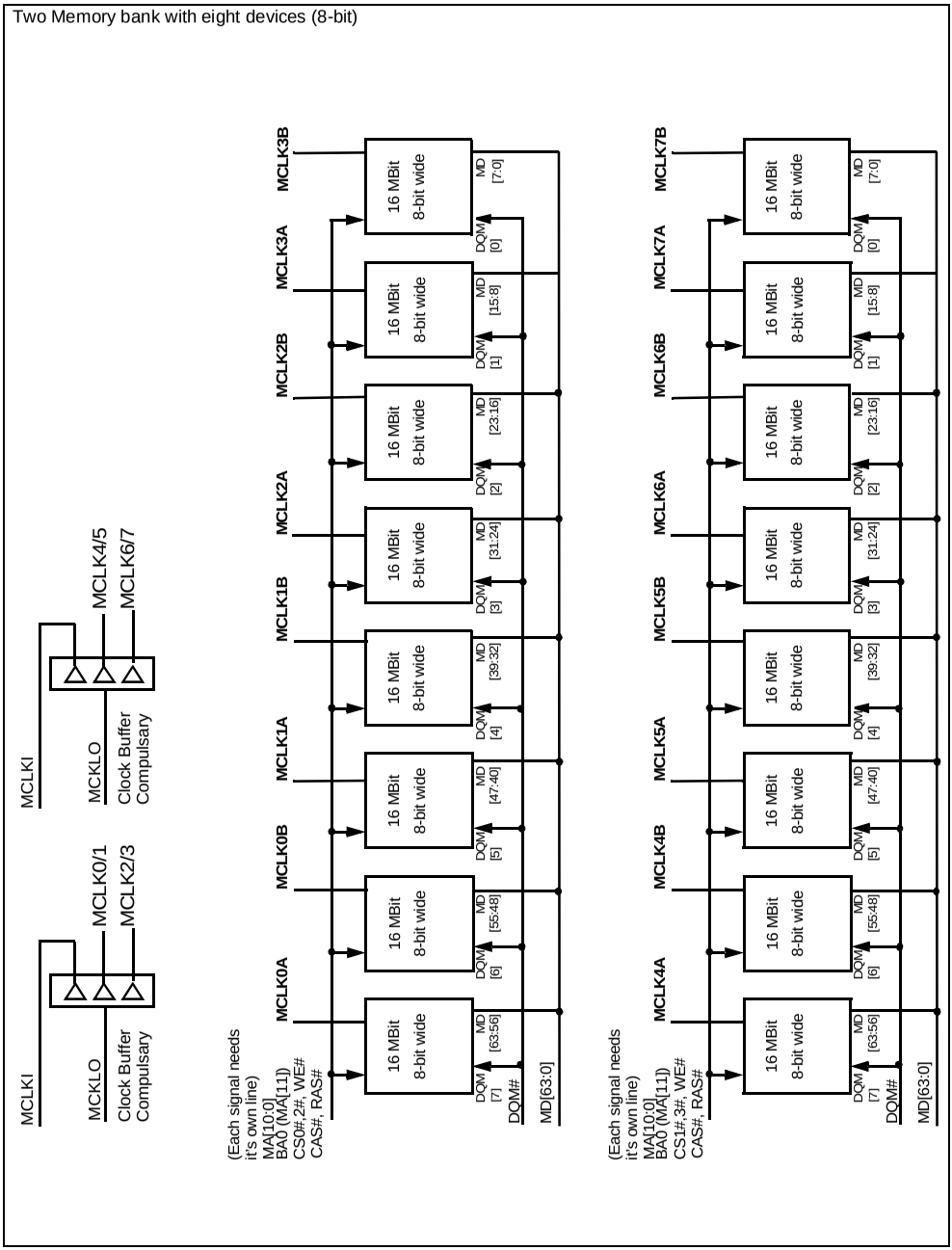


Figure 7-2. Two memory banks with eight chips on each (8-Bit)



7.10.1. 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 SDRAM.

The bank attributes can be retrieved from a lookup table to select the final SDRAM row and column address mappings. (Table 7-6). Also Table 7-5 shows the Standard DIMM Pinout for the users that wish to design with DIMMs.

Table 7-5. Standard Memory DIMM Pinout

Memory Banks pin number	16Mbit(2 banks)
...	MA[10:0]
123	-
126	-
39	-
122	BA0(MA11)

Table 7-6. Address Mapping

Address Mapping: 16 Mbit - 2 banks												
iDragon I/F	BA0(MA11)	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
RAS ADDRESS	A11	A22	A21	A2	A19	A18	A17	A16	A15	A14	A13	A12
CAS ADDRESS	A11	0	A24	A23	A10	A9	A8	A7	A6	A5	A4	A3

7.11 SDRAM ARBITRATION:

Following agents compete for the system SDRAM 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 SDRAM 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 SDRAM. The SDRAM arbiter employs a dynamic arbitration algorithm to optimize the SDRAM 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 CRT and System master request. A System master re quest will terminate an ongoing Graphics service at the nearest CAS boundary while a CRTC re quest can terminate a on-going graphics service at the end of a sequence of read/write.

- -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 be havior changes significantly. The goal of the arbi ter here is to ensure that the CRTC and Video FIFO occupancy is above a programmable mini mum number of Bytes. This is necessary because, some memory and screen configurations do not have sufficient bandwidth availability. Since the drain rate is equal to the peak available band width, it can not be sustained if all the pixels are to be fetched on demand. To overcome this, the ar biter 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 res ervoir. 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 dis tance before the video window x position, the arbi ter switches over to a different set of low wate 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 be tween CRTC and Video requests, the arbiter em ploys a programmable burst length to arbitrate be tween 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 windo repaints starts, the low water marks decrease lin early over the size of the window, to reflect the de creasing 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.

8. PCI CONTROLLERS

8.1 INTRODUCTION

The PCI bus is the main data communication link to the iDragon chip.

Two PCI devices are present internally in the iDragon, a “North Bridge” and a “South Bridge”.

The iDragon 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 iDragon

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 numbe 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.

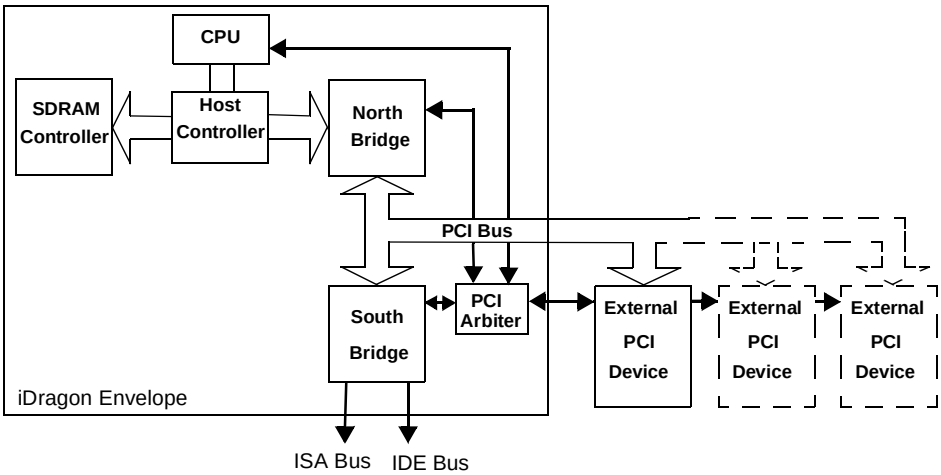
The South Bridge, as a PCI bus agent (expansion bridge class) fully complies with PCI specification 2.1. The South bridge implements two PCI functions:

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 configura tion 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 thePCI 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.

Table 8-1. Register CF8h

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 8-2. Register CFCh

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

8.3 CONFIGURATION ADDRESS REGISTER IO CF8H (CONFIG_ADDRESS)

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

Bit 31 **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 *Reserved.* Must be written to '0'. Read back as '0'.

Bits 23-16 **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 **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 IDSEL line. The North Bridge is assigned the Device Number 0xBh, which corre sponds to IDSEL on AD11 signal. The South Bridge is assigned the Device Number 0xCh, which corresponds to IDSEL on AD12 signal.

Bits 10-8 **Function Number.** During a PCI config uration 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 **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 *Reserved.* Must be written to a '0'. Reads back as '0'.

This register defaults to 00h on reset.

8.4 CONFIGURATION DATA REGISTER IO CFCH (CONFIG_DATA)

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

8.5 THE NORTH BRIDGE CONFIGURATION REGISTERS

Table 8-3. North Bridge Reset Values

31 ---------- 16		15 ---------- 0		
Device ID : 020Ah		Vendor ID : 104Ah		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

The iDragon 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.

8.5.1 North Bridge Vendor Identification register PCI Config. Index 0h-1h (NB_Vend_ID)

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

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

Writes to this register have no effect.

8.5.2 North Bridge Device Identification register PCI Config. Index 2h-3h (NB_Device_ID)

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

Bits 15-0 These bits are hardwired to 0564h

Writes to this register have no effect.

8.5.3 North Bridge PCI Command register PCI Config. Index 4h-5h (NB_Command)

This is the 16-bit PCI command register.

Bits 15-9 *Reserved.* These bits are hardwired to '0'. Writes have no effect on them.

Bit 8 **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 B ridge 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 detect ed. If this bit is set to '0', the North Bridge will not assert SERR#.

Bit 7 **Address/Data stepping enable.** This bit is hardwired to a '0'. Writes to it have no effect.

Bit 6 **PERR# response.** Must always be set to '0'.

Bit 5 **VGA Palette Snoop enable.** This bit 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.

Bit 3 **Enable Special cycles.** This bit is hardwired to a '0'. Writes to it have no effect.

Bit 2 **Bus Master enabled.** This bit is hardwired to a '1'. Writes to it have no effect.

Bit 1 **Memory Enable.** This bit is hardwired to a '1'. Writes to it have no effect.

Bit 0 **IO Enable.** This bit is hardwired to a '1'. Writes to it have no effect.

This register defaults to 0007h after reset.

8.5.4 North Bridge PCI Status register PCI Config. Index 6h-7h (NB_Status)

This is the 16-bit PCI Status register.

Bit 15 **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 **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 **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 **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 **Signaled Target Abort.** This bit is hardwired to '0'.

Bit 10-9 **DEVSEL Timing.** These bits are hardwired for medium timing to '01'. Writes have no effect.

Bit 8 **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 **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 *Reserved.* These bits are hardwired to '0's.

This register defaults to 0280h after reset.

8.5.5 North Bridge PCI Revision ID register PCI Config. Index 8h (NB_Rev_ID)

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

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

8.5.6 North Bridge Device Class Code register PCI Config. Index 9h-Bh (NB_Class_Code)

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

Bits 31-24 (Bh) **Base Class Code.** These bits are hardwired to 00h

Bits 23-16 (Ah) **Sub Class Code.** These bits are hardwired to 00h

Bits 15-8 (09h) **Programming Interface Register.** These bits are hardwired to 00h.

8.5.7 North Bridge Header Type register PCI Config. Index Eh (NB_head)

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

8.5.8 North Bridge PCI Control register PCI Config. Index 50h (NB_Control)

Bit 31-23 *Reserved.* Hardwired to '0'.

Bit 22 **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 **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 **PCI to Host Write Posting Enable.** If this bit is set to '1', all burst writes from a PCI maste 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.

Bit 19-5 *Reserved.* Hardwired to '0'.

Bit 4 **PERR_ on read data parity error enable.**

Bit 3 **PERR_ on write data parity error enable.**

Bit 2 **PERR_ on address parity error enable.**

Bit 1 **SERR_ on PERR_ enable.**

Bit 0 **SERR_ on received target abort.**

8.5.9 North Bridge PCI Error Status register PCI Config. Index 54h (NB_Error_Status)

Bit 31-5 *Reserved.* Hardwired to '0'.

Bit 4 **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 **Write Data Parity Error Status.** This bit is set when a PCI write data parity error is detected. Writing a '1' will clear it.

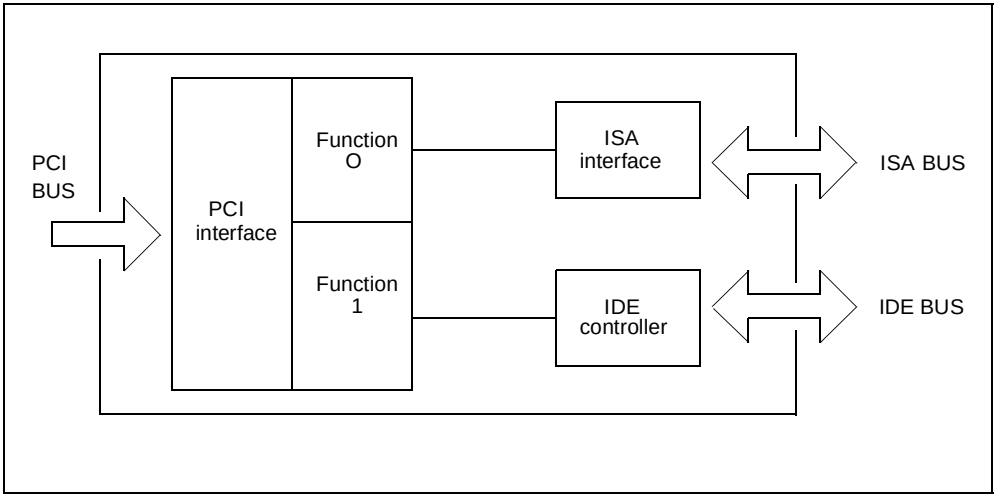
Bit 2 **Address Parity Error Status.** This bit is set when a PCI address parity error is detected. Writing a '1' will clear it.

Bit 1 **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 **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 iDragon South Bridge configuration registers are accessed using the values below :

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

Bus = 0

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

Function = 0(ISA bridg e)

- Responds to IO / memory / config
- Translates Master ISA to PCI
- Translates PCI to Slave ISA

Function = 1 (IDE c ontroller)

- Responds to IO / config

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

8.7 SOUTH BRIDGE PCI FUNCTION 0 CONFIGURATION REGISTERS

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

31		16 15	0	
Device ID: 021Ah		Vendor ID: 104Ah		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

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

8.7.1 South Bridge Vendor Identification register PCI Config. F#0 Index 0h-1h (SB_Vend_ID0)

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

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

Writes to this register have no effect.

8.7.2 South Bridge Device Identification register PCI Config. F#0 Index 02h-03h (SB_Device_ID0)

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

Bits 15-0 These bits are hardwired to 55CCh

Writes to this register have no effect.

8.7.3 South Bridge PCI Command register PCI Config. F#0 Index 4h-5h (SB_Command0

This is the 16-bit PCI command register.

Bits 15-9 *Reserved.* These bits are hardwired to '0's. Writes have no effect on them.

Bit 8 **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 **Address/Data stepping enable.** This bit is hardwired to a '0'. Writes to it have no effect.

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 no effect.

Bit 3 **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 **Bus Master enabled.** This bit is hardwired to a '1'. Writes to it have no effect.

Bit 1 **Mem Enable.** This bit is hardwired to a '1'. Writes to it have no effect.

Bit 0 **IO Enable.** This bit is hardwired to a '1'. Writes to it have no effect.

This register defaults to 000Fh after reset.

8.7.4 South Bridge PCI Status register PCI Config. F#0 Index 06h-07h (SB_Status0)

This is the 16-bit PCI Status register.

Bit 15 *Reserved.* This bit is hardwired to '0'.

Bit 14 **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 registe is set to a '1' to enable SERR# signaling.

Bit 13 **Signaled Master Abort.** This bit is hardwired to a '0'.

Bit 12 **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 **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.

Bit 10-9 **DEVSEL Timing.** These bits are hardwired for medium timing to '01'. Writes have no effect.

Bit 8 **Data Parity Error Detected.** This bit is hardwired to '0'.

Bit 7 **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 *Reserved.* These bits are hardwired to '0'.

This register defaults to 0280h after reset.

8.7.5 South Bridge PCI Revision ID register PCI Config. F#0 Index 08h (SB_Rev_ID0)

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

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

8.7.6 South Bridge Device Class Code register PCI Config. F#0 Index 09h-0Bh (SB_Class_Code0)

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

Bits 31-24 (0Bh) **Base Class Code.** These bits are hardwired to 06h (Bridge Device).

Bits 23-16 (0Ah) **Sub Class Code.** These bits are hardwired to 01h (ISA Bridge).

Bits 15-8 (09h) **Programming Interface Register.** These bits are hardwired to 00h.

8.7.7 South Bridge Header Type PCI Config. F#0 Index 0Eh (SB_Header0)

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

8.7.8 South Bridge Miscellaneous register PCI Config. F#0 Index 40h (SB_Misc0)

Bit 7-1 *Reserved.* Hardwired to 00h.

Bit 0 **PCI 2.0 Enable.** If this bit is set to '1', South Bridge will be compatible with PCI 2.0 standard. If

8.8 SOUTH BRIDGE PCI FUNCTION 1 CONFIGURATION REGISTERS

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

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

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

8.8.1 South Bridge Vendor Identification register PCI Config. F#1 Index 00h-01h (SB_Vend_ID1)

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

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

Writes to this register have no effect.

8.8.2 South Bridge Device Identification register PCI Config. F#1 Index 02h-03h (SB_Device_ID1)

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

Bits 15-0 These bits are hardwired to 55CCh

Writes to this register have no effect.

8.8.3 South Bridge PCI Command register PCI Config. F#1 Index 04h-05h (SB_Command1)

This is the 16-bit PCI command register.

Bits 15-9 *Reserved.* These bits are hardwired to '0's. Writes have no effect on them.

Bit 8 **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 **Address/Data stepping enable.** This bit is hardwired to a '0'. Writes to it have no effect.

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.

Bit 3 **Enable Special cycles.** This bit is hardwired to a '0'. Writes to it have no effect.

Bit 2 Reserved.

Bit 1 **Mem Enable.** This bit is hardwired to a '0'. Writes to it have no effect.

Bit 0 **IO Enable.** Setting this bit to a '1' enables access to the IDE IO registers.

This register defaults to 0000h after reset.

8.8.4 South Bridge PCI Status Register PCI Config. F#1 Index 06h-07h (SB_Status1)

Bit 15 *Reserved.* This bit is hardwired to '0'.

Bit 14 **Signaled SERR#.** This bit is set to a '1'.

Bit 13 **Signaled Master Abort.** This bit is set to a '1' .

Bit 12 **Received Target Abort.** This bit is set to a '1'.

Bit 11 *Reserved.* This bit is hardwired to '0'.

Bit 10-9 **DEVSEL Timing.** These bits are hardwired for medium timing to '01'. Writes to these bits have no effect.

Bit 8 **Data Parity Error Detected.** This bit is hardwired to '0'.

Bit 7 **Fast Back-to-Back Capable.** This bit is hardwired to '1'.

Bits 6-0 *Reserved.* These bits are hardwired to '0'.

This register defaults to 0280h after reset.

8.8.5 South Bridge Revision ID Register PCI Config. F#1 Index 08h (SB_Rev_ID1)

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

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

8.8.6 South Bridge Programming Interface Register PCI Config. F#1 Index 09h (SB_Prog_Interf)

Bit 7 *Reserved.* This bit is hardwired to '1'. Writes to have no effect on this bit.

Bits 6-4 *Reserved.* These bits are hardwired to '0'.

Bit 3 This bit is hardwired to '1' indicating that the secondary channel is programmable to be eithe in legacy or native mode.

Bit 2 This bit selects the operating mode of the secondary channel.

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.

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 pri mary channel.

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'.

This register defaults to 8Ah after reset.

8.8.7 South Bridge Sub-class code register PCI Config. F#1 Index 0Ah (SB_Sub_Class)

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

8.8.8 South Bridge Base-class code register PCI Config. F#1 Index 0Bh (SB_Base_Class)

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

8.8.9 South Bridge Latency timer control register PCI Config. F#1 Index 0Dh (SB_Latency_Timer)

Bits 7-0 *Reserved.* These bits are hardwired to '0'.

This register defaults to 00h after reset.

8.8.10 South Bridge Header Type PCI Config. F#1 Index 0Eh (SB_Header_Timer

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

8.8.11 South Bridge IDE Base Address 0 register PCI Config. F#1 Index 10h-13h (SB_Base0)

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.

Bits 31-3 **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 *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.

8.8.12 South Bridge IDE Base Address 1 register PCI Config. F#1 Index 14h-17h (SB_Base1)

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.

Bits 31-2 **Base Address.** This field specifies the 4-Byte IO address range where the primary channel command registers are located.

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.

8.8.13 South Bridge IDE Base Address 2 register PCI Config. F#1 Index 18h-1Bh (SB_Base2)

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 leg acy mode operation, the secondary channel's command registers are decoded at 170h IO ad dress.

Bits 31-3 **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 ad dress occupies 4-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.

8.8.14 South Bridge IDE Base Address 3 register PCI Config. F#1 Index 1Ch-1Fh (SB_Base3)

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

Bits 31-2 **Base Address.** This field specifies the 4-Byte IO address range where the secondary channel command registers are located.

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.

8.8.15 South Bridge IDE Base Address 4 register PCI Config. F#1 Index 20h-23h (SB_Base4)

Bits 31-0 *Reserved***.**

8.8.16 South Bridge Primary IDE Timing register PCI Config. F#1 Index 40h-43h (SB_Prime_IDE_Time)

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.

Primary Slave Register (Bits 31-16) **:**

Bits 31-30 *Reserved.*

Bits 29-28 *Reserved.*

Bits 27-26 *Reserved.*

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.

The transfers to non-data registers of the IDE device will always have a recovery time of 10 clocks.

Refer to the Table 8-6 to determine the number of clocks for recovery times of the read/write signals:

Table 8-6. Recovery R/W Signal Time

Bits 25-23	PCI Clocks	
000	1	
001	2	
010	3	
011	4	
100	8	
101	9	
110	10	Default
111	12	

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 an active time of 10 clocks.

Refer to the Table 8-7 to determine the number of clocks for active times of the read/write signals:

Table 8-7. Active R/W Signal Time

Bits 22-20	PCI Clocks	
000	1	
001	2	
010	3	
011	4	
100	8	
101	9	
110	10	Default
111	12	

Note that the address setup time is implied. Afte initial startup latency, the address setup time is as follows in Table 8-8:

Table 8-8. Address Setup Time

22-21	Address setup time
00	1 clock
01	2 clocks
1X	3 clocks

Bit 19 **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 **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 **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 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching fo ATAPI commands (when A0h is written into the command register - Register Index 07h for the de vice). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are support ed by ATAPI devices.

Prefetch for the IDE controller is given in Table 8 9:

Primary Master Control Register (Bits 15-0) **:**

Bits 15-14 *Reserved.*

Table 8-9. Prefetch encoding

17-16	Prefetch
0X	Completely disabled
10	Enabled for non-ATAPI commands only.
11	Enabled for all commands

Bits 13-12 *Reserved.*

Bits 11-10 *Reserved.*

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 Table 8-10 below to determine the number of clocks for recovery times of the read/write signals:

Table 8-10. PIO R/W Signal Recovery Time

Bits 9-7	PCI Clocks	
000	1	
001	2	
010	3	
011	4	
100	8	
101	9	
110	10	Default
111	12	

Bits 6-4 **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 8-11 below to determine the number of clocks for active times of the read/write signals:

Table 8-11. PIO R/W Signal Active Time

Bits 6-4	PCI Clocks
000	1
001	2
010	3
011	4
100	8
101	9

Table 8-11. PIO R/W Signal Active Time

Bits 6-4	PCI Clocks	
110	10	Default
111	12	

Note that the address setup time is implied. Afte initial startup latency, the address setup time is given in Table 8-12:

Table 8-12. Address Setup Time Encoding

6-5	Address setup time
00	1 clock
01	2 clocks
1X	3 clocks

Bit 3 **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 **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 **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 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching fo ATAPI commands (when A0h is written into the command register - Register Index 07h for the de vice). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are support ed by ATAPI devices.

Prefetch for the IDE controller is given in Table 8 13:

Table 8-13. Prefetch IDE Controller Encoding

1-0	Prefetch
0X	Completely disabled
10	Enabled for non-ATAPI commands only.
11	Enabled for all commands

This register defaults to 97609760h at reset.

8.8.17 South Bridge Secondary IDE Timing register PCI Config. F#1 Index 44h-47h (SB_Second_IDE_Time)

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.

Secondary Slave Register (Bits 31-16) **:**

Bits 31-30 *Reserved.*

Bits 29-28 *Reserved.*

Bits 27-26 *Reserved.*

Bits 25-23 **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 8-14 below to determine the number of clocks for recovery times of the read/write signals:

Table 8-14. PIO R/W Recovery Time

Bits 25-23	PCI Clocks	
000	1	
001	2	
010	3	
011	4	
100	8	
101	9	
110	10	Default
111	12	

Bits 22-20 **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 8-15. PIO Active Time Encoding

Bits 22-20	PCI Clocks
000	1
001	2
010	3

Table 8-15. PIO Active Time Encoding

Bits 22-20	PCI Clocks	
011	4	
100	8	
101	9	
110	10	Default
111	12	

Note that the address setup time is implied. Afte initial startup latency, the address setup time is given in Table 8-15:

22-21	Address setup time
00	1 clock
01	2 clocks
1X	3 clocks

Bit 19 **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 **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 **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 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching fo ATAPI commands (when A0h is written into the command register - Register Index 07h for the de vice). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are support ed by ATAPI devices.

Prefetch for the IDE controller is given in Table 8 16:

Table 8-16. IDE Controller Prefetch Encoding

17-16	Prefetch
0X	Completely disabled
10	Enabled for non-ATAPI commands only.
11	Enabled for all commands

Secondary Master Control Register (Bits 15-0) **:**

Bits 15-14 *Reserved.*

Bits 13-12 *Reserved.*

Bits 11-10 *Reserved.*

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 Table 8-17 below to determine the number of clocks for recovery times of the read/write signals:

Table 8-17. PIO R/W Signal Recovery Time

Bits 9-7	PCI Clocks	
000	1	
001	2	
010	3	
011	4	
100	8	
101	9	
110	10	Default
111	12	

Bits 6-4 **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 8-18 below to determine the number of clocks for active times of the read/write signals:

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

Bits 6-4	PCI Clocks	
000	1	
001	2	
010	3	
011	4	
100	8	
101	9	
110	10	Default
111	12	

Note that the address setup time is implied. After initial startup latency, the address setup time is given in Table 8-19:

Table 8-19. Address Setup Time Encoding

6-5	Address setup time
00	1 clock
01	2 clocks
1X	3 clocks

Bit 3 **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 **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 **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 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching fo ATAPI commands (when A0h is written into the command register - Register Index 07h for the de vice). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are support ed by ATAPI devices.

Prefetch for the IDE controller is given in Table 8 20:

Table 8-20. IDE Controller Prefetch encoding

1-0	Prefetch
0X	Completely disabled
10	Enabled for non-ATAPI commands only.
11	Enabled for all commands

This register defaults to 97609760h at reset.

8.8.18 South Bridge Miscellaneous Register PCI Config. F#1 Index 48h (SB_Misc)

This register contains miscellaneous informations.

Bit 7 **Soft Reset.** When set to 1, the IDE controller is reset. It does not affect the timing control reg ister. The FIFOs and the internal state machines are cleared.

Bits 6-4 *Reserved.*

Bit 3 **Disable Secondary channel.** When set to 1 the secondary channel interrupts and accesses

to the secondary channel IDE command and control registers are disabled.

Bit 2 **Disable Primary channel.** When set to 1 the primary channel interrupts and accesses to the primary channel IDE command and control registers are disabled.

Bit 1 **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 **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.

This register defaults to 00h at reset.

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 controller 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 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 ISA bus and 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 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).

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 PCI / ISA CYCLES

9.2.1 PCI TO ISA READ AND WRITE

The PCI transfers data 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 zero 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 50. 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 duplicat ed 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 iDragon, 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, four input pins of the RTC (CS#, AS, RW#, DS) are controlled directly by the iDragon. 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 AS is directly connected to the RTCAS output. AS is the address strobe for the RTC, and it is asserted (high) during any IO write to port 70h.
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 O of the South Bridge outputs RTCDS and ISAOE#. DS is the read pulse for the RTC, and it will be as serted (low) during any IO read of port 71h.
The RTC interrupt output IRQ# is directly connect ed to the IRQ8B input. There is an internal inverte between the pin IRQ8B and the interrupt controlle 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. Fo writes to the keyboard controller that are not inter cepted, 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 iDragon 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 lo 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 gen erated 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 iDragon 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)

DMA 1 controls 8 bit DMA transfers. Channel 0 corresponds to pin DRQ0B, channel 1 to DRQ1B, channel 2 to DRQ2B, and channel 3 corresponds to DRQ3B.

There are 16 DMA 1 registers. They are as shown in Table 9.1

Table 9.1- DMA 1 Registers

IO address bits 15-0	Reset Value	Register Name
XXXX XX00 000x 0000	xxxx xxxx	DMA 1 Channel 0 Base and Current Address
XXXX XX00 000x 0001	xxxx xxxx	DMA 1 Channel 0 Base and Current Count
XXXX XX00 000x 0010	xxxx xxxx	DMA 1 Channel 1 Base and Current Address
XXXX XX00 000x 0011	xxxx xxxx	DMA 1 Channel 1 Base and Current Count
XXXX XX00 000x 0100	xxxx xxxx	DMA 1 Channel 2 Base and Current Address
XXXX XX00 000x 0101	xxxx xxxx	DMA 1 Channel 2 Base and Current Count
XXXX XX00 000x 0110	xxxx xxxx	DMA 1 Channel 3 Base and Current Address
XXXX XX00 000x 0111	xxxx xxxx	DMA 1 Channel 3 Base and Current Count
XXXX XX00 000x 1000	xxxx 0000	DMA 1 Read Status/Write Command register
XXXX XX00 000x 1001	1111 xxxx	DMA 1 Request register
XXXX XX00 000x 1010	0000 0000	DMA 1 Read Command/Write Single Mask register
XXXX XX00 000x 1011	0000 0000	DMA 1 Mode register
XXXX XX00 000x 1100	1111 1111	DMA 1 Set/Clear Byte pointer flip-flop
XXXX XX00 000x 1101	0000 0000	DMA 1 Read Temp register/Master Clear
XXXX XX00 000x 1110	1111 1111	DMA 1 Clear Mask/Clear all requests
XXXX XX00 000x 1111	1111 1111	DMA 1 Read/Write all Mask register bits

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

9.5.2 INTERRUPT CONTROLLER 1 REGISTERS (IC 1)

Table 9.2- Interrupt Controller 1 registers

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

Interrupt controller 1 is the master interrupt controller. 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.

There are two Interrupt controller 1 registers. They are as shown in Table 9.2

Note that not all bits of the address are used.

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. 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 B (see section Section 9.5.4).

There are 4 Interval Timer registers. They are as shown in Table 9.3

Table 9.3- Interval Timer registers

IO address bits 15-0	Reset Value	Register Name
XXXX XX00 010x xx00	xxxx xxxx	Counter 0 count
XXXX XX00 010x xx01	xxxx xxxx	Counter 1 count
XXXX XX00 010x xx10	xxxx xxxx	Counter 2 count
XXXX XX00 010x xx11	1111 1111	Command Mode register

Note that not all bits of the address are decoded.

9.5.4 PORT B (PORT B

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

Bit 7 *Reserved.* This bit is read-only, the state should be ignored by the programmer.

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.

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 **ISA T/C 2 State.** This bit reflects the output of Timer/Counter 2 without any synchronization. This bit is read only.

Bit 4 **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 **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 clea the F/F and then set to a '0' to enable further IO CHK# assertions. This bit is read/write and cleared to a '0' by ISA reset.

Bit 2 *Reserved.* Although this bit is read/write, it is-cleared to a '0' by an ISA reset.

Bit 1 **ISA 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 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.

This register defaults to 00h.

9.5.5 PORT 60 AND 64 (PORT_60 & PORT _64

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 iDragon uses these ports to generate HA20M# and Fast CPU reset.

HA20M# is generated in the following manner. Whenever the iDragon 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 iDragon CPU core. Neither write cycles are forwarded to the keyboard controller.

Fast host CPU only reset is generated by two methods:

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

(2) Whenever the iDragon 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 70

This 8-bit write-only register contains the NMI enable bit and is located at XXXX XX00 0111 0xx1 IO address. Writing to this address also sets the address register in the Real Time Clock (RTC, not part of the iDragon, it is normally connected via the ISA interface).

Bit 7 **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.

Bits 6-0 *Reserved.* must be written to '0's. Read back is undefined.

This register defaults to 80h at reset, disabling NMI generation.

9.5.7 INTERRUPT CONTROLLER 2 REGISTERS (IC2)

Interrupt controller 2 is the slave interrupt control ler. Interrupt controller 2 input IR2 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.

Interrupt controller 2 occupies two register locations. They are as shown in Table 9.4:

Table 9.4- Interrupt Controller 2 registers

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

Note that not all address bits are decoded.

9.5.8 DMA CONTROLLER 2 REGISTERS (DMA 2)

There are 16 DMA 2 registers. They are as shown in Table 9.5

Table 9.5- DMA Controller 2 registers

IO address bits 15-0	Reset Value	Register Name
XXXX XX00 1100 000x	xxxx xxxx	DMA 2 Channel 0 Base and Current Address
XXXX XX00 1100 001x	xxxx xxxx	DMA 2 Channel 0 Base and Current Count
XXXX XX00 1100 010x	xxxx xxxx	DMA 2 Channel 1 Base and Current Address
XXXX XX00 1100 011x	xxxx xxxx	DMA 2 Channel 1 Base and Current Count
XXXX XX00 1100 100x	xxxx xxxx	DMA 2 Channel 2 Base and Current Address
XXXX XX00 1100 101x	xxxx xxxx	DMA 2 Channel 2 Base and Current Count
XXXX XX00 1100 110x	xxxx xxxx	DMA 2 Channel 3 Base and Current Address
XXXX XX00 1100 111x	xxxx xxxx	DMA 2 Channel 3 Base and Current Count
XXXX XX00 1101 000x	1111 xxxx	DMA 2 Read Status/Write Command register
XXXX XX00 1101 001x	0000 0000	DMA 2 Request register
XXXX XX00 1101 010x	0000 0000	DMA 2 Read Command/Write Single Mask register
XXXX XX00 1101 011x	0000 0000	DMA 2 Mode register
XXXX XX00 1101 100x	1111 1111	DMA 2 Set/Clear Byte pointer flip-flop
XXXX XX00 1101 101x	0000 0000	DMA 2 Read Temporary/Master Clear
XXXX XX00 1101 110x	1111 1111	DMA 2 Clear Mask/Clear all requests register
XXXX XX00 1101 111x	1111 1111	DMA 2 Read/Write all Mask register bits

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

9.5.9 DMA PAGE REGISTERS (DMA PAGE)

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.

Table 9.6- DMA Page registers

IO address bits 15-0	Reset Value	Register Name
XXXX XX00 100x 0000	xxxx xxxx	DMA Page Register Port 80 (Reserved)
XXXX XX00 100x 0001	xxxx xxxx	DMA Page Register Channel 2
XXXX XX00 100x 0010	xxxx xxxx	DMA Page Register Channel 3
XXXX XX00 100x 0011	xxxx xxxx	DMA Page Register Channel 1
XXXX XX00 100x 0100	xxxx xxxx	DMA Page Register Port 84 (Reserved)
XXXX XX00 100x 0101	xxxx xxxx	DMA Page Register Port 85 (Reserved)
XXXX XX00 100x 0110	xxxx xxxx	DMA Page Register Port 86 (Reserved)
XXXX XX00 100x 0111	xxxx xxxx	DMA Page Register Channel 0
XXXX XX00 100x 1000	xxxx xxxx	DMA Page Register Port 87 (Reserved)
XXXX XX00 100x 1001	xxxx xxxx	DMA Page Register Channel 6
XXXX XX00 100x 1010	xxxx xxxx	DMA Page Register Channel 7
XXXX XX00 100x 1011	xxxx xxxx	DMA Page Register Channel 5
XXXX XX00 100x 1100	xxxx xxxx	DMA Page Register Port 8B (Reserved)
XXXX XX00 100x 1101	xxxx xxxx	DMA Page Register Port 8C (Reserved)
XXXX XX00 100x 1110	xxxx xxxx	DMA Page Register Port 8D (Reserved)
XXXX XX00 100x 1111	xxxx xxxx	DMA Page Register Port 8E (Reserved)

9.6 ISA CONFIGURATION REGISTERS

9.6.1 MISCELLANEOUS CONTROL REGISTER 0 - INDEX50H (MISC_CONT0)

Bit 7 **ISA Write Post Enable.** If '1', posted writes to host memory by ISA DMA or ISA bus master are enabled.

Bit 6 **ISA Read Buffer Enable.** If '1', buffered reads of host memory by ISA DMA or ISA bus master are enabled.

Bit 5 **ISA Wait Insert Control.** This bit controls if extra wait state is inserted for slower ISA devices.

0 = no extra wait state for ISA cycle.
1 = one extra wait state for ISA cycle.

Bit 4 **ISA Clock Frequency Select.** This bit selects the ISA clock frequency.

0 = ISA clock is 14.31818MHz / 2
1 = ISA clock is PCICLK /4

Bit 3 **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 Deturbo.** These three bits define the ratio CPU is held.(see Table 9.7)

Table 9.7- CPU Deturbo Encoding

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.

This register defaults to 00h at reset.

9.6.2 MISCELLANEOUS CONTROL REGISTER 1 - INDEX51H (MISC_CONT1)

Bit 7 **IPC Write control.** This bit controls the ISA master writes to the IPC register

0 = ISA master writes to IPC register disabled
1 = ISA master writes to IPC register enabled

Bit 6 **CLK24 Enable.** This bit controls the output of CLK24

0 = CLK24 generated normally
1 = Clock synthesizer for CLK24 is disabled (CLK24 will not toggle).

Bit 5 **HCLK Disable.** This bit controls the genera tion of HCLK

0 = HCLK generated normally
1 = Clock synthesizer for HCLK is disabled (HCLK will not toggle).

Bit 4 *Reserved.*

Bit 3 **ROM Write Protect Enable.** This bit, if set to a '1', disables write cycles to ROM BIOS on ex tended 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 af ter shadow is enabled, all write to BIOS should be forwarded to extended bus.

Bit 2 **Segment E Share.** This bit controls if E0000h-EFFFFh segment shares the FLASH memory with F0000h-FFFFFh segment.

0 = sharing disabled
1 = sharing enabled

Bit 1 **Segment D Share.** This bit controls if D0000h-DFFFFh segment shares the FLAS memory with F0000h-FFFFFh segment.

0 = sharing disabled
1 = sharing enabled

Bit 0 **Segment C Share.** This bit controls if C0000h-CFFFFh segment shares the FLAS

memory with F0000h-FFFFFh segment.

0 = sharing disabled
1 = sharing enabled

This register defaults to 00h at reset.

9.6.3 PIRQA ROUTING CONTROL REGISTER 0 - INDEX52H (PAR_CONT0)

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

Bit 7 **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 *Reserved.* Writes have no affect. Reads return undefined value.

Bits 3-0 **Routing Control A#.** These bits route the PCI interrupt A# as follows (see Table 9.8):

Table 9.8- Routing Control A# encoding

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.

This register defaults to 00h at reset.

9.6.4 PIRQB ROUTING CONTROL REGISTER 0 - INDEX 53H (PBR_CONT0)

9.6.5

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

Bit 7 **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 *Reserved.* Writes have no affect. Reads return undefined value.

Bits 3-0 **Routing Control B#.** These bits route the PCI interrupt B# as follows (see Table 9.9):

Table 9.9- Routing Control B# encoding

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.

This register defaults to 00h at reset.

9.6.6 PIRQC ROUTING CONTROL REGISTER 0 - INDEX 54H (PCR_CONT0)

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

Bit 7 **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 *Reserved.* Writes have no affect. Reads return undefined value.

Bits 3-0 **Routing Control C#.** These bits route the PCI interrupt C# as follows (see Table 9.10):

Table 9.10- Routing Control C# encoding

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.

This register defaults to 00h at reset.

9.6.7 PIRQD ROUTING CONTROL REGISTER 0 - INDEX 55H PDR_CONT0)

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

Bit 7 **Routing Enable D#.** If set to a '1', this bit en ables the routing of PCI interrupt, otherwise the PCI interrupt D# is unconnected.

Bit 6-4 *Reserved.* Writes have no affect. Reads re turn undefined value.

Bit 3-0 **Routing Control D#.** These bits route the PCI interrupt D# as follows (see Table 9.11):

Table 9.11 Routing Control D# encoding

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.

This register defaults to 00h at reset.

9.6.8 INTERRUPT LEVEL CONTROL REGISTER 0 - INDEX 56H (IRQ_LEV_CONT0)

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. This register has the following definition.

Bits 7-3 **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 *Reserved.* Writes have no affect and the reads return undefined value.

This register defaults to 00h.

9.6.9 INTERRUPT LEVEL CONTROL REGISTER 1INDEX 57H (IRQ_LEV_CONT1)

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. This register has the following definition.

Bits 7-6 **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 *Reserved.* Writes have no affect and the reads return undefined value.

Bits 4-1 **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 This bit controls the ISA refresh cycle.Setting to 0 disables ISA refresh and setting to 1 enables ISA refresh.

By setting this bit to 1 enables the toggling the ISA Port B refresh bit (see Section 9.5.4)

9.6.10 IPC CONFIGURATION REGISTER - INDEX 01H (IPC_CONT)

This 8-bit register controls the timing of the DMA controllers, and also the number of wait states fo writes to registers in the IPC. To read or write to this register, write 01 to index register 22h, and then read or write from data register 23h.

Bits 7-6 **IPC Wait States.** These bits specify the number of ISACLK wait states for read or write to IPC register1 This is detailed in Table 9.12 .

Table 9.12 IPC Wait state encoding

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

Bits 5-4 **DMA 16-Bit Wait States.** These bits specify the number of wait states in 16-bit DMA cycles This detailed in Table 9.13 .

Table 9.13 DMA 16-Bit Wait State encoding

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

Bits 3-2 **DMA 8-Bit Wait States.** These bits spec ify the number of wait states in 8 bit DMA cycles

Table 9.14 DMA 8-Bit Wait State encoding

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

This is detailed in Table 9.14

Bit 1 **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 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.

This register defaults to C0h.

9.6.11 VMI IRQ ROUTING CONTROL REGISTER - INDEX 58H (VIR_CONT)

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

Bit 7 **VMI Routing Enable.** If set to a '1', this bit enables the routing of VMI interrupt, otherwise the VMI interrupt is unconnected.

Bit 6-4 *Reserved.* Writes have no affect. Reads return undefined value.

Bit 3-0 **VMI Routing Control.** These bits route the VMI interrupt as follows in Table 9.15 .

Table 9.15- VMI Routing Control encoding

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.

This register defaults to 00h at reset.

9.6.12 ISA SYNCHRONIZER BYPASS - INDEX 59H (ISA_SYNC)

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

Bits 7-1 *Reserved.* Writes have no affect. Reads return undefined value.

Bit 0 **Synchronisation Enable.**

0 = Enabled
1 = Disabled

This register defaults to 00h at reset.

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 & 3F6h and 170h-177h and 376h. 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 Initialising. Each entry in the PRD table is 8 bytes long and will have the format below:

10.2. 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 16 bit data.**

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

Bits 31-1 **Memory region physical address of the first descriptort.**

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

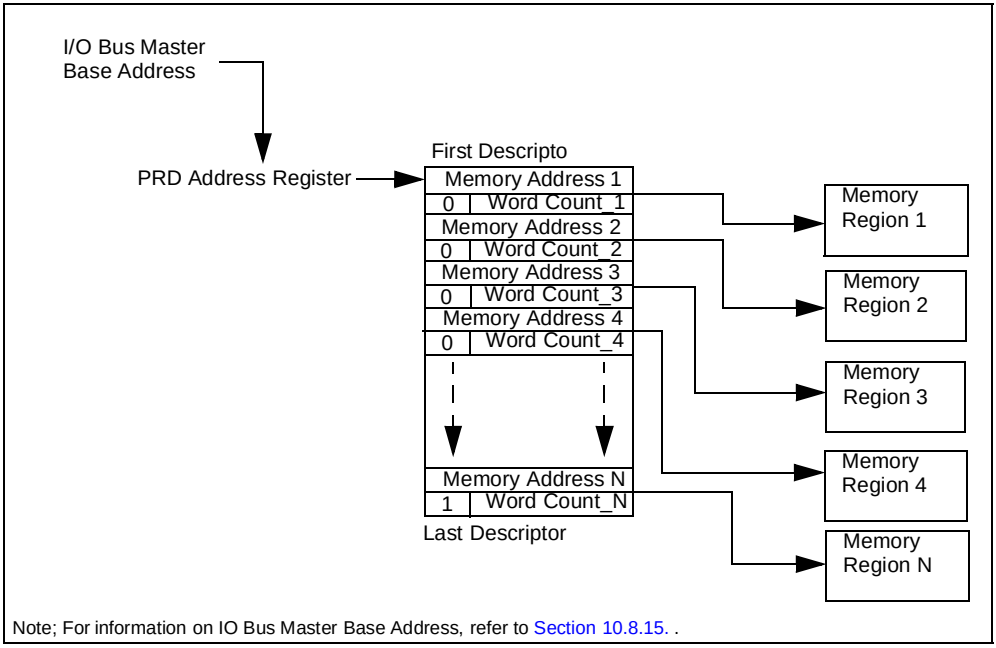
The table must be aligned on a 4 byte boundary and should not cross a 64k boundary.

A memory region also should not cross a 64k boundary. An example of a PRD table is shown in Figure 10-1..

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

To save pins the IDE controller shares pins with the ISA interface. 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 Southt Bridge arbitrates between the IDE controller and the ISA bus bridge to select which has control of the shared pins.

Figure 10-1. PRD Table Entry Example



Note; For information on IO Bus Master Base Address, refer to Section 10.8.15. .

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 de scriptor 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 in crementer 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 re quest. If greater than, then the driver must termi nate the Bus Master transaction (by resetting bit zero of the command register to zero) when the drive issues an interrupt to signal transfer comple tion.

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, or Dword quantities. The description of the 16 bytes of IO registers fol lows:

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

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.

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.

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

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 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.

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.

Bit 7 **Simplex only.** This is hardwired to '0'.

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 controlle has been initialized for optimum performance.

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

Bit 2 **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 **Error.** This bit is set when the controller en counters 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 soft ware.

Bit 0 **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 re gion 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 com mand is visible in system memory, unless the bus master command was aborted.

Reset value is 00h.

10.5.3. 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 10.2.

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

Reset value is 0000 0000h

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 fo 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 regis ter

*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 re sponse 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 in terpret the Interrupt and Active bits in the Control ler 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	

11. VGA CONTROLLER

11.1 Introduction

The iDragon 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 easely 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 iDragon 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 1280 x 1024 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 1280 x 1024 x 16,1024 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 iDragon 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 MBEN 094h (RW)

Bits 7-6 *Reserved,* read as '0's.

Bit 5 **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.

Bit 4 *Reserved,* reads as '0'.

Bit 3 **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 *Reserved,* read as '0's.

If the VGA is configured to operate on an add-in card, the Graphics controller will ignore accesses to port 94h.

The contents of this register are 28h after reset.

Programming notes

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

11.4.2 Add-in VGA Enable Register ADDEN 46E8h (RW)

Bits 7-5 *Reserved,* read as 0's.

Bit 4 **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.

Bit 3 **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.

Bits 2-0 *Reserved,* read as '0's.

If the VGA is configured to operate on an add-in card, the Graphics Controller will ignore accesses to port 46E8h.

This register is defined to be 00h after reset.

Programming notes

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

11.4.3 Video Subsystem Enable 1 register VSE1 102h (RW

Bits 7-1 *Reserved,* read as '0's.

Bit 0 **VSE1 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 affect ed by this bit. The VGA continues to display video data while disabled.

This register is defined to be 00h after reset.

Programming notes

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

11.4.4 Video Subsystem Enable 2 Reg IO Adress 0X3C3h (RW)

Bits 7-1 *Reserved,* read as '0's.

Bit 0 **VSE2 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 accessi ble to allow enabling of the VGA. Port 94h is also not affected by this bit. The VGA continues to dis play video data while disabled.

This register is defined to be 00h after reset.

Programming notes

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

11.4.5 Miscellaneous Output register MISC 3CCh/3C2h (R/W)

Bit 7 **Vertical retrace polarity.**
"0" = active high,
"1" = active low.

Bit 6 **Horizontal retrace polarity**.
"0" = active high,
"1" = active low.

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 11-1 :

Table 11-1. Definition of IBM vertical & horizontal retrace pulses

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

Bit 5 **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 *Reserved*, reads as '0'.

Bits 3-2 **Clock Selects.** Selects one of the four synthesizer pairs when DCLK source is onchip PLL's.

Bit 1 **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 Address.** This bit defines the address map of the following registers ,seeTable 11-2 :

Table 11-2. IO address map

Register	Bit 0='0'	Bit 0='1'
CRTC Registers	03BXh	03DXh
Input #1 Register	03BAh	03DAh

This register is defined to be 00h after reset.

Programming notes

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

11.4.6 Input Status register #0 INP0 3C2h (R)

Bit 7 **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 *Reserved.* These bits read as ones.

Bit 4 **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 *Reserved.* These bits read as zero.

11.4.7 Input Status Register #1 INP1 3XAh (R)

Bits 7-6 *Reserved.* These bits read as zero.

Bits 5-4 **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 11-3 :

Table 11-3. Input status register diagnostics

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

Bit 3 **Vertical Retrace.** A one in this position indicates that a vertical retrace is in progress.

Bits 2-1 *Reserved.* Bit 2 reads as one; bit 1 reads as zero.

Bit 0 **Retrace.** A one in this position indicates that a horizontal OR vertical retrace is in progress.

11.5 VGA Sequencer Registers

11.5.1 Sequencer Index Register SRX 03C4h (RW)

Bits 7-4 *Reserved*, reads as 0's.

Bits 2-0 **Sequencer Index.** These bits point to the register that is accessed by the next read or write to port 03C5h.

The contents of this register are not defined after reset.

Programming notes

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

11.5.2 Sequencer Reset register SR0 03C5h Index 0 (RW)

Bits 7-2 *Reserved,* read as '0's.

Bit 1 **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 **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'.

The contents of this register are not defined after reset.

Programming notes

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

11.5.3 Sequencer Clocking Mode register SR1 03C5h Index 1 (RW)

Bits 7-6 *Reserved,* read as '0's.

Bit 5 **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 **Shift4.** Along with Shift Load (bit 2) this bit controls the loading of the video serializers as in Table 11-4 :

Table 11-4. Screen off video serializer loading

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

Bit 3 **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 in put 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 **Shift Load,** see Bit 4 - Shift4.

Bit 1 *Reserved,* reads as '0'.

Bit 0 **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.

This register is defined to be 00h after reset.

Programming notes

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

11.5.4 Sequencer Plane Mask Register SR2 03C5h Index 2 (RW)

Bits 7-4 *Reserved,* read as '0'.

Bit 3 **Enable Plane 3,** Write enable for plane 3. A '0' in this bit disables writes to plane 3.

Bit 2 **Enable Plane 2.**

Bit 1 **Enable Plane 1**.

Bit 0 **Enable Plane 0.**

The planes are used in different manners by the various modes. These are shown in Table 11-5 :

Table 11-5. Sequenncer Plane mask Modes

Mode	Plane 0	Plane 1	Plane 2	Plane3
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

This register is defined to be 00h after reset.

Programming notes

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

11.5.5 Sequencer Character Map register SR3 03C5h Index 3 (RW)

Bit 5 **Secondary Font Block Select bit 0**

Bit 4 **Primary Font Block Select bit 0**

Bit 3 **Secondary Font Block Select bit 2**

Bit 2 **Secondary Font Block Select bit 1**

Bit 1 **Primary Font Block Select bit 2**

Bit 0 **Primary Font Block Select bit 1**

This register is defined to be ‘0’ after reset.

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 11-6 *and* Table 11-7 *following :*

Table 11-6. Sequencer Character Map Bit 3 = 0

Primary Font				
Bit1	**Bit0**	**Bit4**	**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 11-7. Sequencer Character Map Bit 3 = 1

Secondary Font				
Bit3	**Bit2**	**Bit5**	**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

This register is reset to ‘0’ by the asynchronous re set via SR0 register.

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

11.5.6 Sequencer Memory Mode register SR4 03C5h Index 4 (RW)

Bits 7-4 *Reserved,* read as ‘0’s.

Bit 3 **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 ig nored and the Byte from the plane selected by the two least significant host address bits is returned.

Bit 2 **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 fou 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 **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 *Reserved,* reads as '0'.

This register is defined to be 04h after reset.

Programming notes

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

11.5.7 Extended Register Lock/Unlock SR6 03C5h Index 6 (RW)

Bits 7-0 **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'.

This register is defined to be zero after reset; the extended registers are in the locked state after reset.

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 GRX 03CEh (RW)

Bits 7-4 *Reserved,* reads as '0's.

Bits 3-0 **Graphics Controller Index.** These bits point to the register that is accessed by the next read or write to port 03CFh.

The contents of this register are not defined after reset.

Programming notes

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

11.6.2 Graphics Set/Reset register GR0 03CFh Index 0 (RW)

Bits 7-4 *Reserved,* reads as '0's.

Bits 3-0 **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.

The contents of this register are not defined after reset.

Programming notes

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

11.6.3 Graphics Enable Set/Reset register GR1 03CFh Index 1 (RW)

Bits 7-4 *Reserved,* read as '0'.

Bits 3-0 **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.

The contents of this register are not defined after reset.

Programming notes

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

11.6.4 Graphics Color Compare register GR2 03CFh Index 2 (RW)

Bits 7-4 *Reserved,* read as '0'.

Bits 3-0 **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.

The contents of this register are not defined after reset.

Programming notes

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

11.6.5 Raster Op/Rotate Count register GR3 03CFh Index 3 (RW)

Bits 4-3 **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 11-8 :

Table 11-8. Graphics Controller Raster Op Field Values

Bit4	Bit3	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

Bits 2-0 **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.

The contents of this register are not defined after reset.

Programming notes

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

11.6.6 Graphics Read Map Select register GR4 03CFh Index 4 (RW)

Bits 7-2 *Reserved,* read as '0'.

Bits 1-0 **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.

The contents of this register are not defined after reset.

Programming notes

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

11.6.7 Graphics Mode register GR5 03CFh Index 5 (RW)

Bit 7 *Reserved,* read as '0'.

Bits 6-5 **Shift mode** These values are given in Table 11-9 .

Table 11-9. Graphic Mode Shift Register Behaviour

Bit6	Bit5	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.

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 4 **OddEven.** This bit performs no function. It is, however, readable and writable.

Bit 3 **ReadMode.** 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 Registe (GR4). This is also called Read Mode 0.
When this bit is set to '1', a host read transfer re turns the result of the logical comparison between the data in the four planes selected by the Colo Don't Care Register (GR7) and the contents of the Color Compare Register (GR2). This is also called Read Mode 1.

Bit 2 *Reserved,* reads as '0'.

Bits 1-0 **WriteMode.** These bits select the write mode as follows in Table 11-10

:The contents of this register are not defined afte reset.

Programming notes

Table 11-10. Graphic Mode Write Behaviour

Bit1	Bit0	Write Behaviour
0	0	Write Mode 0
0	1	Write Mode 1
1	0	Write Mode 2
1	1	Write Mode 3

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.

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

11.6.8 Graphics Miscellaneous register GR6 03CFh Index 6 (RW)

Bits 7-4 *Reserved,* read as '0'.

Bits 3-2 **MemoryMap.** These bits specify the map of the display memory buffers in the CPU address space. They are defined as follows in Table 11-11

Table 11-11. Graphics Miscellenous Memory Map

Bit3	Bit2	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)

Bit 1 **Chain2.** This bit performs no function. It is, however, readable and writable.

Bit 0 **GraphicsMode.** When this bit is set to '1' graphics mode is selected; otherwise when set to '0' alphanumeric mode is selected. This bit is du plicated in AR10[0].

The contents of this register are not defined after reset.

Programming notes

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

11.6.9 Graphics Color Don't Care register GR7 03CFh Index 7 (RW)

Bits 7-4 *Reserved,* read as '0'.

Bits 3-0 **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.

The contents of this register are not defined after reset.

Programming notes

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

11.6.10 Graphics Bit Mask register GR8 03CFh Index 8 (RW)

Bits 7-0 **Bit Mask.** Any bit programmed to a '0' in this 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.

The contents of this register are not defined after reset.

Programming notes

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

11.7 Attribute Controller Registers

11.7.1 Attribute Controller Index ARX 3C0h (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).

Bits 7-6 *Reserved.* Must be written as zero.

Bit 5. **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 **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 AR0-ARF 3C1h/3C0h (R/W)

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.

Bits 7-6 **Reserved.** Must be written as zero.

Bits 5-0 **6-bit Color Value.**

11.7.3 Attribute Ctrl Mode register AR10 3C1h/ 3C0h (R/W)

Bit 7 **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 **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 **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 *Reserved.*

Bit 3 **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 **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 **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 **Graphics Mode.** Set this bit to one for graphics mode, zero for text mode.

All the bits in this register reset to zero.

11.7.4 Attribute Ctrl Overscan Color AR11 3C1h/3C0h (R/W)

Bits 7-0 **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 AR12 3C1h/3C0h (R/W)

Bits 7-6 *Reserved.* These bits should be written as zero.

Bits 5-4 **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 11-12 :

Table 11-12. Video Status Mux Control Values

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]

Bits 3-0 **Color Plane Enable.** These four bits are ANDed with the frame buffer data before being in put 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 .

The bits in this register reset to zero.

11.7.6 Attribute Horizontal Pixel Panning AR13 3C1h/3C0h (R/W)

Bits 7-4 *Reserved.* These bits should be written as zero.

Bits 3-0 **Horizontal Pixel Panning.** These bits specify the number of pixels by which to shift the display left. see Table 11-13

Table 11-13. Horizontal Pixel Panning values

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

The bits in this register reset to zero.

11.7.7 Attribute Color Select register AR14 3C1h/3C0h (R/W)

Bits 7-4 *Reserved.* These bits should be written as zero.

Bits 3-2 **Video76.** In all modes except 256 colo mode (mode 13), these bits are output onto bits 7 6 of the VGA pixel data output port.

Bits 1-0 **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.

The bits in this register reset to zero.

11.8 CRT Controller Registers

The iDragon 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 CRX 3X4h (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.

Bits 7 *Reserved.* Must be written to '0's. Read back is undefined.

Bits 6-0 **CRTC Index.** Points to the CRTC register that will be accessed by an IO cycle at 03B5h/ 03D5h.

This register resets to zero.

11.8.2 Horizontal Total register CR0 3X5h Index 0 (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.)

The 8-bit value in this register = Total number of characters - 5.

This register resets to zero.

11.8.3 Horizontal Display end register CR1 3X5h Index 1 (RW)

This 8-bit read/write register defines the total number of displayed characters in a scan line.
The 8-bit value in this register = Total number of displayed characters - 1.

This register is in unknown state after reset.

11.8.4 Horizontal Blanking start register CR2 3X5h Index 2 (RW)

This 8-bit read/write register defines when the hor izontal blanking will start. The horizontal blanking signal becomes active when the horizontal char acter count equals the contents of this register.

This register is in unknown state after reset.

11.8.5 Horizontal Blanking end register CR3 3X5h Index 3 (RW)

Bit 7 *Reserved.* This readable and writable bit must be written to as '1' to ensure proper VGA op eration. It resets to one.

Bits 6-5 **Display Enable Skew Control.** These bits delay the display enable by the specified num ber of character clocks. The result is that the video output stream is delayed by the same amount re sulting in wider left border and shrunk right border. This field is in unknown state after reset.

Bits 4-0 **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 afte 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 Horizontal Retrace start register CR4 3X5h Index 4 (RW)

This 8-bit register defines the character position at which the horizontal sync becomes active.

This register is in unknown state after reset.

11.8.7 Horizontal Retrace end register CR5 3X5h Index 5 (RW)

Bit 7 **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 **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 **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.

This register is in unknown state after reset.

11.8.8 Vertical Total register CR6 3X5h Index 6 (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.

The value programmed in this register = Total number of scan lines - 2.

11.8.9 Overflow register CR7 3X5h Index 7 (RW)

Bit 7 Bit-9 of the 11-bit wide Vertical Retrace start register.

Bit 6 Bit-9 of the 11-bit wide Vertical Display

end register.

Bit 5 Bit-9 of the 11-bit wide Vertical Total register.

Bit 4 Bit-8 of the 11-bit wide Line compare register.

Bit 3 Bit-8 of the 11-bit wide Vertical Blanking start register.

Bit 2 Bit-8 of the 11-bit wide Vertical Retrace start register.

Bit 1 Bit-8 of the 11-bit wide Vertical Display end register.

Bit 0 Bit-8 of the 11-bit wide Vertical Total register.

This register is in unknown state after reset.

11.8.10 Screen A preset row scan register CR8 3X5h Index 8 (RW)

Bit 7 *Reserved.* Must be written to a '0'. Read back is undefined.

Bits 6-5 **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 11-14 :

Table 11-14. Display shift encoding values

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)

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 **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 large 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.

This register is in unknown state after reset and should be changed only during vertical retrace.

11.8.11 Character Cell Height register CR9 3X5h Index 9 (RW)

Bit 7 **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 **Bit-9 of the 11 bit wide Line Compare field.**

Bit 5 **Bit-9 of the 11 bit wide Vertical Blank Start field.**

Bits 4-0 **Scan Lines Per Row.** This field specifies the number of scan lines per character row minus one.

This register is unknown state after reset.

11.8.12 Cursor Start register CRA 3X5h Index A (RW)

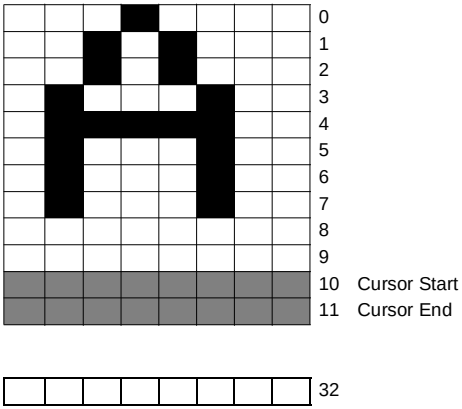
Bits 7-6 *Reserved.* Must be written to '0's. Read back is undefined.

Bit 5 **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 **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 scan lineFigure 11-1

This register is in unknown state after reset.

Figure 11-1Cursor start and end scan line



11.8.13 Cursor End register CRB 3X5h Index (RW)

Bit 7 *Reserved.* Must be written to '0'. Read back is undefined.

Bits 6-5 **Cursor Skew Control.** This field skews the cursor location (defined by the cursor location register) by the specified number of characte clocks to the right.

Bits 4-0 **Cursor End Scan Line.** This field, in conjunction with the Cursor Start Scan line field defines the cursor shape. Ths is illustrated in Figure 11-1.

This register is in unknown state after reset.

Figure 11-1 illustrates the cursor start and end registers

11.8.14 Start Address High register CRC 3X5h Index C (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 CR 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).

11.8.15 Start Address Low register CRD 3X5h Index D (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.

11.8.16 Text Cursor Offset High register CRE 3X5h Index E (RW)

Bits 7-0 **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.

This register is undefined after reset.

11.8.17 Text Cursor Offset Low register CRF 3X5h Index F (RW)

This register is the VGA compatible Cursor Offset Low register.

Bits 7-0 **Cursor offset bits 7-0.** This is the lower half of the cursor offset.

This register is undefined after reset.

11.8.18 Vertical Retrace Start register CR10 3X5h Index 10 (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.

11.8.19 Vertical Retrace End register CR11 3X5h Index 11 (RW)

Bit 7 **CR Protect.** This bit when set to a '1', write protects CR0-7 registers except CR7 bit 4.

Bit 6 *Reserved.* This bit is both readable and write able.

Bit 5 **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 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 **Vertical Retrace Width.** These bits deter mines 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.

This register defaults to binary 0x10xxxx after re set.

11.8.20 Vertical Display End register CR12 3X5h Index 12 (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 overflo register and the bit-10 in the Repaint Control Reg ister 4.

Programming notes

The value in this register = Total number of displayed scan lines - 1.

11.8.21 Offset register CR13 3X5h Index 13 (RW)

This register defines bits 7-0 of the 10-bit wide log ical width of the line displayed on the screen. Ex tended register CR19 contains the upper two bits. The first scan line displayed on the screen, starts at the address specified in registers CRC, CR and extended register CR19. The starting address of the next scan line is computed as the Byte start ing address of the current row + 2*offset.

This register is undefined after reset.

11.8.22 Underline Location register CR14 3X5h Index 14 (RW)

Bit 7 *Reserved.* Must be written to a '0'. Read back is undefined.

Bit 6 **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 **Count by 4.** Setting this bit to one causes the memory address counter to increment every four character clocks.

Bits 4-0 **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 **Underline Enable.** Setting this bit to '1' enables underlining.

This register is undefined after reset.

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 Blanking Start reg CR15 3X5h Index 15 (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.

This register is undefined after reset.

11.8.24 Vertical Blanking End register CR16 3X5h Index 16 (RW)

This 8-bt register defines the width of the vertical blanking pulse as follows:

This register is undefined after reset.

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 CR17 3X5h Index 17 (RW)

Bit 7 **H/V Retrace Enable.** A '0' in this bit position disables the horizontal and vertical retraces and a '1' enables them.

Bit 6 **Byte/Word# Mode.** A '1' value in this bit position selects the Byte mode and a '0' selects the word mode. Following Table 11-15 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.

Table 11-15. Memory addressing for Byte, Word and Dword Generation

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

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'.

Bit 5 **VGA Memory Address Size.** When set to a '0', this bit, in Word addressing mode (see above propagates bit iA13 on the least significant memo

ry 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 *Reserved.* Must be written to a '0'. Read back is undefined.

Bit 3 **Count by 2.** Setting this bit to one causes the memory address counter to increment every second character clock.

Bit 2 **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 **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 **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'.

This register defaults to 00h after reset.

11.8.26 Line Compare register CR18 3X5h Index 18 (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.

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 horizontal ly with Screen A or not.

Split screen function can be disabled by program ming the Line compare field to a value larger, typ ically 3FFh, than the Vertical Total field. This field must be programmed to 3FFh for optimal system performance in native display modes.

This register is undefined after reset.

11.8.27 Graphics Control Data CR22 3X5h Index 22 (R)

Bits 7-0 **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 registe (GR4) specify which latch is read.

The contents of this register is not defined after re set.

Programming notes

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

11.8.28 Attribute Address Flipflop CR24 3X5h Index 24 (R)

Bit 7 **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 *Reserved.* Read as zero.

11.8.29 Attribute Index Readback CR26 3X5h Index 26 (R)

Bits 7-6 *Reserved.* Read as zero.

Bit 5 **Palette Address Source.** This is a read-only copy of Attribute Controller Index register (ARX bit 5.

Bits 4-0 **Attribute Controller Index.** This is a read-only copy of bits 4-0 of the Attribute Control ler 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 AX, 5706h

out DX, AX

11.9.1 Repaint Control Register 0 CR19 3X5h Index 19 (RW)

Bit 7-6 *Reserved.* Must be written to '0'.

Bits 5-4 **CRTC Offset register Bits 9-8.** See CRTC register 13 for details.

Bits 3-0 **CRTC Start Address Field Bits 19-16.** See CRTC register C, D for explanation of the Start Address.

This register defaults to 00h after reset.

11.9.2 Repaint Control Register 1 CR1A 3X5h Index 1A (RW)

Bit 7 **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 **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 *Reserved.* This bit always reads as a one.

Bit 4 **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 *Reserved.* This bit always reads as a one.

Bit 2 **Line Compare Enable.** Set this bit to zero for 1280x1024 mode and one for all others.

Bit 1 **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 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 ze roed).
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.

This register resets to 3Fh.

Programming notes

Note; Vertical / Horizontal Synch not toggling is defined by the VESA specification for Monito Power Down State.

11.9.3 Repaint Control Register 2 CR1B 3X5h Index 1B (RW)

Bits 7-3 **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. Al so, do not set this register to a value greater than that programmed into the high water mark (regis ter CR27).

Bit 2 *Reserved.* This bit is not writable. It reads as zeroes.

Bit 1 **Video FIFO Underflow.** This read-only bit is set to one when the video refresh FIFO under flows. As with bit 0, writes to this register clear this bit to zero.

Bit 0 **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 wate mark value) reset this bit to zero.

This register defaults to 20h at reset.

11.9.4 Repaint Control Register 3 CR1C 3X5h Index 1C (RW)

Bit 7 **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 *Reserved,* read as '0's.

Bits 4-3 *Reserved.* These bits read as '0's.

Bit 2 **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 **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 11-16

Table 11-16. Page Select Control values

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

Bit 0 **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 soft ware 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 Table 11-17 :

Table 11-17. Enable Overlapped Page Bit MA18 & MA17

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	

for Sequential Chain-4, MA16 and MA15 are changed as follows in Table 11-18 :

Table 11-18. Enable Overlapped Page Bit MA16 & MA15

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	

This register is defined to be 00h after reset.

Programming notes

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

11.9.5 Page Register 0 CR1D 3X5h Index 1D (RW)

Bit 7 *Reserved,* read as '0'.

Bits 6-0 **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 Table 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.

The contents of this register is not defined after reset.

11.9.6 Page Register 1 CR1E 3X5h Index 1E (RW)

Bit 7 *Reserved*, read as '0'.

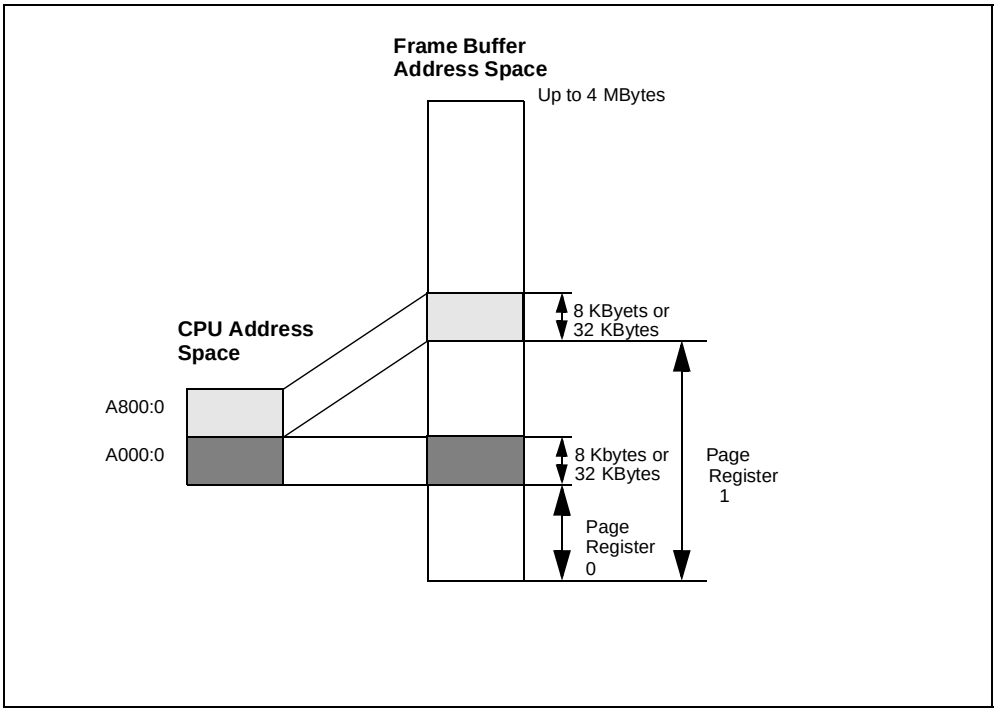
Bits 6-0 **Page 1 Bits 6-0.** 7-bit Page register 1 is used to extend the host address to allow the VGA to be located anywhere in the 4 MB frame buffer space. The pages are located on 32K boundaries for Normal, Odd/Even and Chain-4 modes and on 8K boundaries for Sequential Chain-4 mode. This is in Table 11-2

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.

The contents of this register is not defined after re set.

Figure 11-2. Illustration of Page Register 0 and Page Register 1



11.9.7 Graphics Extended Enable Register CR1F 3X5h, Index 1F (RW

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'.

Bits 6-0 *Reserved,* read as '0's

This register is defined to be 00h after reset.

Programming notes

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

11.9.8 Graphics Extended GBASE Register CR20 3X5h, Index 20 (RW)

Bits 7-3 *Reserved,* these bits read as '0'.

Bits 2-0 **Gbase.** This range defines the bits 26 to 24 of the CPU address space where the GE Extended Frame Buffer and registers are located.

This register is defined to be 00h after reset.

Programming notes

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

11.9.9 Graphics Extended Aperture Register CR21 3X5h, Index 21 (RW)

Bits 7-0 **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.

This register is defined to be FFh after reset.

Programming notes

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

11.9.10 Repaint Control Register 4 CR25 3X5h Index 25 (RW

Bits 7-6 *Reserved,* read as '0's

Bit 5 *Reserved.*

Bit 4 **Bit 6 of the 7-bit wide Horizontal Blanking End register.**

Bit 3 **Bit 10 of the 11-bit wide Vertical Blanking start register.**

Bit 2 **Bit 10 of the 11-bit wide Vertical Retrace start register.**

Bit 1 **Bit 10 of the 11-bit wide Vertical Display end register.**

Bit 0 **Bit 10 of the 11-bit wide Vertical Total reg ister.**

This register is set to 00h after reset.

11.9.11 Repaint Control Register 5 CR27 3X5h Index 27 (RW

Bits 7-3 **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 *Reserved*, read as '0'.

This register is set to a value of D0h after reset.

11.9.12 Palette Control Register CR28 3X5h Index 28 (RW

Bit 7 **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 Setup.** This bit specifies the blanking pedestal. Zero indicates a blanking pedestal of 0 IRE, one indicates 7.5 IRE.

Bit 5 **Sense Power Down.** Setting this bit to one forces the DDC monitor sense circuits to powe down.

Bit 4 *Reserved.* Must be writtezn to '1'.

Bit 3 **LUT Bypass.** Setting this bit to one bypasses the RAMDAC look up table (LUT) and allows pix els 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 **Pixel Format.** These 3-bits specify the pixel color depth and are encoded as follows in Table 11-19 :

Table 11-19. Palette Control Pixel Format encoding

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

This register resets to 08h after reset.

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 (see Table 11-20):

Table 11-20. Supported resolutions at 75 Hz with 64 bit DRAM banks

Pixel Format	Maximum Resolution
VGA (other than mode 13)	1280 x 1024
VGA (mode 13)	640 x 480
8 bit (non-VGA)	1280 x 1024
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 CR29 3X5h Index 29 (RW)

For the description of this register, see section section 12.11.1

Must be written to ‘0’ when not using a Hardware cursor

11.9.14 Cursor Color 0 Register A CR2A 3X5h Index 2A (RW)

For the description of this register, see section section 12.11.2

Must be written to ‘0’ when not using a Hardware cursor

11.9.15 Cursor Color 0 Register B CR2B 3X5h Index 2B (RW)

For the description of this register, see section section 12.11.3

Must be written to ‘0’ when not using a Hardware cursor

11.9.16 Cursor Color 0 Register C CR2C 3X5h Index 2C (RW)

For the description of this register, see section section 12.11.4

Must be written to ‘0’ when not using a Hardware cursor

11.9.17 Cursor Color 1 Register A CR2D 3X5h Index 2D (RW)

For the description of this register, see section section 12.11.5

Must be written to ‘0’ when not using a Hardware cursor

11.9.18 Cursor Color 1 Register B CR2E 3X5h Index 2E (RW)

For the description of this register, see section section 12.11.6

Must be written to ‘0’ when not using a Hardware cursor

11.9.19 Cursor Color 1 Register C CR2F 3X5h Index 2F (RW)

For the description of this register, see section section 12.11.7

Must be written to ‘0’ when not using a Hardware cursor

11.9.20 Graphics Cursor Address Register 0 CR30 3X5h Index 30 (RW)

For the description of this register, see section section 12.11.8

Must be written to '0' when not using a Hardware cursor

11.9.21 Graphics Cursor Address Register 1 CR31 3X5h Index 31 (RW)

For the description of this register, see section section 12.11.9

Must be written to '0' when not using a Hardware cursor

11.9.22 Graphics Cursor Address Register 2 CR32 3X5h Index 32 (RW)

For the description of this register, see section section 12.11.10

Must be written to '0' when not using a Hardware cursor

11.9.23 Urgent Start CR33 3X5h Index 33 (RW)

Bits 7-0 **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.

A typical value to be programmed is the one in CR00.

11.9.24 Displayed Frame Y Offset 0 CR34 3X5h Index 34 (RW

Bits 7-0 **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.

This register is undefined after reset.

11.9.25 Displayed Frame Y Offset 1 CR35 3X5h Index 35 (RW

Bits 7-0 **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.

This register is undefined after reset.

11.9.26 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 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

.

Table 11-21. Implementation Number

VALUE	IMPLEMENTATION
01h	iDragon Implementation

11.9.28 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.

Table 11-22. Graphics Version Number

VALUE	IMPLEMENTATION
01h	iDragon Implementation

11.9.29 DRAM Timing Parameter register CR3C 3X5h Index 3C (RW)

This register defines the type of memory populated and controls RAS# and CAS# timing.

Bit 7-6 **Memory type**:

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

Bit 5-3 These three bits control the allowed timing templates as in Table 11-23.:

While the suggested use of different templates is listed above, any template can be used with eithe 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:

Bit	Description
0	0.5 clock wide pulse width.
1	0.75 clock wide pulse width.

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.

Bit	Decription
0	1 clock delay
1	2 clock delay

It has to be programmed to 1 (2-clock delay) for -60 EDO parts.

Bit 0 **RASoff**: This bit controls if RAS is kept active after the current DRAM access.

Bit	Description
0	Keep RAS# active
1	deassert RAS#

DRAM Timing Parameters default to 00h.

Table 11-23. Timing Template Settings

Bit 5	Bit 4	Bit 3	cas-to-ras	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

11.9.30 DRAM Arbitration control register 0 CR3D 3X5h Index 3D (RW)

Bits 7-0. *Reserved.* Must be set to '0'.

11.9.31 DRAM Arbitration control register 1 CR3E 3X5h Index 3E (RW)

Bits 7-0. *Reserved.* Must be set to '0'.

11.9.32 DDC Control Register CR3F 3X5h Index 3F (RW)

Bits 7-6 **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 Read Data.** These read-only bits return the read status of the DDC[1:0] pins.

Bits 3-1 *Reserved.* These bits read are both readable and writable and must be programmed to ones to ensure future compatibility.

Bit 0 *Reserved*. This bit must be programmed to '1' for correct operation.

This register defaults to FFh after reset.

11.9.33 TV Interface Control Register CR40 3X5h Index 40 (RW)

Bit 7 **TV Interface Enable.** This bit enables the TV interface when set high.

Bit 6 **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 **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 **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 11-24):

Table 11-24. Bottom/Top Output Enable Truth Table

CR40[4]	VIP2B[29]	VTV_BT direction
0	0	input
0	1	output from VIP t/g
1	X	output from TVO t/g

Bit 3 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 11-25):

Table 11-25. HSYNC Output Enable Truth Table

CR40[4]	VIP2B[29]	VTV_HSYNC direction
0	0	input
0	1	output from VIP t/g
1	X	output from TVO t/g

Bits 2-0 **Flicker Filter Algorithm.** These bits control the operation of the anti-flicker filter according to Table 11-26 :

Table 11-26. Flicker Filter Algorithm

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

11.9.34 TV Horizontal Active Video Start A CR41 3X5h Index 41 (RW)

Bits 7-0 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 CR42 3X5h Index 42 (RW)

Bit 7 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 Reserved. These read/write bits should be written as zeroes to ensure future compatibility.

Bit 3 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 11-27 .

Table 11-27. : TV Output mode

Bit 3	RED_TV	GREEN_TV	BLUE_TV	CVBS
0	red	green	blue	composite
1	chrominance	luminance	composite	composite

Bits 2-0 These represent bits 10:8 of the eleven bit wide horizontal active video start field. See section 11.9.34 .

11.9.36 TV Horizontal Sync End A CR43 3X5h Index 43 (RW)

Bits 7-0 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 CR44 3X5h Index 44 (RW)

Bit 7 *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 *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 *Reserved.* These read/write bits should be written as zeroes to ensure future compatibility.

Bits 2-0 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 buffe 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 3C6h (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.

This register defaults to FFh after reset.

11.12.2 Palette Read index register 3C7h (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.

11.12.3 Palette State register 3C7h (R)

This is a read only register and contains the two least significant bits of the last IO writes to IO ad dress 3C6h-3C9h.

Bits 7-2 *Reserved.* The read back of this register is undefined.

Bits 1-0 These bits contain the 2 LSBs of the ad dress 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 3C8h (RW)

This register is similar to the Read index registe 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 in dex 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 Pal ette state register.

11.12.5 Palette Data register 3C9h (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 registe 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 regis ters 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 afte 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 ta ble 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.

This register is 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 DCLK00 Index 42

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.

Bit 7 *Reserved*.

Bits 6-3 This the 4-bit M (divisor) value of the Dot Clock Synthesizer.

Bits 2-0 These are bits 7-5 of the 8 bit N (mulitiplier) of the Dot Clock synthesizer.

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 DCKL01 Index 43

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.

Bit 7 This is the bit 0 of the 3-bit P (exponent) value of the Dot clock synthesizer.

Bits 6-2 These are bits 4-0 of the 8-bit N (multiplier) value of the Dot Clock Synthesizer.

Bits 1-0 These are bits 2-1 of the 3-bit P (exponent) value of the Dot clock synthesizer.

This register defaults to 0x95h at reset. This value when combined with the default value of the othe half of this pair results in a Dot Clock of 25.18 MHz assuming 14.318 MHz oscillator clock as the ref erence input.

11.13.3 DCLK control register 10 DCLK10 Index 44

This is one of the 4-pairs of dot clock control reg isters. This pair (10 and 11) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 01.

Bit 7 *Reserved*.

Bits 6-3 This the 4-bit M (divisor) value of the Dot Clock Synthesizer.

Bits 2-0 These are bits 7-5 of the 8 bit N (mulitipli er) of the Dot Clock Synthesizer.

This register defaults to 0x76h at reset. This value when combined with the default value of the othe half of this pair results in a Dot Clock of 28.0 MHz assuming 14.318 MHz oscillator clock as the ref erence input.

11.13.4 DCLK control register 11 DCLK11 Index 45

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.

Bit 7 This is the bit 0 of the 3-bit P (exponent) val ue of the Dot Clock Synthesizer.

Bits 6-2 These are bits 4-0 of the 8-bit N (multipli er) value of the Dot Clock Synthesizer.

Bits 1-0 These are bits 2-1 of the 3-bit P (expo nent) value of the Dot Clock Synthesizer.

This register defaults to 0xEDh at reset. This value when combined with the default value of the othe half of this pair results in a Dot Clock of 28.0 MHz assuming 14.318 MHz oscillator clock as the ref erence input.

11.13.5 DCLK control register 20 DCLK20 Index 46

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.

Bit 7 *Reserved*.

Bits 6-3 This the 4-bit M (divisor) value of the Dot Clock Synthesizer.

Bits 2-0 These are bits 7-5 of the 8 bit N (mulitiplier) of the Dot Clock Synthesizer.

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 DCLK21 Index 47

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.

Bit 7 This is the bit 0 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.

Bits 6-2 These are bits 4-0 of the 8-bit N (multiplier) value of the Dot Clock Synthesizer.

Bits 1-0 These are bits 2-1 of the 3-bit P (exponent) value of the Dot Clock Synthesizer.

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 DCLK30 Index 48

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.

Bit 7 *Reserved*.

Bits 6-3 This the 4-bit M (divisor) value of the Dot Clock Synthesizer.

Bits 2-0 These are bits 7-5 of the 8 bit N (mulitipli er) of the Dot Clock Synthesizer.

This register defaults to 0x6Eh at reset. This value when combined with the default value of the othe half of this pair results in a Dot Clock of 60 MHz assuming 14.318 MHz oscillator clock as the ref erence input.

11.13.8 DCLK control register 31 DCKL31 Index 49

This is one of the 4-pairs of dot clock control reg isters. This pair (30 and 31) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 11.

Bit 7 This is the bit 0 of the 3-bit P (exponent) val ue of the Dot Clock Synthesizer.

Bits 6-2 These are bits 4-0 of the 8-bit N (multipli er) value of the Dot Clock Synthesizer.

Bits 1-0 These are bits 2-1 of the 3-bit P (expo nent) value of the Dot Clock Synthesizer.

This register defaults to 0x69h at reset. This value when combined with the default value of the othe half of this pair results in a Dot Clock of 60 MHz assuming 14.318 MHz oscillator clock as the ref erence input.

12. GRAPHICS ENGINE

12.1 INTRODUCTIO

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.

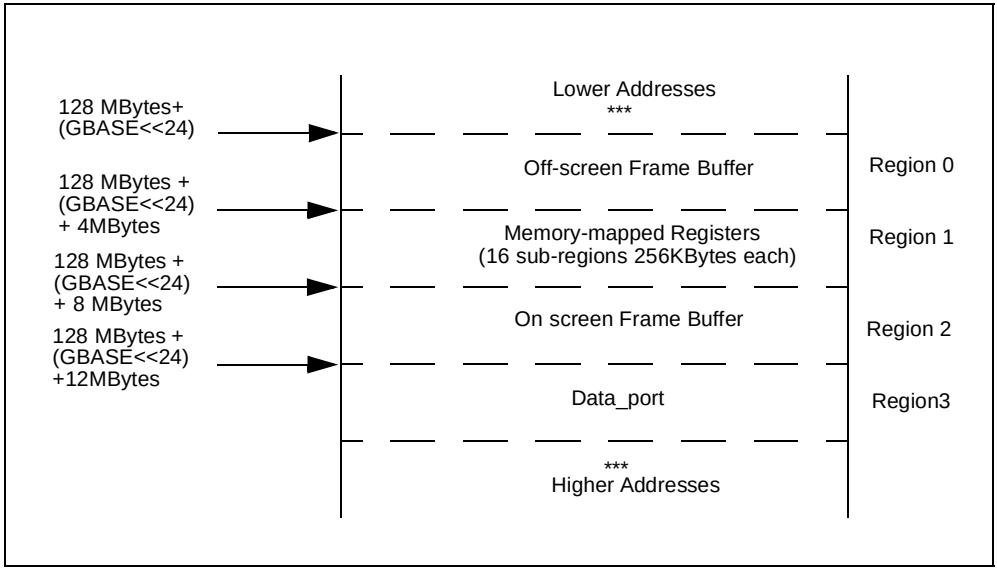
12.2 MEMORY ADDRESS SPACE

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 12-1GE 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:

Table 12-1Graphic memory subdivisions

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

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 ac cess the frame buffer as if it were an unacceleratd 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 ad dresses. The former method allows direct access to the 16 MByte GE address range via 32-bit addresses. The A0000h-AFFFFh addressing meth od maps a 64K window of the 16 MByte GE ad dress space into the address range A0000h-AF FFFh. For additional detail on the A0000h-AF FFFh 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 remap ping 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 sub-system, 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 colo transparent mode, then the “Pat” field of the ROP register must be set to ‘1’. If no pattern or colo 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 correspond ing 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 regis ter 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 :

Table 12-2 Detail GE starting address register

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)

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 :

Table 12-3Shift values supported

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

The operand base addresses must be aligned to 32 Bytes (that is, the 5 least significant bits of the address must be zeros).

Table 12-4Encoded Dst_XY registers

				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		

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 12-4 :

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 12-5 :

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.

Table 12-5CMD operations

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)

The ROP register also has fields that control the Source data (Screen, Host or Foreground colo register), enables/disables bitmap expansion, de termines if the drawing is done in one of the trans parent modes.

Screen-to-screen BitBlts are done as a BitBlt with the Source set as the Screen and bitmap expan sion disabled.

Host-to-screen BitBlts are done as a BitBlt with the Source set as Host and the bitmap expansion dis abled.

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.

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 bit map expanded Host-to-screen BitBlts), the CPU writes 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.

Table 12-6GE and Data_Port access

				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>			

12.9 REGISTER SPECIFICATION

The GE registers are listed in alphabetical order and defined below.

12.9.1 BACKGROUND COLOR REGISTER INDEX = 004H (BACKGROUND)

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.

Bits 31-0 **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.

The content of this register is defined to be zero after reset.

Programming notes

The content of this register is not altered by drawing operations.

12.9.2 CURSOR COORDINATE REGISTER INDEX = 11CH (CURSOR_XY)

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.

Bits 31-29 *Reserved.*

Bits 28-16 **CYUL**. The Y location of the upper-left hand corner of the cursor.

Bits 17-13 *Reserved.*

Bits 12-0 **CXUL**. The X location of the upper-left hand corner of the cursor.

The contents of this register are not defined after reset.

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 INDEX = 804H (DATA_PORT)

This write-only register is the port through which the CPU provides Host data.

Bits 31-0 **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.Destination Operand Base Address Register Index = 018h (DST_Base)Destination Operand Base Address Register Index = 018h (DST_Base)

12.9.4 DESTINATION OPERAND BASE ADDRESS REGISTER INDEX = 018H (DST_BASE)

This register specifies the starting DRAM linear address of the destination operand (aligned to a 32 Byte boundary).

Bits 31-21 *Reserved.*

Bits 21-0 **DstOp_Base.** Base DRAM linear address of the destination operand with 16 Byte alignement. Lower five Bytes are reserved and are set to '0'.

The contents of this register are not defined after reset.

Programming notes

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

12.9.5 DESTINATION PITCH REGISTE INDEX = 028H (DIST_PITCH)

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.

12.9.6 DESTINATION OPERAND COORDINATE REGISTER INDEX = 10000H FOR STANDARD USE (DST_XY)

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.

Bits 31-11 *Reserved.*

Bits 10-9 **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.

Table 12-7DRAM address multiplication factor

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

Bits 8-6 **Dst_shift2.** See Dst_shift3, above.

Bits 5-3 **Dst_shift1.** See Dst_shift3, above.

Bits 2-0 **Dst_shift0.** See Dst_shift3, above.

This register can be accessed via 32-bit or 16-bit transfers.

The contents of this register are not defined after reset.

Programming notes

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

Bits 31-16 **Dst_Y.** The unsigned Y coordinate of the starting corner of the destination operand.

Bits 15-0 **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.

The contents of this register are not defined after reset.

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 INDEX = 034H (FRG_CL)

This register contains the full-color value(s) that a "1" bit is expanded to.

Bits 31-0 **Frg_Cl.** This is the color to be used as the foreground when expanding bitmap '1' values

The contents of this register are defined to be FFFFFFFFh.

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 INDEX = 048H (HEIGHT)

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.

Bits 31-16 *Reserved.*

Bits 15-0 **Height.** The value set in these bits must be one less than the height, in scan lines, of the source and destination areas.

The contents of this register are not defined after reset.

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 Table 12.6.1

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 INDEX = 058H (PATTERN)

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.

Bits 31-22 *Reserved.*

Bits 21-8 **Pattern base.** These bits specify the starting DRAM physical address of the Pattern, operand, aligned to a 256 Byte boundary

Bits 7-5 *Reserved.*

Bits 4-3 **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 *Reserved.*

The contents of this register are not defined after reset.

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 INDEX = 07CH (PIXEL_DEPTH)

This register contains the number of Bytes in a pixel, and bits that control the direction of Screen to-screen BitBlts.

Bits 31-8 *Reserved.*

Bit 7 **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-bitmap-expanded screen-to-screen BitBlts.

Bit 6 **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_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 *Reserved.*

Bits 1-0 **Pixel depth.** The only supported values for this field are shown in Table 12-8 :

Table 12-8Supported Pixel depth values

Bit1	Bit0	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

The contents of this register are not defined after reset.

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 INDEX = 08CH (ROP)

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

Table 12-9Summary of ROP Functions

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

Bits 31-30 **Source.** These bits determine the SRC operand. Possible values are shown in Table 12-10 :

Table 12-10Detail of SRC operand 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	

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 com plete without error, and the data will be ignored unless the Source field is set to HOST.

Bit 29 **PAT present.** This bit is set to '1' if a pattern data is to be used during the operation.

Bit 28 **DST present** This bit is set to '1' if destina tion data is to be read during the operation.

Bit 27 **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 *Reserved.*

Bit 15 **DST transparency mode.** When this is set to '1', pixels are selectively modified based upon a comparison of the DST data from the frame buffe 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 **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 13 **SRC transparency mode.** When this is set to '1', pixels are selectively modified based upon a comparison of the SRC data vs. the SRC trans parency compare register. The results of the com parison are interpreted based upon the SR transparency match bit. This mode is only mean ingful 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 **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 **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 **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 **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 **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 **ROP**, the raster operation used when computing a pixel result value.

The contents of this register are not defined after reset.

Programming notes

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

12.9.12 SOURCE OPERAND BASE ADDRESS REGISTER INDEX = 098H (SCR_BASE)

This register specifies the starting DRAM linear address of the source operand (aligned to a 32 Byte boundary).

Bits 31-22 *Reserved.*

Bits 21-5 **SrcOp_Base.** Base linear address of the source operand.

Bits 4-0 *Reserved.* These bits are set to '0'.

The contents of this register are not defined after reset.

Programming notes

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

12.9.13 SOURCE PITCH REGISTER INDEX = 0ACH (SRC_PITCH)

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.

Bits 31-11 *Reserved.*

Bits 10-9 **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.

Table 12-11Scr_shift3 multiplication factors

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

Bits 8-6 **Src_shift2.** See Src_shift3, above.

Bits 5-3 **Src_shift1.** See Src_shift3, above.

Bits 2-0 **Src_shift0.** See Src_shift3, above.

The contents of this register are not defined after reset.

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 INDEX = 0BCH (SCR_XY)

This register contains the coordinate address of the starting corner of the source operand.

Bits 32-16 **Src_Y.** The source operand starting corner of the unsigned Y coordinate.

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.

The contents of this register are not defined after reset.

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 bit map 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
INDEX = 908H (STATUS)

Bit 31 **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 **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 *Reserved.* These may read as one or zero.

After reset, bits 31-30 read as zero. All other bits are undefined.

12.9.16 WIDTH REGISTER
INDEX = 0C8H (WIDTH

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.

Bits 31-16 *Reserved.*

Bits 15-0 **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.

The contents of this register are not defined after reset.

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
INDEX = 0D4H (XTRA)

This register contains 32 bits of data that software can read from and write to.

This register has no effect on any drawing opera tion or display.

Bits 31-0 **Data for user software use.**

The contents of this register are not defined after reset.

Programming notes

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

12.9.18 SRC TRANSPARENCY COMPARE
INDEX = ECH (ST_COMP)

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 val ue must be replicated in the upper and lower 16 bits of this register.

12.9.19 DST TRANSPARENCY COMPARE
INDEX = FCH (DT_COMP)

This 32-bit register contains the pixel value used for comparison in DST transparency mode. Fo 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 val ue must be replicated in the upper and lower 16 bits of this register.

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 FFFFFFFh 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 BITMAP 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 sig nificant. 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 contin ues 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 BITBLT 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):

1) Set the starting base address of the screen or region into Dst_base
2) Set the pitch of the screen or region into Dst_pitch
3) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to '0',
4) Set Width.Width to one less than the width of the rectangular area, in Bytes
5) Set Height.Height to one less than the height of the rectangular area, in Bytes
6) Set ROP.ROP to the raster operation, ROP.Expand to "0", and ROP.Transparent OPAQUE
6) Set ROP.Source to CONSTANT_FILL
7) Set the Background color register to the colo to be painted
8) Set the Pixel_depth for the number of Bytes per-pixel
9) If a pattern is to be applied, set the Pattern reg ister with the starting address of the pattern data
10) Write the upper-left coordinates of the area to be filled 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, 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 buffe at its full bandwidth. Thus, result pixels are computed in groups of 32 bits, to allow one 64-bit re sult 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 BITBLT

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)) -> (Des tination)

If these rectangular areas are overlapping, then the direction of the BitBlt must be carefully select ed:

* *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) Set the Pixel_depth for the number of Bytes-per-pixel
11) If a pattern is to be applied, set the Pattern register with the starting address of the pattern data
12) If ROP.Expand is set, then also set the Foreground and Background color registers
13) Write the starting coordinates of the source area to be copied into the Src_XY register
14) 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 buffe 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 BITBLT

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 CP 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).

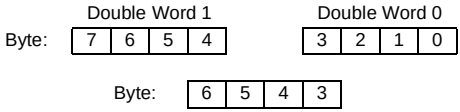
The function performed during the BitBlt is:

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 raste 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:



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):

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 Src_pitch to 0 if packed source data or to a non-zero value, otherwise
4) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to '0'
5) Set Width.Width to one less than the width of the destination rectangular area, in Bytes
6) Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values
7) Set ROP.Source to HOST
8) If Color Transparency mode is set, set the Background color register to the color to be made transparent
9) Set the Pixel_depth for the number of Bytes-per-pixel
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) Set Src_XY to indicate alignment of the Host data (only the least 3 significant bits are important)
13) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.
14) 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 o 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 raste 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

Table 12-12Bit representation

top line					bottom line
01110	10001	00010	00100	00000	00100

Increasing memory addresses ------>

Breaking this up into Bytes see Table 12-13

Table 12-13Byte representation

	01110100	01000100	01000000	000100XX
Byte	0	1	2	3

Breaking this up into a double-word as inTable 12-14 :

Table 12-14Double word representation

	000100XX	01000000	01000100	01110100
Byte	3	2	1	0
Word	0		1	

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.

To perform a Packed Text 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 ROP.Packed to 1 for packed source data
4) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, Pixel_depth.Y_dst_dir to '0'
5) Set Height to the height of the destination character, in scan lines
6) Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values
7) Set ROP.Source to HOST
8) Set the Pixel_depth for the number of Bytes per-pixel
9) If a pattern is to be applied, set the Pattern register with the starting address of the pattern data
10) set the Foreground and Background color registers
11) Set Src_XY to indicate alignment of the Host data (only the least 5 significant bits are important)
12) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.
13) 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_di and Y_dst_dir fields of the Pixel_depth registe 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 :

Table 12-15Cursor Arrays

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

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.

12.11.1 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 curso XOR operation is performed before the look up ta ble. The default behavior, when this bit is set to ze ro, is for the XOR operation to happen after the look up table. This is correct for 15, 16, 24 bit pe 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.

This register is set to 00h after reset.

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 curso 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 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 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 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.

12.11.5 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 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 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 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.

Bit 0 *Reserved.* This bit should be written as zero.

This register is undefined after reset.

Programming notes

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 located 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 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 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.

12.12 GRAPHICS CLOCK REGISTERS

The GCLK is used to the Graphics Engine operations

12.12.1 GCLK CONTROL REGISTER 0 GCLK00 INDEX 40

Bit 7 This is unused.

Bits 6-3 This the 4-bit M (divisor) value of the Graphics clock synthesizer.

Bits 2-0 These are bits 7-5 of the 8 bit N (mulitiplier) of the Graphics clock synthesizer.

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 GCLK01 INDEX 41

Bit 7 This is the bit 0 of the 3-bit P (exponent) value of the Graphics clock synthesizer.

Bits 6-2 These are bits 4-0 of the 8-bit N (multiplier) value of the Graphics clock synthesizer.

Bits 1-0 These are bits 2-1 of the 3-bit P (exponent) value of the Graphics clock synthesizer.

This register defaults to 0xEC at reset. This value when combined with the default value of the othe half of this pair results in a graphics clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

13 VIDEO CONTROLLER

13.1 INTRODUCTIO

The iDragon SCX501 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.

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.

13.1.3 The Video Output Port

The Video Output Port is a high performance PAL/NTSC digital encoder. It converts a 4:2:2 digital video stream into a standard analog baseband PAL/NTSC signal and into RGB analog components.

The Video Output Port can handle interlaced mode (with 525/625 line standards) and non-interlaced mode. Encoding supported includes NTSC-M, PAL-B, D, G, H, I, N, M, plus NTSC-4.43 encoding. It can perform Closed-Caption, or CGMS encoding.

13.2 VIDEO INPUT PORT (VIP) 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.

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 ac cesses 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 arbitra tion insures maximum bandwidth for critical tasks. The time-out period is programmable to accom modate devices with long access times.

13.3 DIGITAL VIDEO INPUT FORMATS

The video input port can be programmed to de code one of several video formats. The following sections discuss this functionality in more detail.

Existing video input formats are shown in Figure 13.1.:

Table 13.1. Video Encoding Formats

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 data 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 inde pendently 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 iDragon SCX501 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 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, the 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.

Sliced VBI data cannot be transferred as ancillary data. It is intended that sliced VBI data be transferred from VIP compliant devices using the Host Port since only this method is available during MPEG playback.

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 SIP CRTC supports both color and chroma based video mixing.

Table 13.2. Video Input Module Address Space

AD[31:28]	[27:24]	[23:20]	[19:18]	[17:15]	Description
0000	GBASE	0110	00	100	Video Input Port Registers - PIXCLK domain

13.5 VIDEO INPUT MODULE ADDRESS SPACE

VIP Target devices are memory mapped into Graphics Register Space in the 4 Mbyte 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 13.2..

13.6 VIP VIDEO 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 (FB1_Adr) Index 00h

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.

Bits 31-22 *Reserved.* This Read-Only field is reserved. When read it returns '0's

Bits 21-0 **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 (Vin_CFG) Index 04h

The top byte of Vin_CFG is reserved for enabling and disabling interrupts.

Bit 32 *Reserved.*

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 pre serves the existing value.

Writing a '1' to both the enable and reset enable field at the same time produces undefined results.

Bit 31 **Reset Buffer Full IRQ Enable, Rst_BFIEn.** This is a Write-only

Bit 31	Buffer Full IRQ Enable
0	Preserve Buffer Full IRQ enable
1	Reset Buffer Full IRQ enable

Bit 30 **Reset Field IRQ Enable, Rst_FIEn.** This is a Write-only

Bit 30	Field IRQ Enable
0	Preserve Field IRQ enable
1	Reset Field IRQ enable

Bit 29 **Reset Vertical Blank IRQ Enable, Rst_VBIEn.** This is a Write-only

Bit 29	Vertical Blank IRQ Enable
0	Preserve Vertical Blank IRQ enable
1	Reset Vertical Blank IRQ enable

Bit 28 **Reset Buffer Overflow IRQ Enable, Rst_BOEn.** This is a Write-only

Bit 28	Buffer Overflow IRQ Enable
0	Preserve Buffer Overflow IRQ enable
1	Reset Buffer Overflow IRQ enable

Bit 27 **Buffer Full IRQ Enable, BF_IEn.**)

Bit 27	Buffer Full IRQ
0	Preserve existing BF_IEn value
1	IRQ is generated when either video input buffer goes full

Bit 26 **Field Change IRQ Enable, F_IEn.**)

Bit 26	Field Change IRQ Enable
0	Preserve existing F_IEn value
1	IRQ is generated when the internal Field bit changes

Bit 25 **Vertical Blank IRQ Enable, VB_IEn.**)

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.

Bit 24 **Video Input Buffer Overflow Enable, BO_IEn.**

Bit 24	Video Input Buffer Overflow
0	Preserve existing BO_IEn value
1	IRQ is generated when either video input buffer overflows (see vin_stat bit 24.)

Bits 23-22 **VCLK source.**

Bit 23	Bit 22	VCLK source
0	1	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*

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.

Bit 21 **Start Buffer, Start_BF**
Controls which video input buffer will be filled first

Bit 21	Video Input Buffer Fill
0	video input buffer 0 filled first
1	video input buffer 1 filled first

Bit 20 **Decimator Enable, Dec_En**

Bit 20	Decimator Enable
0	no decimation of input pixels
1	Enable 2:1 video decimator

Bit 19 **Auto Update Auto_Up**
This bit enables automatic updating of the dis played video buffer.

Bit 19	Auto Update
0	Buffer must be updated by the driver.
1	Buffer automatically switched to the most recently completed display buffer.

Bits 18-16 **Video Input Format, VI_Form**
This field controls how the video stream will be decoded.

Bit 18	Bit 17	Bits 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*

In VIP 1.0 Video Mode the timing information is re-covered 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 exter nal source. Video timing generation is enabled fo formats 0-3 and disabled for modes 4-7.

Bit 15 *Reserved.* This Read-Only bit is reserved. When read it returns undefined data.

Bit 14 **Byte Swap.**

Bit 12	Byte Swap
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.

Bits 13-12 **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 **Frame Drop Control.**.

Bit 11	Bits 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

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.

Bits 9-7 **Field Capture Control.** .

Bit 9	Bit 8	Bits 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

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 res olution but can cause motion artifacts where there are areas of movement. When capturing inter laced video in double buffer mode, the buffer is switched at the end of each enabled field.

Bit 6 **Double Buffer Enable.** Double Buffer Ena ble allows the amount of Frame Buffer memory to be reduced when capturing at less than full video rates. When single buffering is selected, all cap tured fields are written to the Frame Buffer using the buffer selected by the start buffer field.

Bit 5	Double Buffer Enable
0	single buffer
1	double buffer

Bit 4 **Enable Video Capture.**

Enable Video Capture starts or stops video cap ture operation. Video capture starts at the first en abled 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.

Bit 4	Video Capture Enable
0	Video capture will end after current frame
1	Video capture will begin at start of next frame or task

Bits 3-1 **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 **Input Port Enable.** The Input Port Enable al lows 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.

Bit 0	Input Port Enable
0	Video input port disabled, counters/state machines initialized, capture of video stopped.
1	Video input port enabled

13.6.3 Video Input Port Status Register (vin_stat) Index 08h

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. Read only bits are unaffected by write cycles. Reserved bits are undefined and must be masked off before making comparisons.

Bits 31-12 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bit 11 **Channel 2 Interrupt Pending Copy, Ch2Int.** 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 **Channel 1 Interrupt Pending Copy, Ch1Int.** 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 present.** This read-only bit reflects the prescence of the VCLK signal.

Bit 9	VCLK Present
0	VCLK is not present
1	VCLK is present

Bit 8 **Vblank.** This read-only bit reflects the value of the internal vertical blank signal.

Bit 8	Vblank
0	Active video region
1	Vertical blanking region

Bit 7 **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 **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 **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 **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 4	Odd/Even Field
0	Field 1 (Top field) is currently being processed.
1	Field 2 (Bottom field) is currently being processed.

Bit 3 **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.

Bit 2 **Buffer Overrun IRQ.** This bit is set and latched when either of the buffers receives a write from the pixel packer and the corresponding buffe 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 **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 **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].

13.6.4 Video Input Buffer Addr 0 (vin_ad0) Index 0Ch

Lines of video always start at a quad word boundary in the Frame Buffer. When the display windo 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.

Bits 31-22 *Reserved.* This Read-Only field is reserved. When read it returns '0's

Bits 21-0 **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'.

13.6.5 Video Input Buffer Addr 1 (vin_ad1) Index 10h

The LS 3 bits of this register are hardwired to zero to force QWORD alignment

Bits 31-22 *Reserved.* This Read-Only field is reserved. When read it returns '0's

Bits 21-0 **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 Dest Pitch (vin_dp) Index 14h

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.

Bits 31-14 *Reserved.* This Read-Only field is reserved. When read it returns '0's

Bits 13-0 **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'.

13.6.7 External Timing Generator 1 (vtg_ext1) Index 28h

Bit 31 **VTG enable.** .

Bit 31	VTG Enable
0	Video timing is reset to the start of field 1
1	Video timing generator is enabled

Bit 30 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bit 29 **Output enable for the B/T# video timing signal , bt_oe.** Set to '1' when the Video Controller is generating the system video timing signals..

Bit 29	B/T# Direction
0	B/T# is an input
1	B/T# is an output

Bit 28 **Output enable for the HSYNC- video timing signal, hsync_oe.** Set to '1' when the Video Controller is generating the system video timing signals..

Bit 28	HSYNC# Direction
0	HSYNC# is an input
1	HSYNC# is an output

Bit 27 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bit 26 **B/T# polarity, bt_pol.** 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..

Bit 26	B/T# Polarity
0	B/T# is low for field 1
1	B/T# is high for field 1

Bit 25 **HSYNC# polarity, hsync_pol.** 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..

Bit 25	HSYNC# Polarity
0	HSYNC# is low true
1	HSYNC# is high true

Bit 24 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 23-21 **Genlock Mode.** Defines the method for genlocking to an external source..

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
		1xx	*Reserved*

13.6.8 External Timing Generator 2 (vtg_ext2) Index 28h (Continued)

Bit 20-18 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 17-12 **Leading edge of HSYNC# in pixels, HS_St.** 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.

Bits 11-9 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 8-1 **Trailing edge of HSYNC# in pixels, HS_End.** 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 **HSYNC# Odd compensation, HS_Odd.** 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.

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.

External Timing Generator 2 (vtg_ext2)
Index 2Ch

Bits 31-21 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 20-11 **B/T# delay for field 1 in pixels, BT_Dly1.** 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 **B/T# delay for field 2 in pixels, BT_Dly2.** This field determines the location relative to the horizontal counter that B/T# switches to indicate field 2 of an interlaced frame.

Bit 0 **B/T## Odd compensation, BT_Odd.** 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) Index 30h

Bits 31-22 **Horizontal start of active video in pixels, H_Start.** 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 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 20-11 **Horizontal end of active video in pix els, H_End.**

Bit 10 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 9-0 **Total number of horizontal pixels, H_Total.** This field contains the total number of pixels per line.

13.6.10 Video Timing Generator (vtg_vt) Index 34h

V_Start specifies the first line of active video in each field.

V_End specifies the last line of active video in each field.

Bits 31-22 **Vertical Field Start, V_ 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 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 20-11 **Vertical Field End, V_ End.** Line number of the last line of active video in each field.

Bit 10 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 9-0 **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 OUTPUT PORT

13.7.1 Functional Description

The Video Output Port can operate either in master mode, where it supplies all sync signals, or in 6 slave modes, where it locks onto incoming sync signals.

13.7.2 Data Input Format

The digital input from the Video Input Port Passthrough mode, or generated from the video signam anticipates a time-multiplexed ITU-R656 /D1-type [Cb, Y, Cr, Y] 8-bit stream. (Note that "ITU-R" was formerly known as "CCIR"). Input samples are latched in on the rising edge (by default) of the clock signal CKREF, whose nominal frequency is 27MHz. Figure 1 illustrates the expected data input format. Alternatively, a 54-Mbit/s stream can be fed to the Video Output Port, refer to Section IV.17 ("dual encoding") for details.

The Video Output Port is able to encode interlaced and non-interlaced video. One bit is sufficient to automatically direct the Video Output Port to process non-interlaced video. Update is performed internally on the first frame sync active edge following the programing of this bit. The non-interlaced mode is a 624/2 = 312 line mode or a 524/2 = 262 line mode, where all fields are identical.

An `autotest' mode is available by setting 3 bits (sync[2:0]) within the configurations register0. In this mode, a color bar pattern is produced, independently from video input, in the appropriate standard.

As this mode sets the Video Output Port in master mode, VSYNC/ODDEV and HSYNC pins are then in output mode.

13.7.3 Video Timing

The Video Output Port outputs interlaced or non-interlaced video in PAL-B, D, G, H, I, PAL-N, PAL-M or NTSC-M standards and `NTSC- 4.43' is also possible.

The 4-frame (for PAL) or 2 frame (for NTSC) burst sequences are internally generated, subcarrie generation being performed numerically with CK REF as reference. Rise and fall times of synchronization tips and burst envelope are internally con trolled according to the relevant ITU-R and SMPTE recommendations.

It is possible to allow encoding of incoming YCrCb data on those lines of the VBI that do not bear line sync pulses or pre/post-equalisation pulses. This mode of operation is refered to as "partial blank ing" and is the default set-up. It allows to keep in the encoded waveform any VBI data present in digitized form in the incoming YCrCb stream (e.g. WSS data, VPS, supplementary Closed-Captions line or StarSight data, etc.). Alternatively, the com plete VBI may be blanked (no incoming YCrCb data encoded on these lines, "full blanking").

The complete VBI comprises of the following lines:

for 525/60 systems (SMPTE line numbering convention): lines 1 to 19 and second half of line 263 to line 282.

for 625/50 systems (CCIR line numbering conven tion): second half of line 623 to line 22 and lines 311 to 335.

The `partial' VBI consists of :

for 525/60 systems (SMPTE line numbering convention): lines 1 to 9 and second half of line 263 to line 272.

for 625/50 systems (CCIR line numbering conven tion): second half of line 623 to line 5 and lines 311 to 318.

Full or partial blanking is controlled by configura tion bit 'blkli in configuration register1'.

Note that:

line 282 in 525/60/SMPTE systems is either fully blanked or fully active.

line 23 in 625/60/CCIR systems is always fully active.

In an ITU-R656-compliant digital TV line, the ac tive portion of the digital line is the portion included between the SAV (Start of Active Video) and EAV (End of Active Video) words. However, this digital active line starts somewhat earlier and may end

slightly later than the active line usually defined by analog standards. The Video Output Port allows two approaches :

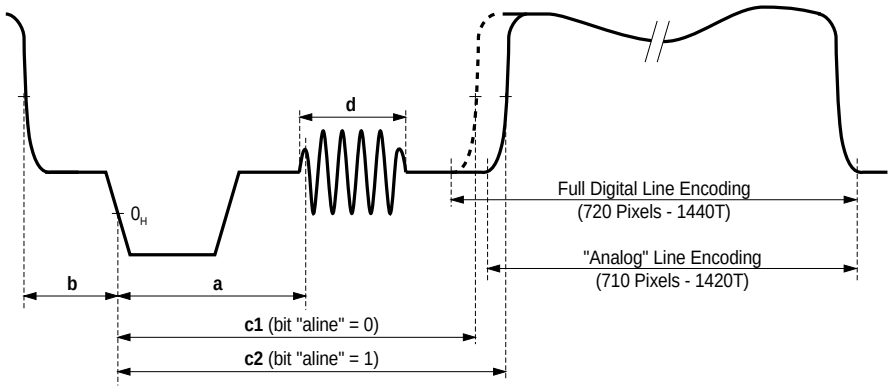
It is possible to encode the full digital line (720 pixels / 1440 clock cycles). In this case, the output waveform will reflect the full YCrCb stream included between SAV and EAV.

Alternatively, it is possible to drop some YCrCb samples at the extremities of the digital line so that the encoded analog line fits within the 'analog' ITU-R/SMPTE specifications.

Selection between these two modes of operation is performed with bit 'aline' in configuration register 4.

In all cases, the transitions between horizontal blanking and active video are shaped to avoid too steep edges within the active video. Figure 13.1. gives timings concerning the horizontal blanking interval and the active video interval.

Figure 13.1. Horizontal Blanking Interval and Active Timings



	NTSC-M	PAL-BDGHI	PAL-N	PAL-M
a	5.38µs (even lines) 5.52µs (odd lines)	5.54µs (A-type) 5.66µs (B-type)	5.54µs (A-type) 5.66µs (B-type)	5.73µs (A-type) 5.87µs (B-type)
	These are typical values. Actual values will depend on the static offset programmed for subcarrier generation.			
b	1.56µs	1.28µs	1.28µs	1.28µs
c1	8.8µs	9.3µs	9.3µs	9.3µs
c2	9.3µs	10.1µs	10.1µs	10.1µs
d	9 Cycles of 3.58MHz	10 Cycles of 4.43MHz	9 Cycles of 3.58MHz	9 Cycles of 3.58MHz

13.7.4 Reset Procedure

After a hardware reset the Video Outout Port is set in HSYNC+ODDEV (line-locked) slave mode, for NTSC-M, interlaced ITU-R601 encoding. Closed-captioning is disabled.

Then the configuration can be customized by writing into the appropriate registers. A few registers are never reset, their contents is unknown until the first loading (refer to the Register Contents and Description).

It is also possible to perform a software reset by setting bit'softreset' in Reg 6. The Video Output Port's response in this case is similar to its response after a hardware reset, except that Configuration Registers (Reg 0 to 6) and a few other registers (see description of bit `softreset') are not altered .

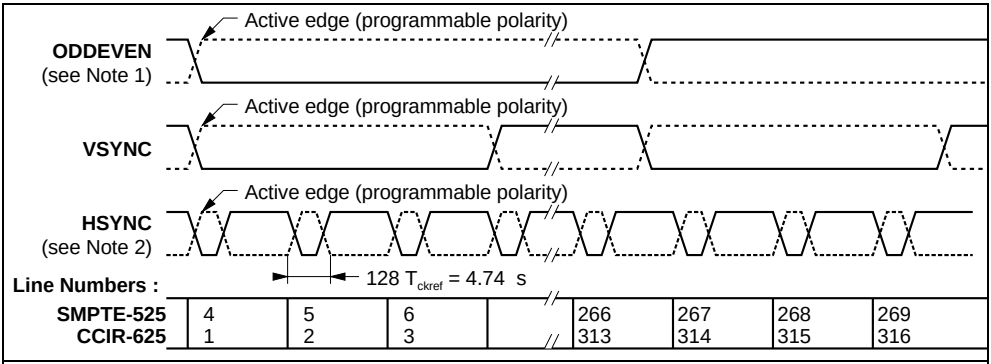
13.7.5 Master Mode

In this mode, the Video Output Port supplies HSYNC and ODDEV sync signals (with independently programmable polarities) to drive other blocks. Refer to Figure 13.2. and Figure 13.3. for timings and waveforms

The Video Output Port starts encoding and counting clock cycles as soon as the master mode has been loaded into the control register (Reg.0).

Configuration bits "Syncout_ad[1:0]" (Reg4) allo to shift the relative position of the sync signals by up to 3 clock cycles to cope with any YCrCb phas ing.

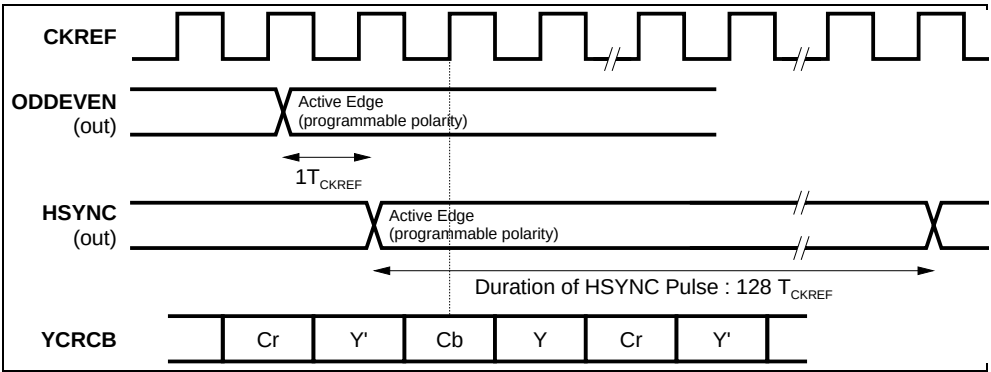
Figure 13.2. ODDEVEN, VSYNC and HSYNC Waveforms



Note 1.; When ODDEVEn is a sync input, only one edge (the “active edge”) of the incoming ODDEVEN is taken into account for synchronization. The “non-active” edge (2nd edge in this Figure) is not critical and its position may differ by H/2 from the location shown.

Note 2.; The HSYNC pulse width indicated is valid when the iDragon SCX501 supplies HSYNC. In those slave modes where it receives HSYNC, only the edge defined as active is relevant, and the width of the HSYNC pulse it receives is not critical

Figure 13.3. Master Mode Sync Signals



Note1. This Figure is valid for bits “syncout_ad[1:0]” = default

13.7.6 Slave Modes

Six slave modes are available : ODDEV+HSYNC based (line-based sync), VSYNC+HSYNC based (another type of line-based sync), ODDEV-only based (frame-based sync), VSYNC-only based (another type of frame-based sync), or sync-in-data based (line locked or frame locked).

ODDEV refers to an odd/even (also known as not-top/bottom) field flag, HSYNC is a line sync signal, VSYNC is a vertical sync signal. Their waveforms are depicted in Figure 13.3.. The polarities of HSYNC and VSYNC/ODDEV are independently programmable in all slave modes.

13.7.7 HSYNC+ODDEV Based Line Synchronization

Synchronization is performed on a line-by-line basis by locking onto incoming ODDEV and HSYNC signals. Refer to Figure 11 for waveforms and timings. The polarities of the active edges of HSYNC and ODDEV are programmable and independent.

The first active edge of ODDEV initializes the internal line counter but encoding of the first line does not start until an HSYNC active edge is detected (at the earliest, HSYNC may transition at the same time as ODDEV). At that point, the internal sample counter is initialized and encoding of the first line starts. Then, encoding of each subsequent line is individually triggered by HSYNC active edges. The phase relationship between HSYNC and the incoming YCrCb data is normally such that the first clock rising edge following the HSYNC active edge samples "Cb" (i.e. a `blue' chroma sample within the YCrCb stream). It is however possible to internally delay the incoming sync signals (HSYNC+ODDEV) by up to 3 clock cycles to cope with different data/sync phasings, using configuration bits "Syncin_ad" (Reg. 4).

The Video Output Port is thus fully slaved to the HSYNC signal, which means that lines may contain more or less samples than typical 525/625 system requirement.

If the digital line is shorter than its nominal value: the sample counter is re-initialized when the 'early' HSYNC arrives and all internal synchronization signals are re-initialized.

If the digital line is longer than its nominal value: the sample counter is stopped when it reaches its nominal end-of-line value and waits for the `late' HSYNC before reinitializing.

The field counter is incremented on each ODDEV transition. The line counter is reset on the HSYN following each active edge of ODDEV.

13.7.8 HSYNC+VSYNC Based Line Synchronization

Synchronization is performed on a line-by-line ba sis by locking onto incoming VSYNC and HSYN signals. The polarities of HSYNC and VSYNC are programmable and independent.The incoming VSYNC signal is immediately transformed into a waveform identical to the odd/even waveform of an ODDEV signal, therefore the behavior of the core is identical to that described above for ODD-EV+HSYNC based synchronization. Again, the phase relationship between HSYNC and the in coming YCrCb data is normally such that the first clock rising edge following the HSYNC active edge samples "Cb" (i.e. a `blue' chroma sample within the YCrCb stream). It is however possible to internally delay the incoming sync signals (HSYNC+VSYNC) by up to 3 clock cycles to cope with different data/sync phasings, using configuration bits "Syncin_ad" (Reg. 4).The field counter is incremented on each active edge of VSYNC.

13.7.9 ODDEV-only Based Frame Synchronization

Synchronization is performed on a frame-by frame basis by locking onto an incoming ODDEV signal. A line sync signal is derived internally and is also output as HSYNC. The phase relationship between ODDEV and the incoming YCrCB data is normally such that the first clock rising edge fol lowing the ODDEV active edge samples "Cb" (i.e. a `blue' chroma sample within the YCrCb stream). It is however possible to internally delay the incoming ODDEV signal by up to 3 clock cycles to cope with different data/sync phasings, using con figuration bits "Syncin_ad" (Reg. 4).

The first active edge of ODDEV triggers genera tion of the analog sync signals and encoding of the incoming video data. Frames being supposed to be of constant duration, the next ODDEV active transition is expected at a precise time after the last ODDEV detected.So, once an active ODDEV edge has been detected, checks that the following ODDEV are present at the expected instants are performed.

Encoding and analog sync generation carry on un less three successive fails of these checks occur.

In that case, three behaviors are possible, accord ing to the configuration programmed (Reg. 1-2):

if 'free-run' is enabled, the Video Output Port carries on outputting the digital line sync HSYNC and generating analog video just as though the expected ODDEV edge had been present. However, it will re-synchronize onto the next ODDEV active edge detected, whatever its location.

if 'free-run' is disabled but bit 'sync_ok' is set in configuration register1, the Video Output Port sets the active portion of the TV line to black level but carries on outputting the analog sync tips (on Ys and CVBS) and the digital line sync signal HSYNC. (When programmed, AGC pulses are also present in the analog sync waveform).

if 'free-run' is disabled and the bit 'sync_ok' is not set, all analog video is at black level and neither analog sync tips nor digital line sync are output.

Note that this mode is a frame-based sync mode, as opposed to a field-based sync mode, that is, only one type of edge (rising or falling, according to bit 'polv'in Reg 0) is of interest to the Video Output Port, the other one is ignored.

13.7.10 VSYNC only Based Frame Synchronization

Synchronization is performed on a frame-by-frame basis by locking onto an incoming VSYNC signal. An auxiliary line sync signal HSYNC must also be fed to the Video Output Port, which uses it to reconstruct from VSYNC and HSYNC information an internal odd/even waveform identical to that of an ODD-EVEN signal. Therefore the behavior of the core is identical to that described above for ODDEVEN only based synchronization (except that nothing is output on HSYNC pin since it is an input port in that mode).

Note that HSYNC is an input but has no other use than allowing the Video Output Port to decide whether an incoming VSYNC pulse flags an odd or an even field. In other words, the Video Output Port does not lock onto HSYNC in this mode since this is NOT a line-locked mode.

The phase relationship between VSYNC and the incoming YCrCb data is normally such that the first clock rising edge following the VSYNC active edge samples "Cb" (i.e. a `blue' chroma sample within the YCrCb stream). It is however possible to internally delay the incoming sync signals (VSYNC+HSYNC) by up to 3 clock cycles to cope with different data/sync phasings, using configuration bits "Syncin_ad" (Reg. 4).

13.7.11 `End-of-frame' Word Based Data-embedded Synchronization

Synchronization is performed by extracting the 1 to-0 transitions of the 'F' flag (end-of-frame) from the `EAV' (End-of-Active Video) sequence embedded within ITU-R656 / D1 compliant digital video streams. Both a frame sync signal and a line sync signal are derived and are made available exter nally as ODDEVEN and HSYNC.

The first successful detection of the `F' flag trig gers generation of the analog sync signals and en coding of the incoming video data. Frames being supposed to be of constant duration, the next EAV word containing the 'F' flag is expected at a precise time after the latest detection.

So, once an active 'F' flag has been detected, checks that the following flags are present within the incoming video stream at the expected times are performed.

Encoding and analog sync generation carry on un less three successive fails of these checks occur.In that case, three behaviors are possible, ac cording to the configuration programmed :

if 'free-run' is enabled, the Video Output Port car ries on generating the digital frame and line syncs (ODD-EVEN and HSYNC) and generating analog video just as though the expected 'F' flag had been present. However, it will re-synchronize onto the next 'F' flag detected within the incoming CCIR656/D1 video stream.

if 'free-run' is disabled but the bit 'sync_ok' is set in the configuration registers, the Video Output Port sets the active portion of the TV line to black level but carries on outputting the analog sync tips (on Ys and CVBS) and the digital frame and line sync signals ODDEVEN and HSYNC. (When pro grammed, AGC pulses are also present in the an alog sync waveform).

if 'free-run' is disabled and the bit 'sync_ok' is not set, all analog video is at black level and neithe analog sync tips nor digital frame/line sync are output.

The SAV and EAV words are Hamming-decoded. After detection of two successive errors, a bit is set in the status register to inform the micro-controller of the poor transmission quality.

13.7.12 `End-of-line' Word Based Data-embedded Synchronization

Synchronization is performed by extracting the 'F' and 'H' flags from the 'SAV' (Start of Active Video)

and 'EAV' (End of Active Video) words embedded within ITU-R656/D1 compliant digital video streams.

A line sync signal and a frame sync signal are derived internally from these flags and are output on the HSYNC and ODDEVEN/VSYNC pins in output mode. These signals are also exploited by the core of the circuit which treats them like it treats incoming ODDEVEN and HSYNC signals in HSYNC+ODDEV based synchronization.

13.7.13 Input Demultiplexer

The incoming 27Mbit/s YCrCb data is demultiplexed into a `blue-difference' chroma information stream, a `red-difference' chroma information stream and a luma information stream. Incoming data bits are treated as blue, red or luma samples according to their relative position with respect to the sync signals in use and to the content of configuration bits "Syncin_ad" (slave modes) or "Syncout_ad" (master mode).

The ITU-R601 recommendation defines the black luma level as Y = 16dec and the maximum white luma level as Y = 235dec. Similarly it defines 225 quantization levels for the color difference components (Cr, Cb), centered around 128.

Accordingly, incoming YCrCB samples can be saturated in the input multiplexer with the following rules :

for Cr or Cb samples :
Cr, Cb > 240 $\Rightarrow$ Cb saturated at 240
Cr,Cb < 16 $\Rightarrow$ Cr, Cb saturated at 16

for Y samples :
Y > 235 $\Rightarrow$ Y saturated at 235
Y < 16 $\Rightarrow$ Y saturated at 16

This avoids having to heavily saturate the composite video codes before digital-to-analog conversion in case erroneous or unrealistic YCrCb samples are input to the encoder (there may otherwise be overflow errors in the codes driving the DACs), and therefore avoids generating a distorded output waveform.

However, in some applications, it may be desirable to let 'extreme' YCrCb codes pass through the demultiplexer. This is also possible, provided that bit "maxdyn" is set in configuration register 6. In this case, only codes 00h and FFh are overridden: if such codes are found in the active video samples, they are forced to 01h and FEh.

In any case, the YCrCb codes are not overridden for EAV/SAV decoding

The demultiplexer is also able to handle 54Mbit/s YCrCb streams for dual encoding applications. Refer to Section "Dual Encoding Application - 54Mbit/s YCrCB interface".

13.7.14 Sub-carrier Generation

A Direct Digital Frequency Synthesizer (DDFS using a 24-bit phase accumulator, generates the required color sub-carrier frequency. This oscilla tor feeds a quadrature modulator which modulates the baseband chrominance components. The sub-carrier frequency is obtained from the follow ing equation :

Fsc = (24-bit Increment Word / 2^{24}) x CKREF

Hard-wired Increment Word values are available for each standard (except for `NTSC-4.43') and can be automatically selected.

Alternatively (according to bit `selrst' in Reg. 2.), the frequency can be fully customized by pro gramming other values into a dedicated Increment Word Register (Reg. 10-11-12). This allows for in stance to encode "NTSC-4.43" or "PAL-M-4.43".

This is done with the following procedure :

Program the required increment in Registers 10 to 12

Set bit `selrst' to `1' in Configuration Register 2

Perform a software reset (Reg. 6).

Caution : this sets back all bits from Reg. 7 on wards to their default value, when they can be re set.

Warning : if a standard change occurs after the software reset, the increment value is automatical ly re-initialized with the hardwired or loaded value according to bit selrst.

The reset phase of the color sub-carrier can also be software-controlled (Reg. 13-14).The sub-car rier phase can be periodically reset to its nominal value to compensate for any drift introduced by the finite accuracy of the calculations. Sub-carrie phase adjustment can be performed every line, every eight field, every four field, or every two field (Register 2 bits valrst[1:0]).

13.7.15 Burst Insertion

The color reference burst is inserted so as to always start with a positive zero crossing of the subcarrier sine wave.

The first and last half-cycles have a reduced amplitude so that the burst envelope starts and ends smoothly.

The burst contains 9 or 10 sine cycles of 4.43361875MHz or 3.579545MHz according to the standard programmed in the Control Register (Reg. 0, bits std[1:0]), as follows:

NTSC-M	9 cycles of 3.579542MHz
PAL-BDGHI	10 cycles of 4.43361875MHz
PAL-M	9 cycles of 3.57561149MHz
PAL-N	9 cycles of 3.5820558MHz

It is possible to turn the burst off (no burst insertion) by setting configuration bit `bursten' to 0 (register2).

Notes :

- Two strategies exist for burst insertion: one is to merely gate and shape the subcarrier for burst insertion, the other is more elaborate and is to always start the burst with a positive-going zero crossing. In the first case the phase of the subcarrier when the burst starts is not controlled, with the consequence that some of its first and last cycles are more heavily distorted. The second solution guarantees smooth start and end of burst with a maximum of undistorted burst cycles and can only be beneficial to chroma decoders. This is the solution implemented in the Video Output Port.
- While the first option gave constant burst start time but uncontrolled initial burst phase,, the second solution guarantees start on a positive-going zero crossing with the consequence that two burst start locations are visible over successive lines, according to the line parity. This is normal and explained below.
- In NTSC, the relation between subcarrier frequency and line length creates a 180° subcarrier phase difference (with respect to the horizontal sync) from one line to the next according to the line parity.So if the burst always starts with the same phase (positive-going zero crossing), this means the burst will be inserted at time X or at time $X+T_{NTSC}/2$ after the horizontal sync tip according to the line parity,, where T_{NTSC} is the duration of one cycle of the NTSC burst.
- With PAL, a similar rationale holds, and again there will be two possible burst start locations. The subcarrier phase difference (with respect to the horizontal sync) from one line to the next in that case is either 0 or 180° with the following series: A-A-B-B-A-A-...-etc. where A denotes ‘A-type' bursts and B denotes ‘B-type' bursts, A-type and B-type being 180° out of phase with respect to the horizontal sync. So 2 locations are possible, one for A-type, the other for B-type.
- This assumes a periodic reset of the subcarrier is automatically performed (see bits valrst[1:0] in Reg 2). Otherwise, over several frames,, the start of burst will drift within an interval of one a subcarrier's cycle. THIS IS NORMAL and means the burst is correctly locked to the colors encoded. The equivalent effect with a gated burst approach would be the following: the start location would be fixed but the phase with which the burst starts (with respect to the horizontal sync) would be drifting.

13.7.16 Luminance Encoding

The demultiplexed Y samples are band-limited and interpolated at CKREF clock rate. The resulting luminance signal is properly scaled before insertion of any Closed-captions, CGMS or Teletext data and synchronization pulses.

The interpolation filter compensates for the sin(x)/x attenuation inherent to D/A conversion and greatly simplifies the output stage filter. In addition, the luminance that is added to the chrominance to create the composite CVBS signal can be trap-filtered at 3.58MHz (NTSC) or 4.43MHz (PAL). This allows to cope with application oriented towards low-end TV sets which are subject to cross-color if the digital source has a wide luminance bandwidth (e.g. some DVD sources). Note that the trap filter does not affect the S-VHS luminance output nor the RGB outputs.

A 7.5 IRE pedestal can be programmed if needed with all standards (see Reg1, bit setup). This allows in particular to encode Argentinian and non-Argentinian PAL-N, or Japanese NTSC (NTSC with no set-up).

A programmable delay can be inserted on the luminance path to compensate any chroma/luma delay introduced by off-chip filtering (chroma and luma transitions being coincident at the DAC output with default delay) (Reg3, bits del[2:0]).

13.7.17 Chrominance Encoding

U and V chroma components are computed from demultiplexed Cb, Cr samples. Before modulating the subcarrier, these are band-limited and interpolated at CKREF clock rate. This processing eases the filtering following D/A conversion and allows a more accurate encoding. A set of 4 different filters is available for chroma filtering to fit a wide variety of applications in the different standards and include filters recommended by ITU-R Rec624-4 and SMPTE170-M. The available 3dB bandwidths are 1.1, 1.3, 1.6 or 1.9MHz (Reg1, bits flt[1:0]).

The narrower bandwidths are useful against cross-luminance artefacts, the wider bandwidthsallow to keep higher chroma contents and then an improved image quality.

13.7.18 Composite Video Signal Generation

The composite video signal is created by adding the luminance (after optional trap filtering, Reg 3 bits entrap and trap_pal) and the chrominance components. A saturation function is included in the adder to avoid overflow errors should extreme luminance levels be modulated with highly satu rated colors (this does not correspond to natural colors but may be generated by computers o graphic engines).

A `color killing' function is available (Reg 1, bit coki) whereby the composite signal contains no chrominance, i.e. replicates the trap-filtered luminance. Note that this function does not suppress the chrominance on the S-VHS outputs (neverthe less suppressing the S-VHS chrominance is possible using bit "bkg_c" in Reg 5).

13.7.19 RGB Encoding

After demultiplexing, the Cr and Cb samples feed a 4 times interpolation filter. The resulting base band chroma signal has a 2.45MHz bandwidth and is combined with the filtered luma component to generate R, G, B samples at 27MHz.

13.7.20 Closed Captioning

Closed-captions (or data from an Extended Data Service as defined by the Closed-Captions specification) can be encoded by the circuit. The closed caption data is delivered to the circuit through the writing the data to User registers 39 and 40. Two dedicated pairs of bytes (two bytes per field), each pair preceded by a clock run-in and a start bit can be encoded and inserted on the luminance path on a selected TV line.The Clock Run-In and Start code are generated by the Video Output Port.

Closed-caption data registers are double-buffered so that loading can be performed anytime, even during line 21/284 or any other selected line.

User register 39 (resp. 41) contains the first byte to send (LSB first) after the start bit on the appro priate TV line in field 1 (resp. field 2), and user reg ister 40 (resp. 42) contains the second byte to send. The TV line number where data is to be en coded is programmable (Reg. 37, 38). Lines that may be selected include those used by the StarSight data broadcast system. Closed-caption data has priority over any CGMS or signals programmed for the same line.

The internal Clock Run-In generator is based on a Direct Digital Frequency Synthesizer. The nominal instantaneous data rate is 503.5kbit/s (i.e. 32 times the NTSC line rate). Data LOW corresponds nominally to 0 IRE, data HIGH corresponds to 50 IRE at the DAC outputs.

When closed-captioning is on (bits cc1/cc2 in Reg.1), the CPU should load the relevant registers (reg. 39 and 40, or 41 and 42) once every frame at

most (although there is in fact some margin due to the double-buffering). Two bits are set in the status register in case of attempts to load the closed-caption data registers too frequently, these can be used to regulate loading rate.

The closed caption encoder considers that closed caption data has been loaded and is valid on completion of the write operation into register 40 for field1, into register 42 for field 2. If closed caption encoding has been enabled and no new data bytes have been written into the closed caption data registers when the closed caption window starts on the appropriate TV line, then the circuit outputs two US-ASCII NULL characters with odd parity after the start bit.

13.7.21 CGMS Encoding

CGMS (Copy Generation Management System - also known as VBID and described by standard CPX-1204 of EIAJ) data can be encoded by the circuit. Three bytes (20 significant bits) are to be written by the user into User Registers 31, 32 and 33. Two reference bits ('1' then '0') are encoded first, followed by 20 bits of CGMS data (including a Cyclic Redundancy Check sequence, not computed by the device and supplied to it as part of the 20 data bits). The reference bits are generated locally by the Video Output Port.

When CGMS encoding is enabled, the CGMS (see bit encgms in Reg 3) waveform is continously present once in each field, on lines 20 and 283 (SMPTE-525 line numbering). The CGMS data register is double-buffered, which means that it can be loaded anytime (even during line 20/283) without any risk of corrupting CGMS data that could be in the process of being encoded.The CGMS encoder considers that new CGMS data has been loaded and is valid on completion of the write operation into register 33

13.7.22 Line Skip / Line Insert Capability

This patented feature of the Video Output Port offers the possibility to cut the cost of the application by suppressing the need for a VCXO.

Ideally, the master clock used on the application board and fed to the MPEG decoding IC would have exactly same frequency as the clock that was used when the MPEG data was encoded. Obviously this is not realistic; up to now a solution commonly chosen is to dynamically adjust the clock on the board as close to the `ideal' clock as possible with the help of time stamps embedded within the MPEG stream. Such a kind of tracking often involves the use of a VCXO : when the MPEG data buffer fills up to more than some threshold the clock frequency is increased, when it empties down to some other threshold the clock frequency is lowered.

The Video Output Port offers an alternative, cost saving solution: by programming the two bits jump and dec_ninc in configuration Reg6, the Video Output Port is able to reduce or increase the length of some frames in a way that will not introduce visible artefacts (even if comb-filtering is used). These bits should be set according to the level of the MPEG data buffer. Refer to Section VI.2 Register 6, Register 9 and Registers 21-22-23 for complete bit description.

Operation with the Video Output Port as sync master is as follows :

If the MPEG data buffers fills up too much: set bit "jump" to '1' and bit "dec_ninc" to '1'.The Video Output Port will reduce the length of the current frame (Bit "jump" will then automatically be reset to '0').

If the MPEG data buffers empties too much: set bit "jump" to '1' and bit "dec_ninc" to '0'.The Video Output Port will increase the length of the current frame (Bit "jump" will then automatically be reset to '0').

These operations can be repeated until the MPEG data buffer is inside its fixed limits.

It is also possible to use the line skip/repeat capability in non-interlaced mode.

This functionality of the Video Output Port is also available in slave mode, in this case the sync signals supplied to the Video Output Port must be in accordance with the modified frame lengths programmed.

13.7.23 CVBS, S-VHS and RGB Analog Outputs

Four out of six video signals (composite CVBS, SVHS (Y/C) and RGB) can be directed to 4 analog output pins through 9-bit D/A converters operating at the reference clock frequency.

The available combinations (see the section 11.9.35 bit 3) are:

S-VHS (Y/C) + CVBS + CVBS1 when bit 3 is set to '1'
or : R, G, B + CVBS1 when bit3 is set to 0.

A single external analog power supply pair is used for all DACs, but two independent pairs of current

and voltage references are needed. Each current reference pin is normally connected externally to a resistor tied to the analogue ground, whilst each voltage reference pin is normally connected to a capacitance tied to the analogue ground.

The internal current sources are independent from the positive supply, thanks to a bandgap, and the consumption of the DACs is constant whatever the codes converted.

Any unused DAC may be independently disabled by software, in which case its output is at 'neutral' level (blanking for luma and composite outputs, no color for chroma output, black for RGB outputs). For applications where a single CVBS output is required, the RGB/CVBS+S-VHS Triple DAC should be disabled and Pins $I_{REF(RGB)}$, VR_RGB tied to analog power supply.

13.8 VIDEO OUTPUT PORT REGISTE ACCESS

The Video Output Port Registers are memory mapped to the 4MBytes which are reserved fo Graphics and Video. The Video Output Port Ad dress Map is described below.

As GBASE is usually set to 1000 (Default value), the base address to access the Video Output Port Space is 084CA000h. Note; this is only applicable if GBASE is 1000.

Bits 31-28, These bits should be set to '0'.

Bits 27-24, **GBase,** These bits are determined by the GBase Register in section 11.9.8 .

Bits 23-12, **Video Output Port Base Register Address,** These bits must be set to CAh.

Bits 11-0, **Video Output Port Register Select,** The location and access to these registers is given in Table 13.3. .

Bits 31-28	Bits 27-24	Bits 23-20	Bits 19-12	Bits 11-0
0000	GBASE	0100	1100 1010	Register Address

13.9 VIDEO OUTPUT PORT REGISTERS

Table 13.3. Video Output Port Register/Bit Mapping

Register		Addr.	MSB							LSB
Configuration0	R/W	00h	std1	std0	sync2	sync1	sync0	polh	polv	freerun
Configuration1	R/W	01h	blki	flti	flt0	sync_ok	coli	setup	cc2	cc1
Configuration2	R/W	02h	nintrl	enrst	bursten	x	selrst	rstossc	valrst1	x
Configuration3	R/W	03h	entrap	trap_pal	encgms	noosd	del2	del1	del0	x
Configuration4	R/W	04h	syncin_ad1	syncin_ad0	synoutc_ad1	synoutc_ad0	aline	txdl2	txdl1	txdl0
Configuration5	R/W	05h	Reserved	bkcvbs1	reserved	reserved	bk_ys	bk_c	bk_cvbs	dacinv
Configuration6	R/W	06h	softreset	jump	dec_ninc	free_jump	x	x	chgi2c	maxdyn
Reserved	x	07h	x	x	x	x	x	x	x	x
Reserved	x	07h	x	x	x	x	x	x	x	x
Status	R	09h	hok	atfr	b2_free	b1_free	fieldct2	fieldct1	fieldct0	jumping
Increment_dfs	R/W	10h	d23	d22	d21	d20	d19	d18	d17	d16
Increment_dfs	R/W	11h	d15	d14	d13	d12	d11	d10	d9	d8
Increment_dfs	R/W	12h	d7	d6	d5	d4	d3	d2	d1	d0
Phase_dfs	R/W	13h	-	-	-	-	-	-	o23	o22
Phase_dfs	R/W	14h	o21	o20	o19	o18	o17	o16	o15	o14
Reserved	x	15h	x	x	x	x	x	x	x	x
Reserved	x	16h	x	x	x	x	x	x	x	x
id1	R	17h	0	1	1	1	0	1	1	1
id2	R	18h	0	0	0	0	0	0	0	1
Reserved	x	19h	x	x	x	x	x	x	x	x
Reserved	x	20h	x	x	x	x	x	x	x	x
line_reg	R/W	21h	ltarg8	ltarg7	ltarg6	ltarg5	5ltarg4	ltarg3	ltarg2	ltarg1
line_reg	R/W	22h	ltarg0	lref8	lref7	lref6	lref5	lref4	lref3	lref2
line_reg	R/W	23h	lref1	lref0	-	-	-	-	-	-
cgms_bit_1-4	R/W	31h	-	-	-	-	bit1	bit2	bit3	bit4
cgms_bit_5-12	R/W	32h	bit5	bit6	bit7	bit8	bit9	bit10	bit11	bit12
cgms_bit_13-20	R/W	33h	bit13	bit14	bit15	bit16	bit17	bit18	bit19	bit20
Reserved	x	34h	x	x	x	x	x	x	x	x
...	x	...								
Reserved	x	38h	x	x	x	x	x	x	x	x
c.c.c.F1	R/W	39h	opc11	c117	c116	c115	c114	c113	c112	c111
c.c.c.F1	R/W	40h	opc12	c127	c126	c125	c124	c123	c122	c121
c.c.c.F2	R/W	41h	opc21	c217	c216	c215	c214	c213	c212	c211
c.c.c.F2	R/W	42h	opc22	c227	c226	c225	c224	c223	c222	c221
cclif1	R/W	43h	x	x	x	l1_4	l1_3	l1_2	l1_1	l1_0
cclif2	R/W	44h	x	x	x	l2_4	l2_3	l1_1	l1_1	l1_0
Reserved	x	45h	x	x	x	x	x	x	x	x
...	x	...								
Reserved	x	63h	x	x	x	x	x	x	x	x

13.10 VIDEO OUTPUT PORT REGISTER CONTENTS AND DESCRIPTION

13.10.1 REGISTER_0 - Configuration 0

Bit 7-6 **Standard Selected, std[1-0].**

std1	std0	Standard Selected
0	0	PAL BDGHI
0	1	PAL N (see bit set-up)
1	0	NTSC M
1	1	PAL M

Note 1 : Standard on hardware reset is NTSC; any standard modification selects automatically the right parameters for correct subcarrier generation.

Bits 5-3 **Configuration, sync[2-0].**

(refer to Functional Description, Sections IV.4 and IV.5)

sync2	sync1	sync0	Configuration
0	0	0	ODDEVEN based SLAVE mode (frame locked)
0	0	1	F only based SLAVE mode (frame locked)
0	1	0	ODDEV+HSYNC based SLAVE mode (line locked)
0	1	1	'F'+'H' based SLAVE mode (line locked)
1	0	0	VSYNC-only based SLAVE mode (frame locked) (see Note)
1	0	1	VSYNC+HSYNC based SLAVE mode (line locked)
1	1	0	MASTER mode
1	1	1	AUTOTEST mode (color bar pattern)

Caution: In VSYNC-only based slave mode (sync[2:0]="100"),HSYNC is nevertheless needed as an input.

Bit 2 **Synchro H, polh.** Active edge of HSYNC selection (when input) or polarity of HSYNC (when output).

Bit 2	Synchro H
0	HSYNC is a negative pulse (128 T_{CKREF} wide) or falling edge is active
1	HSYNC is a positive pulse (128 T_{CKREF} wide) or rising edge is active

in 525/60 system: before line 20(SMPTE) or before line 283(SMPT

Bit 1 **Synchro V, polv.** Active edge of ODDEVEN/ VSYNC selection (when input) or polarity of ODD EV (when output) - See Note 2.

Bit 1	Synchro V
0	Falling edge of ODDEVEN flags start of field1 (odd field) or VSYNC is active low
1	Rising edge of ODDEVEN flags start of field1 (odd field) or VSYNC is active high

Note 2: In master mode: polarity of ODDEVE output.
In slave by F (from EAV) : polv = 0 (cf D1 encod ing) and ODDEVEN polarity is the image of F ex tracted from EAV words.

Bit 0 **Freerun, freerun.** .

Bit 0	Freerun
0	Disabled
1	Enabled

Caution: This bit is taken into account in ODDEV only, VSYNC-only or 'F' based slave modes and is irrelevant to other synchronization modes.

The reset state of this register is 92h.

13.10.2 REGISTER_1 - Configuration 1

Bit 7 **Vertical Blanking Interval selection for active video, blkli.** Blanking lines area

Bit 7	Vertical Blanking Interval
0	('partial blanking') Only following lines inside Vertical Interval are blanked NTSC-M: lines [1..9], [263(half)..272] (525-SMPTE) PAL-M: lines [523..6],, [260(half)..269] (525-CCIR) other PAL: lines [623(half)..5], [311..318] (625-CCIR) This mode allows preservation of VBI data embedded within incoming YCrCb, e.g. Wide Screen signalling (full line 23), Video Programing Service (line16), etc.).
1	('full blanking') All lines inside VBI are blanked NTSC-M: lines [1..19], [263(half)..282] (525-SMPTE) PAL-M: lines [523..16], [260(half)..279] (525-CCIR) other PAL: lines [623(half)..22], [311..335] (625-CCIR)

Note: blkli must be set to '0' when closed captions and are to be encoded on following lines:

in 625/50 system: before line 23(CCIR) or before line 336(CCIR)

For CGMS and Teletext encodings, blkli value is not taken into account.

Bits 6-5 **U/V Chroma filter bandwidth selection.**

Bit 6 flt1	Bit 5 flt0	3dB Bandwidth	Typica lApplication
0	0	f-3 = 1.1MHz	Low definition NTSC filter
0	1	f-3 = 1.3MHz	Low definition PAL filter
1	0	f-3 = 1.6MHz	High definition NTSC filter (ATSC compliant) & PAL M/N (ITU-R 624.4 compliant)
1	1	f-3 = 1.9MHz	High definition PAL filter: Rec 624 - 4 for PAL BDG/I compliant

Bit 4 **Sync Ok, sync_ok.** Availability of sync signals (analog and digital) in case of input synchronization loss with no free-run active (i.e. freerun='0')

Bit 4	Sync OK Signal
0	No synchro output signals
1	Output synchros available on YS, CVBS and, when applicable, HSYNC (if output port), ODDEVEN (if output port), i.e same behavior as free-run except that video outputs are blanked in the active portion of the line.

Caution: This bit is taken into account in ODDEV-only, VSYNC-only or `F' based slave modes and is irrelevant to other synchronization modes.

Bit 3 **Color killer, coki.**

Bit 3	Color Killer
0	Color ON
1	Color suppressed on CVBS (and CVBS1) output signal (CVBS=YS) but color still present on C and RGB outputs. For color suppression on chroma DAC 'C', see register 5 bit bkg_c.

Bit 2 **Pedestal enable, setup.**

0	Blanking level and black level are identical on all lines (eg: Argentinian PAL-N, Japan NTSC-M, PAL-BDGHI)
1	Black level is 7.5 IRE above blanking level on all lines outside VBI (eg :Paraguayan and Uruguayan PAL-N)

In all standards, gain factor is adjusted to obtain the required levels for chrominance.

Bits 1-0 **Closed caption encoding mode.**

Bit 1 cc2	Bit 2 cc1	Encoding Mode
0	0	No closed caption/extended data encoding
0	1	Closed caption/extended data encoding enabled in field 1 (odd)
1	0	Closed caption/extended data encoding enabled in field 2 (even)
1	1	Closed caption/extended data encoding enabled in both fields

The reset state of this register is 44h.

13.10.3 REGISTER_2 - Configuration 2

Bit 7 **Non-interlaced mode select, nintrl.**

Bit 7	Non Interlace Mode Select
0	Interlaced mode (625/50 or 525/60 system)
1	Non-interlaced mode(2x312/50 or 2x262/60 system)

Note: 'nintrl' update is internally taken into account on beginning of next frame.

Bit 6 **Cyclic update of DDFS phase, enrst.**

Bit 6	DDFS Phase Cycle Update
0	No cyclic subcarrier phase reset
1	Cyclic subcarrier phase reset depending of valrst1 and valrst0 (see below)

Bit 5 **Chrominance burst control, bursten.**

Bit 5	Chrominance Burst Control
0	Burst is turned off on CVBS (and CVBS1), C and RGB outputs are not affected
1	Burst is enabled

Bit 4 **Select set of reset values for Direct Digital Frequency Synthesizer, selrst.**

Bit 4	DDFS Select Reset Value
0	Hardware reset values for phase and increment of subcarrier oscillator
1	Loaded reset values selected (see contents of Registers 10 to 14)

Bit 3 *Reserved.* This bit reads as '0'.

Bit 2 **Software phase reset of DDFS (Direct Digital Frequency Synthesizer), rstosc.**

Bit 2	Software Phase Reset of DDFS
0	inactive
1	a 0-to-1 transition resets the phase of the subcarrier to either the hard-wired default phase value or the value loaded in Register 13-14 (according to bit 'selrst')

Note: Bit `rstosc' is automatically set back to '0' after the oscillator reset has been performed.

Bits 1-0 **Oscillator Value Reset, valrst[1-0].** Note: valrst[1-0] is taken into account only if bit 'enrst' is set.

valrst1	valrst0	Selection
0	0	Automatic reset of the oscillator every line
0	1	Automatic reset of the oscillator every 2nd field
1	0	Automatic reset of the oscillator every 4th field
1	1	Automatic reset of the oscillator every 8th field

Resetting the oscillator means here forcing the value of the accumulator phase to its nominal value to avoid accumulating errors due to the finite number of bits used internally. The value to which the accumulator is reset is either the hard-wired default phase value or the value loaded in Register 13-14 (according to bit `selrst'), to which a 0°, 90°, 180°, or 270° correction is applied according to the field and line on which the reset is performed.

The reset state of this register is 20h.

13.10.4 REGISTER_3 - Configuration 3

Bit 7 **Enable trap filter, entrap.**

0	Trap filter disabled
1	Trap filter enabled

Bit 6 **Trap Filter Select, rap_pal.** Note: 'trap_pal' is taken into account only if bit `entrap' is set.

0	To select the NTSC trap filter (centered around 3.58MHz)
1	To select the PAL trap filter (centered around 4.43MHz)

Bit 5 **CGMS encoding enable, encgms.**

Bit 5	CGMS Encoding
0	Disabled
1	Enabled

Note: When encgms is set to '1' Closed-Captions/ Extended Data Services should not be pro grammed on lines 20 and 283 (525/60, SMPTE line number convention).

Bit 4 **CKREF Actif Edge Select, nosd.** Choice of active edge of 'ckref' (master clock) that samples incoming YCrCb data.

Bit 4	CKREF Actif Edge Select
0	'ckref' rising edge (e.g. data with OSD coming from STi3520M)
1	'ckref' falling edge (e.g. data without OSD coming from STi3520M)

Bits 3-1 **Luma Path Delay, del[2-0].** Delay on luma path with reference to chroma path

Bit 3 del2	Bit 2 del1	Bit 1 del0	Delay on luma path with reference to chroma path
			one pixel corresponds to 1/13.5MHz (74.04ns)
0	0	0	+ 2 pixel delay on luma
0	0	1	+ 1 pixel delay on luma
0	1	0	+ 0 pixel delay on luma
0	1	1	- 1 pixel delay on luma
1	0	0	- 2 pixel delay on luma
Other			+ 0 pixel delay on luma

Bit 0 *Reserved.* This bit reads as '0'.

The reset state of this register is 00h.

13.10.5 REGISTER_4 - Configuration 4

Bit 7-6 **Adjustment of incoming sync signals, syncin_ad[1-0].** Used to insure correct interpreta tion of incoming video samples as Y, Cr or Cb when the encoder is slaved to incoming sync sig nals (incl. 'F/H' flags stripped off ITU-R656/D1 da ta).

Bit 7 syncin_ad1	Bit 6 syncin_ad0	Internal delay undergone by incoming sync
0	0	Nominal
0	1	+1 ckref
1	0	+2 ckref
1	1	+3 ckref

Bits 5-4, **Adjustment of outgoing sync signals, syncout_ad[1-0].** Used to insure correct interpretation of incoming video samples as Y, Cr or Cb when the encoder is master and supplies sync signals..

Bit 5 syncout_ad1	Bit 4 syncout_ad0	Delay added to sync signals before they are output
0	0	Nominal
0	1	+1 ckref
1	0	+2 ckref
1	1	+3 ckref

Bit 3 **Video active line duration control, aline.**

Bit 3	Video Active Line Duration Control
0	Full digital video line encoding (720 pixels - 1440 clock cycles)
1	Active line duration follows ITU-R/SMPTE 'analog' standard requirements

Bits 2-0 *Reserved*. These bits read as '0'.

The reset state of this register is 00h.

13.10.6 REGISTER_5 - Configuration 5

Bit 7 *Reserved*. This bit read as '0'.

Bit 6 **Blanking of DAC CVBS, bkcvbs.**

Bit 6	Blanking of DAC CVBS
0	DAC CVBS in normal operation
1	DAC input code forced to blanking level

Bit 5 **Blanking of DAC 'Ys', bk_ys.**

Bit 5	Blanking of DAC 'Ys'
0	DAC G/Y in normal operation
1	DAC input code forced to blanking level

Bit 4 **Blanking of DAC 'C' , bk_c.**

Bit 4	Blanking DAC 'C'
0	DAC R/C in normal operation
1	DAC input code forced to neutral level [no color]

Bit 3 **Blanking of DAC 'R' , bk_cvbs.**

Bit 3	Blanking DAC 'R'
0	DAC R in normal operation
1	DAC input code forced to black level

Bit 2 **Blanking of DAC 'G' , bk_cvbs.**

Bit 2	Blanking DAC 'G'
0	DAC G in normal operation
1	DAC input code forced to black level

Bit 1 **Blanking of DAC 'B' , bk_cvbs.**

Bit 1	Blanking DAC 'B'
0	DAC B in normal operation
1	DAC input code forced to black level

Bit 0 **DAC Code Invert, dacinv.** 'Inverts' DAC codes to compensate for an inverting output stage in the application.

Bit 0	DAC Code Inverter
0	DAC non inverted inputs
1	DAC inverted inputs

The reset state of this register is 30h.

13.10.7 REGISTER_6 - Configuration6

Bit 7 **Software reset, softreset.**

Bit 7	Software Reset
0	No reset
1	Software reset

Note: Bit 'softreset' is automatically reset after in ternal reset generation.
Software reset is active during 4 CKREF periods. When softreset is activated,, all the device is reset as with hardware reset except for the first six user registers (configurations) and for registers 10 up to 14 (increment and phase of oscillator), 31-33, 34-37 and 39-42.

Bit 6 **jump.**

Bit 5 **dec_ninc.**

Bit 4 **free_jump.**)

Bit 6 jump	Bit 5 dec_ninc	Bit 4 free_jump	Description Line Mode
0	0	0	Normal mode (no line skip/insert capability) CCIR : 313/312 or 263/262 non-interlaced : 312/312 or 262/ 262

Bit 6	Bit 5	Bit 4	Description
0	x	1	Manual mode for line insert ("dec_ninc" = '0') or skip ("dec_ninc"= '1') capability. Both fields of all the frames following the writing are modified according to "lref" and "ltarg" bits of registers 21-22-23 (by default,, "lref" = '0' and "ltarg" = '1' which leads to normal mode above).
1	0	0	Automatic line insert mode. The 2nd field of the frame following the writing is increased. Line insertion is done after line 245 in 525/60 and after line 330 in 625/50. "lref" and "ltarg" bits are ignored.
1	1	0	Automatic line skip mode.The 2nd field of the frame following the writing is decreased. Line suppression is done after line 245 in 525/60 and after line 330 in 625/50. "lref" and "ltarg" bits are ignored.
1	x	1	Not be used

Note: bit "jump" is automatically reset after use.

Bits 3-2 **MColour frequency control via CFC line, cfc[1-0].**

Bit 3	Bit 2	MColor frequency Control
0	0	Disable (update done by loading of registers 10,11 and 12).
0	1	Update of increment for DDFS just after serial loading via CFC
1	0	Update of increment for DDFS on next active edge of HSYNC.
1	1	Update of increment for DDFS just before next colour burst.

Bits 1 *Reserved*. This bit read as '0'.

Bit 0 **Max dynamic magnitude allowed on YCrCb inputs for encoding, maxdyn.**

Bit 0	Max Dynamic Magnitude
0	10h to EBh for Y, 10h to E0h for chrominance (Cr,Cb)
1	01h to FEh for Y, Cr and Cb

Note: In any case, EAV and SAV words are replaced by blanking values before being fed to the luminance and chrominace processings,

The reset state of this register is 00h.

13.10.8 REGISTER_7 and 8 : *Reserved*

13.10.9 REGISTER_9 - Status (read only)

Bit 7 **Hamming OK, hok.** Hamming decoding of frame sync flag embedded within ITU-R656/D1 compliant YCrCb streams.

Bit 7	Hamming OK
0	Consecutive errors
1	A single or no error

Note: signal quality detector is issued from Hamming decoding of EAV,SAV from YCrCb.

Bit 6 **Frame synchronization flag, atfr.** (slave mode only).

Bit 6	Frame Synchronisation Flag
0	Encoder not synchronized
1	Encoder synchronized

Bit 5 **Closed caption registers access condition for field 2, buf2_free.** Closed caption data for field 2 is buffered before being output on the relevant TV line; buf2_free is reset if the buffer is temporarily unavailable. If the microcontroller can guarantee that registers 41 and 42 (cccf2) are never written more than once between two frame reference signals,, then bit 'buf2_free' will always be true (set). Otherwise,, closed caption field2 registers access might be temporarily forbidden by resetting bit 'buf2_free' until the next field2 closed caption line occurs.
Note that this bit is false (reset) when 2 pairs of data bytes are awaiting to be encoded,, and is set back immediately after one of these pairs has been encoded (so at that time,encoding of the last pair of bytes is still pending)

Reset value = '1' (access authorized)

Bit 4 **Closed caption registers access condition for field 1, buf1_free.** Same as buf2_free but concerns field 1.

Reset value = '1' (access authorized)

Bits 3-1 **Digital field identification number, fieldct[2-0].**

Bit 3	Bit 2	Bit 1	Digital Field Identification Number
0	0	0	Indicates field 1
...	...	...	...
1	1	1	Indicates field 8

fieldct[0] also represents the odd/even information (odd='0', even='1') .

Bit 0 **Frame Jumping, jumping.** This bit indicates whether a frame length modification has been programmed at '1' from programming of bit 'jump' to end of frame(s) concerned.

Refer to register 6 and registers 21-22-23.

13.10.10 REGISTERS_10_11_12 - *Increment_dfs : Increment for digital frequency synthesizer*

	MSB							LSB
register_10	d23	d22	d21	d20	d19	d18	d17	d16
register_11	d15	d14	d13	d12	d11	d10	d9	d8
register_12	d7	d6	d5	d4	d3	d2	d1	d0

These registers contain the 24-bit increment used by the DDFS if bit 'selrst' equals '1' to generate the frequency of the subcarrier i.e. the address that is supplied to the sine ROM. It therefore allows to customize the subcarrier frequency synthesized.

1 LSB ~ 1.6 Hz

The procedure to validate usage of these registers instead of the hard-wired values is the following:

Load the registers with the required value

Set bit 'selrst' to 1 (Reg 2)

Perform a software reset (Reg 6)

Note: The values loaded in Reg10-11-12 are taken into account after a software reset, and ONLY IF bit 'selrst' = '1' (Reg. 2).

These registers are never reset and must be explicitly written into to contain sensible information.

On hardware reset (⇒ 'selrst'=0) or on soft reset with selrst='0', the DDFS is initalized with a hard-wired increment,, independent of Registers 10-12. These hardwired values being out of any user register these cannot be read out of the Video Output Port.

These values are:

Value d[23:0]	Frequency Synthesized f	Ref. Clock
21F07Ch for NTSC M	3.5795452MHz	27MHz
2A098Bh for PAL BGHIN	4.43361875MHz	27MHz
21F694h for PAL	3.5820558MHz	27MHz
21E6F0h for PAL M	3.57561149MHz	27MHz

'NTSC-4.43' can be obtained with d[23:0] value like for PAL BGHI but with standard fixed as NTSC.

13.10.11 REGISTERS_13_14 - Phase_dfs *Static phase offset for digital frequency synthesizer (10 bits only)*

	MSB							LSB
register_13	-	-	-	-	-	--	o23	o22
register_14	o21	o20	o19	o18	o17	o16	o15	o14

Under certain circumstances (detailed below), these registers contain the 10 MSBs of the value with which the phase accumulator of the DDFS is initialized after a 0-to-1 transition of bit 'rstosc' (Reg 2), or after a standard change, or when cyclic phase readjustment has been programmed (see bits valrst[1:0] of Reg 2). The 14 remaining LSBs loaded into the accumulator in these cases are all '0's (this allows to define the phase reset value with a 0.35° accuracy).

The procedure to validate usage of these registers instead of the hard-wired values is the following:

Load the registers with the required value

Set bit 'selrst' to 1 (Reg 2)

Perform a software reset (Reg 6)

Notes: Registers 13-14 are never reset and must be explicitly written into to contain sensible information.
If bit 'selrst'=0 (e.g. after a hardware reset) the phase offset used to reinitialize the DDFS is the hard-wired value. The hard-wired values being out of any register,, they cannot be read out of the Video Output Port.

These are:

D9C000h for PAL BDGHI,, N,, M

1FC000h for NTSC-M

13.10.12 REGISTER_15 : *Reserved*

13.10.13 REGISTER_16 : *Reserved*

13.10.14 REGISTER_17 : *Reserved*

13.10.15 REGISTER_18 : *Reserved*

13.10.16 REGISTER_19 : *Reserved*

13.10.17 REGISTER_20 : *Reserved*

13.10.18 REGISTERS_21_22_23 : line_reg = ltarg[8:0] and lref[8:0]

These registers may be used to jump from a reference line (end of that line) to the beginning of a target line of the SAME FIELD.See Table 13.4.

However, not all lines can be skipped or repeated with no problem and, if needed, this functionality has to BE USED WITH CAUTION.

lref[8:0] contains in binary format the reference line from which a jump is required. ltarg[8:0] contains the target line binary number.

Default values: lref[8:0] = 000000000 and ltarg[8:0] = 000000001

Table 13.4. Register_21_22_23

	MSB							**LSB**
register_21	ltarg8	ltarg7	ltarg6	ltarg5	ltarg4	ltarg3	ltarg2	ltarg1
register_22	ltarg0	lref8	lref7	lref6	lref5	lref4	lref3	lref2
register_23	lref1	lref0	-	-	-	-	-	-

13.10.19 REGISTER_31-32-33 - cgms_bit[1:20] : CGMS Data registers *(20 bits only)* See Table 13.5.

Table 13.5. Register_31_32_33

	MSB							LSB
register_31	-	-	-	-	b1	b2	b3	b4
register_32	b5	b6	b7	b8	b9	b10	b11	b12
register_33	b13	b14	b15	b16	b17	b18	b19	b20

Table 13.6. Register_34_35_36_37_38

	MSB							LSB
register_39	opc11	c117	c116	c115	c114	c113	c112	c111

Table 13.7. Register_39-40

	MSB							LSB
register_40	opc12	c127	c126	c125	c124	c123	c122	c121

These registers are never reset.

Word0A ⇒ bit1....bit3

Word0B ⇒ bit4....bit6

Word1 ⇒ bit7....bit10

Word2 ⇒ bit11....bit14

CRC ⇒ bit15...bit20 (not internally computed)

13.10.20 REGISTER_34-35-36-37-38 : *Reserved*

See Table 13.6.

13.10.21 REGISTER_39-40 - cccf1 : Closed caption characters/extended data for field 1

See Table Table 13.7.

First byte to encode in field1 :

Bit 7 **Odd-parity bit of US-ASCII 7-bit character c11[7-1], opc11.**

Second byte to encode in field1:

bit 7 **Odd-parity bit of US-ASCII 7-bit character c12[7:1], opc12.**

Default value: none, but closed caption enabling without loading these registers will issue characte NULL. Registers 39-40 are never reset.

13.10.22 REGISTER_41-42 : cccf2 : Closed caption characters/extended data for field 2

:

Table 13.8. Register_41

	MSB							LSB
register_41	opc21	c217	c216	c215	c214	c213	c212	c211

:

Table 13.9. Register_42

	MSB							LSB
register_42	opc22	c227	c226	c225	c224	c223	c222	c221

Table 13.10. Register 43

	MSB							LSB
register_43	x	x	x	l1_4	l1_3	l1_2	l1_1	l1_0

See Table 13.8.

First byte to encode in field2

Bit 7 **Odd-parity bit of US-ASCII 7-bit character c21[7-1], opc21.**

See Table 13.9.

Second byte to encode in field2:

bit 7 **Odd-parity bit of US-ASCII 7-bit character c12[7:1], opc12.**

Default value : none but closed captions enabling without loading these registers will issue character NULL. Registers 41-42 are never reset.

13.10.23 REGISTER_43 - cclif1 : Closed caption/extended data line insertion for field 1

TV line number where closed caption/extended data is to be encoded in field 1 is programmable through the following register:

:See Table 13.10.

525/60 system : (525-SMPTE line number convention)

Only lines 10 through 22 should be used for closed caption or extended data services (line 1 through 9 contain the vertical sync pulses with equalizing pulses).

l1[4:0] = 00000 no line selected for closed caption encoding

l1[4:0] = 000xx do not use these codes

...

l1[4:0] = i code line (i+6) (SMPTE) selected for encoding

...

l1[4:0] = 11111 line 37 (SMPTE) selected

625/50 system: (625-CCIR/ITU-R line number convention)

Only lines 7 through 23 should be used for closed caption or extended data services.

l1[4:0] = 00000 no line selected for closed caption encoding

...

l1[4:0] = i code line (i+6) (CCIR) selected for encoding (i > 0)

...

l1[4:0] = 11111 line 37 (CCIR) selected

(*) Default value = 01111 line 21 (525/60, 525-SMPTE line number convention).This value also corresponds to line 21 in l625/50 system,(625-CCIR line number convention).

13.11 VIDEO PIPELINE 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 below shows the address format for the Video Pipeline registers.

4Mbyte Region 1

256KbyteSub-region2

31 27	26 24	23 22	21 18	17 16
00001	<GBASE>	01	0010	01

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.

13.11.1 Source Specification registers

13.11.1.1 Video_Src_Base Index = 00h

This register specifies the DRAM linear starting address of the source image, aligned to an 8 byte boundary. 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.

This register is double buffered, the active register is only updated during vertical blanking.

Bits 31-22 *Reserved.*

Bits 21-0 Base linear address of the Video Source Image. Lower address bits 2-0 are reserved and when read return a value of '0'.

This register is cleared to zero after reset

13.11.1.2 Video_Src_Pitch Index = 04h

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.

This register also contains the Y_Vid_Src_dir bit which specifies the Y direction in which the Video Source Image should be read, and the Video_Color_fmt field which defines the the Colo format of the Video Source Image.

Bits 31-14 *Reserved.*

Bits 13-12 **Video Source Image color format**

00 - RGB 555

01 - RGB 565

10 - YUV 422

Bit 11 **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 a 1 bit value that varies.

Bits 10-0 Specifies the amount to add to the cur rent address to get to the address of the corre sponding pixel in the next line. Lower address bits 2-0 are reserved and when read return a value of '0'.

This register is cleared to zero after reset

13.11.1.3 Vid_Src_dim Index = 08h

This register contains the dimensions of the Video Source Image relative the the starting corner.

This register is double buffered. The active regis ter is only updated during vertical sync.

Bits 31-26 *Reserved.*

Bits 25-16 **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 *Reserved.*

Bits 9-0 **dX,** the width in pixels, of a line .

13.11.1.4 CRTC_Burst_length Index = 0Ch

This register contains the CRTC low water mark and burst length.

Bits 31-24 **Delta low water mark, crtc_dlwm.** 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,** 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, delta t.** See the description of crtc_dlwm above.

Bit 10 *Reserved.*

Bits 9-0 **Minimum CRTC burst length, crtc_bl.** 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.

This register is cleared to zero after reset

13.11.1.5 Vid_Burst_Length Index = 10h

This register is the video counterpart of the previ ous register. It specifies the video low water mark and burst length.

Bits 31-24 **Delta low water mark, vid_dlwm.** Together with vid_dt and vid_lwm, this field defines a variable low water mark. When the video windo 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 wa ter mark to remain at a constant value of vid_lwm.

Bits 23-16 **vid_lwm,** the (initial) low water mark fo the video FIFO in bytes. During the video window, if the video FIFO occupancy rises above the lo 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 multi ple of eight (bits 18-16 are not writable and read as zeroes).

Bits 15-11 **vid_dt, delta t.** See the description of vid_dlwm above.

Bit 10 *Reserved.*

Bits 9-0 **Minimum video burst length, vid_bl.** This is the minimum number of bytes that will be sent in one transfer to fill the video FIFO. This val ue 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.

This register is cleared to zero after reset

13.11.2 Destination Specification Registers

13.11.2.1 Vid_Dst_XY
Index = 14h

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.

Bits 31-26 *Reserved.*

Bits 25-16 **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 *Reserved.*

Bits 10-0 **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.

This register is cleared to zero after reset

13.11.2.2 Vid_Dst_dim
Index = 18h

This register contains the dimensions of the video window.

This register is double buffered. The active register is only updated during vertical sync.

Bits 31-26 *Reserved.*

Bits 25-16 **dY,** the height of the video window in screen lines - 1 is entered in this field.

Bits 15-11 *Reserved.*

Bits 10-0 **dX,** the width of the video window in screen pixels.

This register is cleared to zero after reset

13.11.3 Filter Control Registers

13.11.3.1 Horiz_Scale
Index = 20h

This register contains the control for horizontal scaling and decimation.

This register is double buffered. The active regis ter is only updated during vertical sync.

Bits 31-21 *Reserved.*

Bit 20 **Filter Enable 0, F0E.** 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 **Filter Enable 1, F1E.** 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 18-16 *Reserved.*

Bits 15-13 *Reserved.*

Bits 12-0 **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 hpi is 4096.

This register is cleared to zero after reset

13.11.3.2 Vert_Scale
Index = 28h

This register contains the control for vertical scal ing and decimation.

This register is double buffered. The active regis ter is only updated during vertical sync.

Bits 31-21 *Reserved.*

Bit 20 **Vertical Filter Enable 0, VF0E.** 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 **Vertical Filter Enable 1, VF1E.** When in YUV modes, this bit enables the interpolators for the U and V channels. If this bit is 0, pixel replica tion is used. This bit controls the RGB components in RGB modes and should be the same as VF0E.

Bits 18-16 *Reserved.*

Bits 15-13 *Reserved.*

Bits 12-0 **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

This register is cleared to zero after reset

13.11.3.3 Color space converter specification Index = 2Ch

This register contains the control for the Color Space Converter.

Bits 31-1 *Reserved.*

Bit 0 **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.

This register is cleared to zero after reset

13.11.4 Video and Graphics mixing control registers

13.11.4.1 Mix Mode Register Index = 30h

This register contains the Mix_Mode field which defines the method used to mix graphics and video.

Bits 31-2 *Reserved.*

Bits 1-0 **Mix_Mode,** controls the way in which graphics and video are mixed:-

1	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 dis played. Note that the video pixel can be compared either before or after the Colo Space Converter. Note also that the chrom key can be programmed to ignore any or all component values.
1	1	*Reserved*

This register is cleared to zero after reset

13.11.4.2 Color Key Register Index = 34h

This register contains the color key value used in color keying mixing.

Bits 31-24 *Reserved.*

Bits 23-0 **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.

This register is cleared to zero after reset

13.11.4.3 Chroma Key Low Register Index = 38h

This register contains the chroma key low limits, the component ignore bits and the color mode bit used in chroma keying mixing.

Bits 31-28 *Reserved.*

Bit 27 **Chroma key mode.**

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)

Bit 26 **Ignore component 2.** If set, component 2 (V or B) is ignored in the chroma key comparison.

Bit 25 **Ignore component 1.** If set, component 1 (U or G) is ignored in the chroma key comparison.

Bit 24 **Ignore component 0**. If set, component 0 (Y or R) is ignored in the chroma key comparison.

Bits 23-16 **ch2low,** the low limit against which component 2 is compared during chroma key operations.

Bits 15-8 **ch1low,** the low limit against which component 1 is compared during chroma key operations.

Bits 7-0 **ch0low,** the low limit against which component 0 is compared during chroma key operations.

This register is cleared to zero after reset

13.11.4.4 Chroma Key High Register Index = 3Ch

This register contains the chroma key high limits used in chroma keying mixing.

Bits 31-24 *Reserved.*

Bits 23-16 **ch2high,** the high limit against which component 2 is compared during chroma key op erations.

Bits 15-8 **ch1high,** the high limit against which component 1 is compared during chroma key op erations.

Bits 7-0 **ch0high,** the high limit against which component 0 is compared during chroma key op erations.

This register is cleared to zero after reset

The operation of the chroma key can be summa rized 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

   else

      Kn = 0

If((I0||K0) && (I1||K1) && (I2||K2))

   display graphics pixel

else

   display video pixel
```

13.11.4.5 Status Register
Index = 40h

This register contains enable bit for the video scaler.

Bit 31 **vid_enable.** Setting the enable bit turns on the video scaler.

Bits 30-0 *Reserved.*

This register defaults to 00h after reset.

14 LOCAL BUS INTERFACE

14.1 INTRODUCTIO

The Local Bus interface of the PC Industrial provides a 66MHz, low latency bus to external peripheral cards. The Local Bus may operate in asychronous or synchronous modes through the 22 bit address and 16 bit data bus.

The Local Bus interface supports up to two memory devices and four I/O devices. It can support up to 4 MBytes of memory for each of the memory chip selects and from 4 Byte to 1 KByte of I/O space for each of the I/O devices. All the chip select timings are individually programmable. This interface can be accessed only by the CPU.

The memory addresses are predefined for the memory chip selects. The first bank of the memory is intended to be used as a boot device.

The starting address for each IO chip select is programmable at 4 Byte boundary. The access range for each of the chip select is also programmable. The size varies from 4 Bytes to 1 KByte.

14.2 LOCAL BUS REGISTERS

The Local Bus configuration registers can be categorized into 3 groups

1. Address Decode Registers
2. Timing Registers
3. Control Register

All the registers,except the Control Register, are 16 bit wide. These registers are accessible only to the CPU. All the registers are accessed through I/O Port 22 and Port 23. Port 22 is used as the index to the register bank and Port 23 is used as the data port. This way CPU can access only 8 bits of register data at a time, so two accesses are required to completely read or write the 16 bit registers. The lower and upper halves of the 16 bit registers have the same index values. First access after reset with the given index will map to the lowe Byte of the register. The next access with the same index will access the upper Byte. For the 8 bit registers (Control Register Index 1Ch and IO WIDTH Register Index 1Eh) data must be written twice as they are seen as 16 bit registers.

There are two other ways in which these registers can be accessed. These approaches are described in Control Register Section 14.5.

Table 14-1. Local Bus Register Indeces

Name	Index	Function
IOAREG0	10h	Address Decode
IOAREG1	11h	
IOAREG2	12h	
IOAREG3	13h	
IOMREG0	14h	
IOMREG1	15h	
TIMEBANK0	16h	Timing
TIMEBANK1	17h	
TIMEIO0	18h	
TIMEIO1	19h	
TIMEIO2	1Ah	
TIMEIO3	1Bh	
CONTROL	1Ch	Control
IOWIDTH	1Eh	

14.3 LOCAL BUS ADDRESS DECODE REGISTERS

14.3.1 IOAREG, IO Slot Base Address Regs.

This 16 bit register defines the starting address (4 Bytes or 32 bit WORD) and I/O address spaces mapped on the local bus. The base addresses for I/O slots 0 to 3 are specified in register IOAREG0, IOAREG1, IOAREG2 and IOAREG3 respectively. Any accesses that hit onto the address range defined by this register and its mask in the corresponding IOMREG register will prevent the cycle being forwarded onto the PCI bus.

Bits 15-2 Starting Address aligned to 4 I/O locations.

Bits 1-0 *Reserved.*

The 2 memory address spaces (for FLASH memory devices) have a fixed range (FFC00000h-FFFFFFFFh for the Boot device and FEC00000h-FEFFFFFFh for the second device).

14.3.2 IOMREG, IO Slot Mask Registers

The address mask register IOMREG0 and IOMEREG1 define the size of the I/O slots. The 8 bit address mask for each slot will mask the address that we do not want to compare during the address decoding process. The 8 bit mask will filter bit 9:2 of the starting address specified in the corresponding IOAREG allowing a selection of 4, 8, 16, 32, 64, 128, 256, 512 or 1K continuous or non continuous locations.

IOMREG0

Bits 15-8 **Address mask for I/O Slot 1.**

Bits 7-0 **Address mask for I/O Slot 0.**

IOMREG1

Bits 15-8 **Address mask for I/O Slot 3.**

Bits 7-0 **Address mask for I/O Slot 2.**

14.4 LOCAL BUS TIMING REGISTERS

14.4.1 TIMEBANK, Memory Timing Templates

TIMEBANK0 and TIMEBANK1 define the timing template for accessing Flash memory bank 0 and 1 respectively. The timings are programmed with reference to the host clock period as a time unit.

Timing for FLASH devices should take into ac count access times of 60ns and 150ns for the Boot Memory.

Bit 15 **Discard the Setup and Hold parameters for read operations**

Bit 14 **Use asyncronous ready signal for com mand termination.** The asynchronous ready ena ble bit sets the local bus logic to wait for an exter nal ready signal before closing the current cycle.

Bits 13-11 **Command Hold time,** and is determined as follows;

Command hold time = (5+Vs)xT

Bits 10-3 **Command Active time**

Bits 2-0 **Command Setup time**

14.4.2 TIMEIO, I/O Timing Templates

TIMEIO0, TIMEIO1, TIMEIO2 and TIMEIO3 regis ters define the timing template for accessing de vices in I/O Slots 0,1,2 and 3 respectively. The timings are programmed with reference to the host clock period as a time unit.

Bit 15 **Discard the Setup and Hold parameters for read operations**

Bit 14 **Use asyncronous ready signal for com mand termination**

Bits 13-11 **Command Hold time** This is determined by the following formular;

Hold Time = (4+Vh) x T

Where Vh is the Register value for Hold Time

Bits 10-3 **Command Active time** This is determined by the following formular;

Active Time = (4+Va) x T

Where Va is the Register value for the Active Time

Bits 2-0 **Command Setup time** This is determined by the following formular;

Setup Time = (4+Vs) x T

Where Vs is Register value for Setup Time

T is period = 1/Frequency.

At 66MHz, the results should be as follows;

Hold Time = 60ns

Active Time = 15ns

Setup Time = 60ns

14.5 LOCAL BUS CONTROL REGISTER

14.5.1 Control Register

The control register is 8-bit wide.

Bits 7-6 **Register Access Map.**

Bits 7-6	Register Access
00	Access with same index will alternately map to lower and upper and lower Bytes of the 16 bit register as decribed in Section 14.2
01	Register accesses will always map to the lower Byte
10	Register accesses will always map to the upper Byte
11	Access with same index will alternately map to lower and upper and lower Bytes of the 16 bit register as decribed in Section 14.2

Bit 5 **32bit access to Flash memory Enable.** Setting this bit enables 32-bit access to both bank1 and bank0

Bit 4 **Cache Enable for Bank1**

Bit 3 **Cache Enable for Bank0**

Bit 2 **Write Enable for Bank1**

Bit 1 **Write Enable for Bank0**

Bit 0 **Real Mode Boot Access Enable.**

When set this bit enables boot access in Real Mode by mapping 000A0000h to FFFA0000h and 000FFFFFh to FFFFFFFFh

This register defaults to 00h on reset.

14.5.2 IO Width Register

This is an 8 bit register whose 4 less significant bits are used to tell the localbus if an 8 bit peripheral is attached to one of the 4 I/O slots

Bits 7-4, *Reserved.*

Bit 3, If set to 1, the I/O 3 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

Bit 2, If set to 1, the I/O 2 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

Bit 1, If set to 1, the I/O 1 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

Bit 0, If set to 1, the I/O 0 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

This register defaults to 00h on reset.

14.6 CHIP SELECT MEMORY MAP

The address mapping supported by the local Bus interface is summarized by Figure 14-2. Memory address ranges are mapped at fixed addresses while the I/O devices can be mapped from 1 DW (double word) to 256 DW inside a 16 MByte segment.

Table 14-2. Local Bus Address Mapping

Flash Memory Chip Select	**Bank 0**		**Bank 1**	
Address Range	FFC00000h-FFFFFFFFh		FEC00000h-FEFFFFFFh	
Address Space	4 MBytes		4 MBytes	
Boot Address Space (Real Mode)	000A0000h-000FFFFF		-	
I/O Control Chip Select	IOCS#0	IOCS#2	IOCS#3	IOCS#4
Address Range	0000h-FFFFh	0000h-FFFFh	0000h-FFFFh	0000h-FFFFh
Address Space	4 Bytes - 1KByte	4 Bytes - 1KByte	4 Bytes - 1KByte	4 Bytes - 1KByte
Address Mask	8-bit, Selected from 4, 8, 16, 32, 64, 128, 256, 512, 1K			

The memory mapping described in Table 14-2. above is illustrated in Figure 14-1

Figure 14-1. Local Bus Address Mapping Diagram

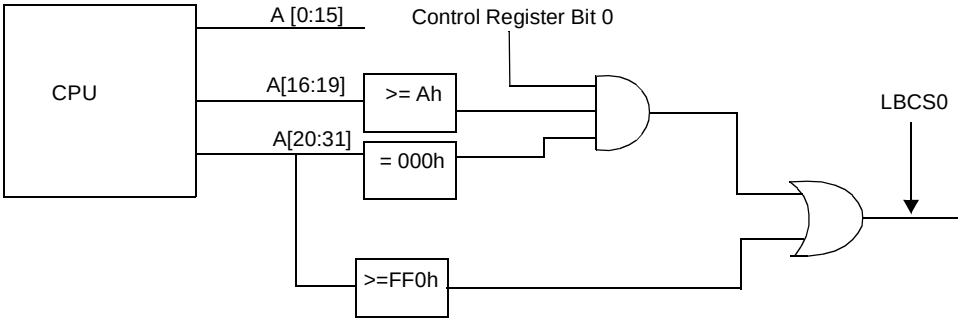
<table>
<tr><td></td><td>FFFFFFFFh

FFC00000h</td><td rowspan="4">BANK 0</td></tr>
<tr><td>Unused</td><td>FF800000h</td></tr>
<tr><td>Unused</td><td>FF400000h</td></tr>
<tr><td>Unused</td><td>FF000000h</td></tr>
<tr><td></td><td>FEFFFFFFh

FEC00000H</td><td rowspan="4">BANK 1</td></tr>
<tr><td>Unused</td><td>FE800000h</td></tr>
<tr><td>Unused</td><td>FE400000h</td></tr>
<tr><td>Unused</td><td>FE000000h</td></tr>
</table>

Memory access to the local bus is based on the following logic scheme see Figure 14-2.;

Figure 14-2. Memory Bank 0 Access Logic



LBCS0 = 0, the CPU accesses to memory are performed on either the DRAM or PCI interfaces.
LBCS0 = 1, the CPU accesses to memory are performed on the Local Bus memory Bank 0.

15. POWER MANAGEMENT

15.1 Introduction

For full information on the action of the SMM, please refer to the STMicroelectronics manual titled “ST486DX SMM programming manual”. This chapter presents the control registers for SMM of the iDragon.

The iDragon 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 appropri ate sub-systems in power down mode, request STPCLK# assertion and stop CPU and other sys tem 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 iDragon is in a power down state will first enable any stopped clocks, wait for a a programmable de lay to allow any internal PLLs to stabilize and then deassert STPCLK# to enable CPU execution. The device can optionally generate SMI interrupt to al low the SMM to bring the system back to power-on state.

The current revision of the iDragon does not im plement support for stopping CPU and other sys tem 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 o 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 ei ther 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 iDragon 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 iDragon decodes the following to detect activities of various kind, see Table 15-1:

Table 15-1. Activity detection

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

Table 15-1. Activity detection

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.

15.2 POWER MANAGEMENT REGISTERS

15.2.1 Timer Register 0 Index 60h (Timer0)

This register controls the timer selection for the length of timeout for doze, standby, and suspend modes.

Bits 7-5 **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 time will reset and start again. These bits are encoded as follows in Table 15-2:

Table 15-2. Suspend timer encoding

7	6	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

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 **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-powe 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, pres ence of an enabled system activity will reset the timer to start counting again. These bits are en coded as follows in Table 15-3:

Table 15-3. Standby timer encoding

4	3	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

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 *Reserved.*

15.2.2 Timer Register 1 Index 61h (Timer1)

Bit 7 *Reserved.*

Bits 6-4 **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 15-4:

Table 15-4. Housekeeping timer encoding

6	5	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

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 **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 15-5:

Table 15-5. Peripheral timeout encoding

3	2	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

The peripheral timer counts whenever it is enabled.

Bit 0 *Reserved.*

This register defaults to 00h after reset.

15.2.3 Timer Register 2 INdex 8Dh (Timer2)

Bits 7-5 **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-powe 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 15-6:

Table 15-6. Doze timeout encoding

7	6	5	Doze Timer reset
0	0	0	disabled
0	0	1	50 milli-seconds

Table 15-6. Doze timeout encoding

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

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 *Reserved.*

This register defaults to 00h at reset.

15.2.4 System Activity Enable Register 0 Index 62h (Sys_activ_en0)

This is the first of the three registers that control which system activity to detect.

Bit 7 **DMA Request (DRQ)**

Bit 6 **PCI master device (PCIM)**

Bit 5 **Parallel IO (PIO)**

Bit 4 **Serial IO (SIO)**

Bit 3 **Keyboard (KBD)**

Bit 2 **Floppy Disk Controller (FDC)**

Bit 1 **Hard Disk Controller (HDC)**

Bit 0 *Reserved.*

This register defaults to 00h at reset, which corresponds to ignoring all of the listed system activities.

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.

15.2.5 System Activity Enable Register 1 Index 63h (Sys_activ_en1)

This is the second of the three registers that control which system activity to detect.

Bits 7-6 *Reserved.* Must be programmed to '0'

Bit 5 **Address range 0**

Bits 4-0 *Reserved.* Must be programmed to '0'

This register defaults to 00h at reset, which corre sponds to ignoring all of the listed system activi ties.

15.2.6 System Activity Enable Register 2 Index 64h (Sys_activ_en2)

This is the third of the three registers that control which system activity to detect.

Bit 7 **IRQ15-** detection enabled

Bit 6 **IRQ0** detection enabled

Bit 5 **NMI detection** enabled

Bits 4-0 *Reserved.*

This register defaults to 00h disabling all activity detection.

This register defaults to 00h disabling detection of all house-keeping activity detection.

15.2.7 House-keeping Activity Enable Reg. 0 Index 65h (HK_activ_en0)

This register controls which house-keeping activi ty 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 pro grammed in the house-keeping timer. Set the fol lowing bits to a '1' to enable activity detection and a '0' to ignore the associated activity.

Bit 7 **DMA Request (DRQ)** activity

Bit 6 **PCI master device** activity

Bit 5 **Keyboards (KBD)** activity

Bit 4 **IRQ15-** activity

Bit 3 **IRQ0** activity

Bit 2 **NMI** activity

Bits 1-0 *Reserved.*

15.2.8 House-keeping Activity Enable Reg. 1 Index 66h (HK_actv_en1)

This is the second house-keeping activity detection enable register.

Bits 7-6 *Reserved.* Must be programmed to '0'

Bit 5 **Address range 0**

Bits 4-0 *Reserved.* Must be programmed to '0'

This register defaults to 00h disabling detection of all house-keeping Activity detection.

15.2.9 Peripheral Inactivity detection Reg. 0 Index 67h (Perif_inactv0)

This register controls which peripheral inactivity is enabled for generating a SMI interrupt on a peripheral time-out.

Bit 7 **Parallel IO (PIO)** activity

Bit 6 **Serial IO (SIO)** activity

Bit 5 **Keyboard (KBD)** activity

Bit 4 **Floppy Disk Controller (FDC)** activity

Bit 3 **Hard Disk Controller (HDC)** activity

Bit 2 **Address range 0**

Bits 1-0 *Reserved.* Must be programmed to '0'

This register defaults to 00h disabling detection of all peripherals.

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.

15.2.10 Peripheral Activity detection Reg. 0 Index 69h (Perif-activ0)

This register controls which peripheral accesses will cause a SMI.

Bit 7 **Parallel port (PIO)** access

Bit 6 **Serial port (SIO)** access

Bit 5 **Keyboard (KBD)** access

Bit 4 **Floppy Disk Controller (FDC)** access

Bit 3 **Hard Disk Controller (HDC)** access

Bit 2 **Address range 0**

Bits 1-0 *Reserved.* Must be programmed to '0'

This register defaults to 00h disabling all SMI generation.

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 fo the associate peripheral and a '0' disables.

15.2.11 Peripheral Activity detection Reg. 1 Index 6Ah (Perif_activ1)

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.

Bits 7-0 *Reserved.* Must be programmed to '0'

This register defaults to 00h.

15.2.12 Address range 0 register 0 Index 6Bh (Address_range0-0

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.

This register defaults to 00h after reset.

15.2.13 Address range 0 register 1 Index 6Ch (Address_rang0-1)

Bits 7-3 These bits are compared with PCI address bits 23-19 if range compare is enabled fo 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.

This register defaults to 00h after reset.

15.2.14 SMI Control register 0
Index 71h (SMI_cont0)

This register controls the generation of SMI interrupt as follows:

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 *Reserved.*

This register defaults to 00h disabling all SMI generation.

15.2.15 SMI status register 0
Index 73h (SMI_stat0)

This register contains the status information pertaining to the SMI interrupt.

Bit 7 **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 **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 **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 **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 **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 2 **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 **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 **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.

15.2.16 SMI Status register 1 Index 74h (SMI_stat1)

This register is similar to SMI Status register 0 in that it reports the cause of the SMI interrupt to the software.

Bit 7 **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 *Reserved.*

This register defaults to 00h after reset.

15.2.17 Peripheral Inactivity status register 0 Index 75h (Perif_status0)

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.

Bit 7 **Parallel IO (PIO)** activity

Bit 6 **Serial IO (SIO)** activity

Bit 5 **Keyboard (KBD)** activity

Bit 4 **Floppy Disk Controller (FDC)** activity

Bit 3 **Hard Disk Controller (HDC)** activity

Bit 2 **Address range 0**

Bits 1-0 R*eserved.*

This register defaults to 00h after reset.
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 ac tivity from the associated peripheral is detected.

The status reflected in this register is not condi tioned by whether the peripheral was enabled fo inactivity detection through the Peripheral Inactivi ty Detection registers or not. The SMI interrupt however will be generated only if any of the en abled peripherals (via Peripheral Inactivity Enable register) were inactive for the entire duration of the peripheral time out.

15.2.18 Activity status register 0 Index 77 (Actv_status0)

This register records presence of activity.

Bit 7 **DMA Request (DRQ)** activity

Bit 6 **PCI master device (PCIM)** activity

Bit 5 **Parallel IO (PIO)** activity

Bit 4 **Serial IO (SIO)** activity

Bit 3 **Keyboard (KBD)** activity

Bit 2 **Floppy Disk Controller (FDC)** activity

Bit 1 **Hard Disk Controller (HDC)** activity

Bit 0 *Reserved.*

This register defaults to 00h after reset.

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.

15.2.19 Activity Status register 1 Index 78h (Activ_stat1)

This register is similar to Activity Status register 0. It contains the status for the following bits.

Bits 7-6 *Reserved.* Must be programmed to '0'

Bit 5 **Address range 0**

Bits 4-0 *Reserved.* Must be programmed to '0'

This register defaults to 00h after reset.

15.2.20 Activity Status register 2 Index 79h (Activ_stat2)

This register is similar to Activity Status registers 0 and 1.

Bit 7 **IRQ15-1** activity

Bit 6 **IRQ0** activity

Bit 5 **NMI** activity

Bits 4-0 *Reserved.*

This register defaults to 00h after reset.

15.2.21 PMU state register Index 7Ah (PMU)

This register contains the state the power management controller currently is in.

Bit 7 *Reserved.*

Bit 6 **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 microsecond clock to tick at oscillator clock frequency instead of every microsecond.

Bit 5 **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 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 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.

Bits 2-0 **PMU state** see Table 15-7

:

Table 15-7. PMU encoding

2	1	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

This register defaults to 00h after reset.

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.

15.2.22 General Purpose Register Index 7Bh (GP)

This is a read/write IO register that can be used by software.

Bit 7 **General Purpose Register Bit 7**

Bit 6 **General Purpose Register Bit 6**

Bit 5 **General Purpose Register Bit 5**

Bit 4 **General Purpose Register Bit 4**

Bit 3 **General Purpose Register Bit 3**

Bit 2 **General Purpose Register Bit 2**

Bit 1 **General Purpose Register Bit 1**

Bit 0 **General Purpose Register Bit 0**

This register defaults to 00h at reset.

Programming notes

Writing to this register also updates the external '373 latch that can be used to control external de vices for power-down purposes. Reads of this reg ister 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.

15.2.23 Clock Control register 0 Index 7Ch (Clock_cont0)

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.

Bits 7-5 **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 15-8:

Table 15-8. STPCL# modulation encoding

7	6	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.*

The STPCLK# is deasserted and the duty-cycle control ignored if a SMI interrupt is pending.

Bits 4-2 **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 15-9:

Table 15-9. D/S/S State encoding

4	3	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

The STPCLK# is deasserted and the duty-cycle control ignored if a SMI interrupt is pending.

Bit 1 **STPCLK# modulation period.** If '1' then the period is 64ms else, if '0', then the period is 64ms.

Bit 0 *Reserved.*

This register defaults to 00h.

15.2.24 Doze timer read back register Index 88h (Doze)

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.

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.

15.2.25 Standby timer read back register Index 89h (Standby)

This read only register is provided for test purposes to read back the current value of 5-bit standby timer.

Bits 7-5 *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.

15.2.26 Suspend timer read back register Index 8Ah (Suspend)

This read only register is provided for test purposes to read back the current value of the 7-bit Suspend timer.

Bit 7 *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.

15.2.27 House-keeping timer read back Reg. Index 8Bh (HK_timer)

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.

Bits 7-0 Bits 8-1 of the house-keeping timer.

Programming notes

This register should not be used by software.

15.2.28 Peripheral timer read back register Index 8Ch (Perif_timer)

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.

Bits 7-0 Bits 8-1 of the Peripheral timer.

Programming notes

This register should not be used by software.