STPC CLIENT

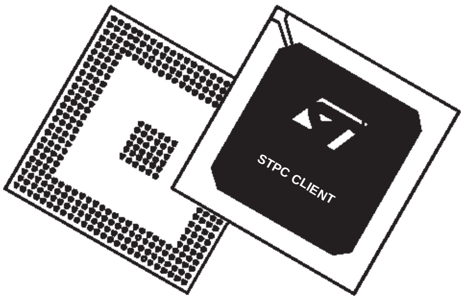
Multimedia PC on a Chip

- POWERFUL X86 PROCESSOR
- 64-BIT 66MHz BUS INTERFACE
- 64-BIT DRAM CONTROLLER
- SVGA GRAPHICS CONTROLLER
- UMA ARCHITECTURE
- VIDEO SCALER
- VIDEO OUTPUT PORT
- VIDEO INPUT PORT
- CRT CONTROLLER
- 135MHz RAMDAC
- 2 OR 3 LINE FLICKER FILTER
- SCAN CONVERTER
- PCI MASTER / SLAVE / ARBITER
- ISA MASTER/SLAVE
- IDE CONTROLLER
- DMA CONTROLLER
- INTERRUPT CONTROLLER
- TIMER / COUNTERS
- POWER MANAGEMENT

DESCRIPTION

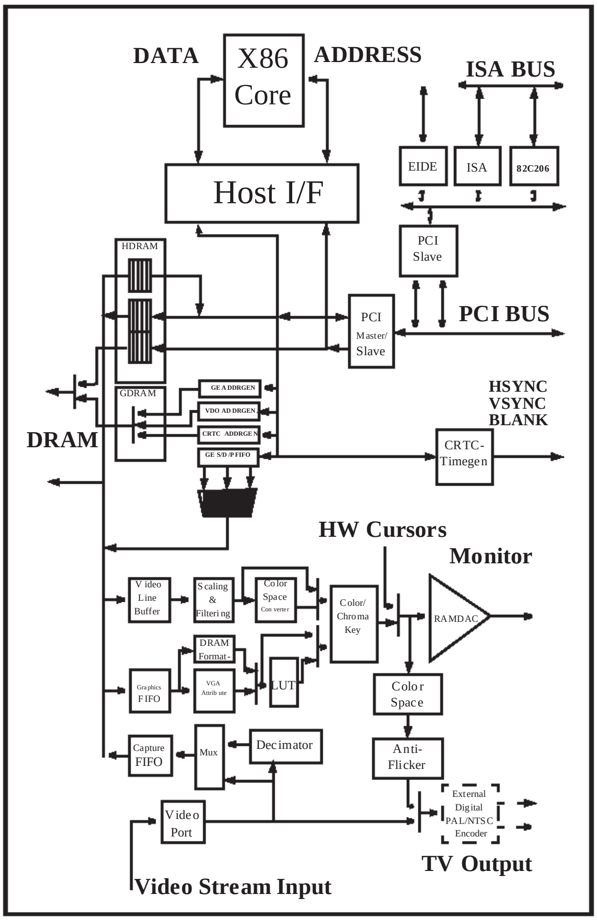
The STPC Client integrates a fully static x86 processor, fully compatible with standard x86 processors, and combines it with powerful chipset, graphics and video pipelines to provide a single Consumer orientated PC compatible subsystem on a single device. The device is packaged in a 388 Ball Grid Array (PBGA).

- **X86 Processor core**
- Fully static 32-bit 5-stage pipeline, x86 processor fully PC compatible.
- Can access up to 4GB of external memory.
- 8Kbyte unified instruction and data cache with write back and write through capability.
- Parallel processing integral floating point unit, with automatic power down.



PBGA388

Figure 1. Logic Diagram



- Clock core speeds up to of 133 MHz.
- Fully static design for dynamic clock control.
- Low power and system management modes.
- Optimized design for 3.3V operation.

- **DRAM Controller**
- Integrated system memory and graphic frame memory.
- Supports up to 128 MBytes system memory in 4 banks and as little as 2Mbytes.
- Supports 4MB, 8MB, 16MB, 32MB single-sided and double-sided DRAM SIMMs.
- Four quad-word write buffers for CPU to DRAM and PCI to DRAM cycles.
- Four 4-word read buffers for PCI masters.
- Supports Fast Page Mode & EDO DRAMs.
- Programmable timing for DRAM parameters including CAS pulse width, CAS pre-charge time, and RAS to CAS delay.
- 60, 70, 80 & 100ns DRAM speeds.
- Memory hole between 1 MByte & 8 MByte supported for PCI/ISA busses.
- Hidden refresh.

- **Graphics Controller**
- 64-bit windows accelerator.
- Backward compatibility to SVGA standards.
- Hardware acceleration for text, bitblts, transparent blts and fills.
- Up to 64 x 64 bit graphics hardware cursor.
- Up to 4MB long linear frame buffer.
- 8-, 16-, and 24-bit pixels.
- Drivers for Windows and other operating systems.

- **CRT Controller**
- Integrated 135MHz triple RAMDAC allowing for 1024 x 768 x 75Hz display.
- Requires external frequency synthesizer and reference sources.
- 8-, 16-, 24-bit pixels.
- Interlaced or non-interlaced output.

- **Video Pipeline**
- Two-tap interpolative horizontal filter.
- Two-tap interpolative vertical filter.
- Color space conversion (RGB to YUV and YUV to RGB).
- Programmable window size.
- Chroma and color keying for integrated video overlay.
- Programmable two tap filter with gamma correction or three tap flicker filter.
- Progressive to interlaced scan converter.

- **Video Input port**
- Accepts video inputs in CCIR 601/656 or ITU-R 601/656, and decodes the stream.
- Optional 2:1 decimator
- Stores captured video in off setting area of the onboard frame buffer.
- Video pass through to the onboard PAL/ NTSC encoder for full screen video images.
- HSYNC and B/T generation or lock onto external video timing source.

- **PCI Controller**
- Fully compliant with PCI 2.1 specification.
- Integrated PCI arbitration interface. Up to 3 masters can connect directly. External PAL allows for greater than 3 masters.
- Translation of PCI cycles to ISA bus.
- Translation of ISA master initiated cycle to PCI.
- Support for burst read/write from PCI master.
- 0.33X and 0.5X CPU clock PCI clock.

- **ISA master/slave**
- Generates the ISA clock from either 14.318MHz oscillator clock or PCI clock
- Supports programmable extra wait state for ISA cycles
- Supports I/O recovery time for back to back I/O cycles.
- Fast Gate A20 and Fast reset.
- Supports the single ROM that C, D, or E. blocks shares with F block BIOS ROM.
- Supports flash ROM.
- Supports ISA hidden refresh.
- Buffered DMA & ISA master cycles to reduce bandwidth utilization of the PCI and Host bus. NSP compliant.

- **IDE Interface**
- Supports PIO and Bus Master IDE
- Supports up to Mode 5 Timings
- Transfer Rates to 22 MBytes/sec
- Supports up to 4 IDE devices
- Concurrent channel operation (PIO & DMA modes) - 4 x 32-Bit Buffer FIFO per channel
- Support for PIO mode 3 & 4.
- Support for DMA mode 1 & 2.
- Support for 11.1/16.6 MB/s, I/O Channel Ready PIO data transfers.
- Supports 13.3/16.6 MB/s DMA data transfers
- Bus Master with scatter/gather capability
- Multi-word DMA support for fast IDE drives
- Individual drive timing for all four IDE devices
- Supports both legacy & native IDE modes
- Supports hard drives larger than 528MB
- Support for CD-ROM and tape peripherals
- Backward compatibility with IDE (ATA-1).
- Drivers for Windows and other OSes

- **Integrated peripheral controller**
- 2X8237/AT compatible 7-channel DMA controller.
- 2X8259/AT compatible interrupt Controller. 16 interrupt inputs - ISA and PCI.
- Three 8254 compatible Timer/Counters.
- Co-processor error support logic.

- **Power Management**
- Four power saving modes: On, Doze, Standby, Suspend.
- Programmable system activity detector
- Supports SMM and APM.
- Supports STOPCLK.
- Supports IO trap & restart.
- Independent peripheral time-out timer to monitor hard disk, serial & parallel ports.
- Slow system clock down to 8MHz
- Slow Host clock down to 8Hz
- Slow graphic clock down to 8Hz
- Supports RTC, interrupts and DMAs wake-up

Table of Contents

Table of Contents

List of Figures

List of Tables

1 GENERAL DESCRIPTION

At the heart of the STPC Client is an advanced processor block, dubbed the 5ST86. The 5ST86 includes a powerful x86 processor core along with a 64-bit DRAM controller, advanced 64bit accelerated graphics and video controller, a high speed PCI local-bus controller and Industry standard PC chip set functions (Interrupt controller, DMA Controller, Interval timer and ISA bus) and EIDE controller.

The STPC Client has in addition to the 5ST86 a Video subsystem and high quality digital Television output.

The STMicroelectronics x86 processor core is embedded with standard and application specific peripheral modules on the same silicon die. The core has all the functionality of the SGS-THOMSON standard x86 processor products, including the low power System Management Mode (SMM).

System Management Mode (SMM) provides an additional interrupt and address space that can be used for system power management or software transparent emulation of peripherals. While running in isolated SMM address space, the SMM interrupt routine can execute without interfering with the operating system or application programs.

Further power management facilities include a suspend mode that can be initiated from either hardware or software. Because of the static nature of the core, no internal data is lost.

The STPC Client makes use of a tightly coupled Unified Memory Architecture (UMA), where the same memory array is used for CPU main memory and graphics frame-buffer. This significantly reduces total system memory with system performances equal to that of a comparable solution with separate frame buffer and system memory. In addition, memory bandwidth is improved by attaching the graphics engine directly to the 64-bit processor host interface running at the speed of the processor bus rather than the traditional PCI bus.

The 64-bit wide memory array provides the system with 320MB/s peak bandwidth, double that of an equivalent system using 32 bits. This allows for higher screen resolutions and greater color depth. The processor bus runs at the speed of the processor (DX devices) or half the speed (DX2 devices).

The ‘standard’ PC chipset functions (DMA, interrupt controller, timers, power management logic) are integrated with the x86 processor core.

The PCI bus is the main data communication link to the STPC Client chip. The STPC Client translates appropriate host bus I/O and Memory cycles onto the PCI bus. It also supports generation of Configuration cycles on the PCI bus. The STPC Client, as a PCI bus agent (host bridge class), fully complies with PCI specification 2.1. The chip-set also implements the PCI mandatory header registers in Type 0 PCI configuration space for easy porting of PCI aware system BIOS. The device contains a PCI arbitration function for three external PCI devices.

The STPC Client integrates an ISA bus controller. Peripheral modules such as parallel and serial communications ports, keyboard controllers and additional ISA devices can be accessed by the STPC Client chip set through this bus.

An industry standard EIDE (ATA 2) controller is built in to the STPC Client and connected internally via the PCI bus.

Graphics functions are controlled by the on-chip SVGA controller and the monitor display is managed bythe 2D graphics display engine.

This Graphics Engine is tuned to work with the host CPU to provide a balanced graphics system with a low silicon area cost. It performs limited graphics drawing operations, which include hardware acceleration of text, bitblts, transparent blts and fills. These operations can operate on offscreen or on-screen areas. The frame buffer size is up to 4 Mbytes anywhere in the physical main memory.

The graphics resolution supported is a maximum of 1280x1024 in 65536 colours at 75Hz refresh rate and is VGA and SVGA compatible. Horizontal timing fields are VGA compatible while the vertical fields are extended by one bit to accommodate above display resolution.

STPC Client provides several additional functions to handle MPEG or similar video streams. The Video Input Port accepts an encoded digital video stream in one of a number of industry standard formats, decodes it, optionally decimates it by a factor of 2:1, and deposits it into an off screen area of the frame buffer. An interrupt request can be generated when an entire field or frame has been captured.

The video output pipeline incorporates a video-scaler and color space converter function and provisions in the CRT controller to display a video window. While repainting the screen the CRT controller fetches both the video as well as the normal non-video frame buffer in two separate internal FIFOs (256-bytes each). The video stream can be color-space converted (optionally) and smooth scaled. Smooth interpolative scaling in both horizontal and vertical direction are implemented. Color and Chroma key functions are also implemented to allow mixing video stream with non-video frame buffer.

The video output passes directly to the RAMDAC for monitor output or through another optional color space converter (RGB to 4:2:2 YCrCb) to the programmable anti-flicker filter. The flicker filter is configured as either a two line filter with gamma correction (primarily designed for DOS type text) or a 3 line flicker filter (primarily designed for Windows type displays). The flicker filter is optional and can be software disabled for use with large screen area's of video.

The Video output pipeline of the STPC Client interfaces directly to the external digital TV encoder (STV0119). It takes a 24 bit RGB non-interlaced pixel stream and converts to a multiplexed 4:2:2 YCrCb 8 bit output stream, the logic includes a progressive to interlaced scan converter and logic to insert appropriate CCIR656 timing reference codes into the output stream. It facilitates the high quality display of VGA or full screen video streams received via the Video input port to standard NTSC or PAL televisions.

The STPC Client core is compliant with the Advanced Power Management (APM) specification to provide a standard method by which the BIOS can control the power used by personal computers. The Power Management Unit module (PMU) controls the power consumption by providing a comprehensive set of features that control the power usage and supports compliance with the United States Environmental Protection Agency's Energy Star Computer Program. The PMU provides following hardware structures to assist the software in managing the power consumption by the system.

- System Activity Detection.

- 3 power-down timers detecting system inactivity:

  - Doze timer (short durations).
  - Stand-by timer (medium durations).
  - Suspend timer (long durations).

- House-keeping activity detection.

- House-keeping timer to cope with short bursts of house-keeping activity while dozing or in stand-by state.

- Peripheral activity detection.

- Peripheral timer detecting peripheral inactivity

- SUSP# modulation to adjust the system performance in various power down states of the system including full power on state.

- Power control outputs to disable power from different planes of the board.

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. Alternatively, system activity in a power down state can generate SMI interrupt to allow the software to bring the system back up to full power on state. The chip-set supports up to three power down states: Doze state, Stand-by state and Suspend mode. These correspond to decreasing levels of power savings.

Power down puts the STPC Client into suspend mode. The processor completes execution of the current instruction, any pending decoded instructions and associated bus cycles. During the suspend mode, internal clocks are stopped. removing power down, the processor resumes instruction fetching and begins execution in the instruction stream at the point it had stopped.

A reference design for the STPC Client is available including the schematics and layout files, the design is a PC ATX motherboard design. The design is available as a demonstration board for application and system development.

The STPC Client is supported by several BIOS vendors, including the super I/O device used in the reference design. Drivers for 2D accelerator, video features and EIDE are availaible on various operating systems.

The STPC Client has been designed using modern reusable modular design techniques, it is possible to add to or remove the standard features of the STPC Client or other variants of the 5ST86 family. Contact your local STMicroelectonics sales office for further information.

Figure 2. Interfaces

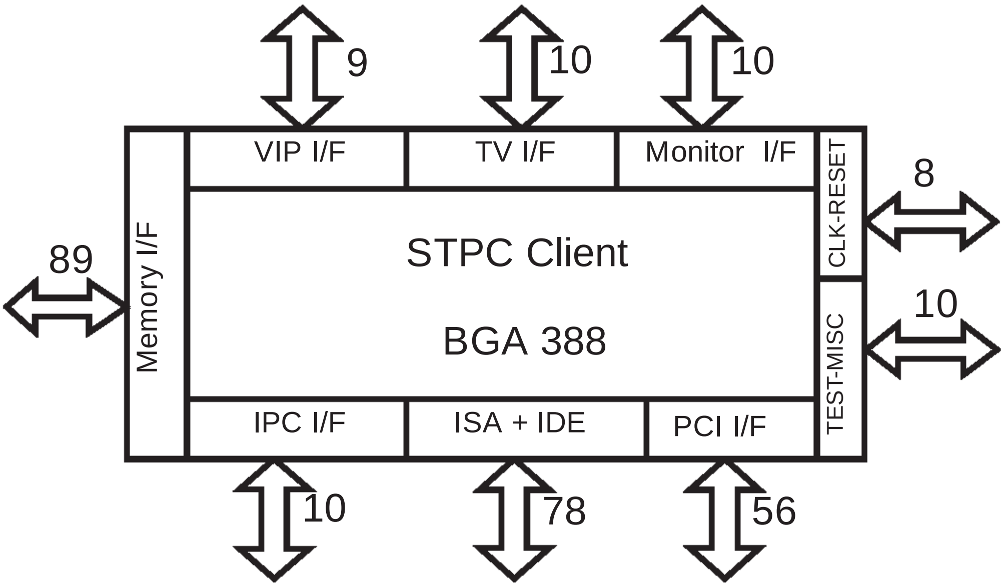


Figure 3. Die Highlight

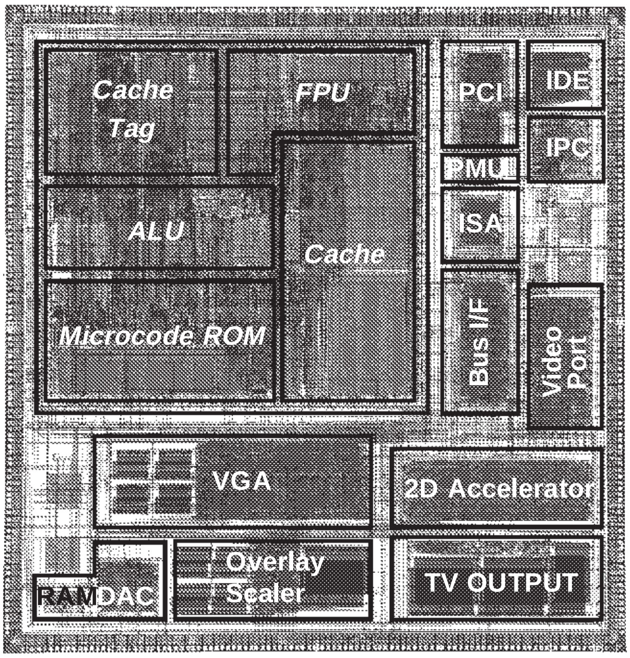


Figure 4. Functionnal description.

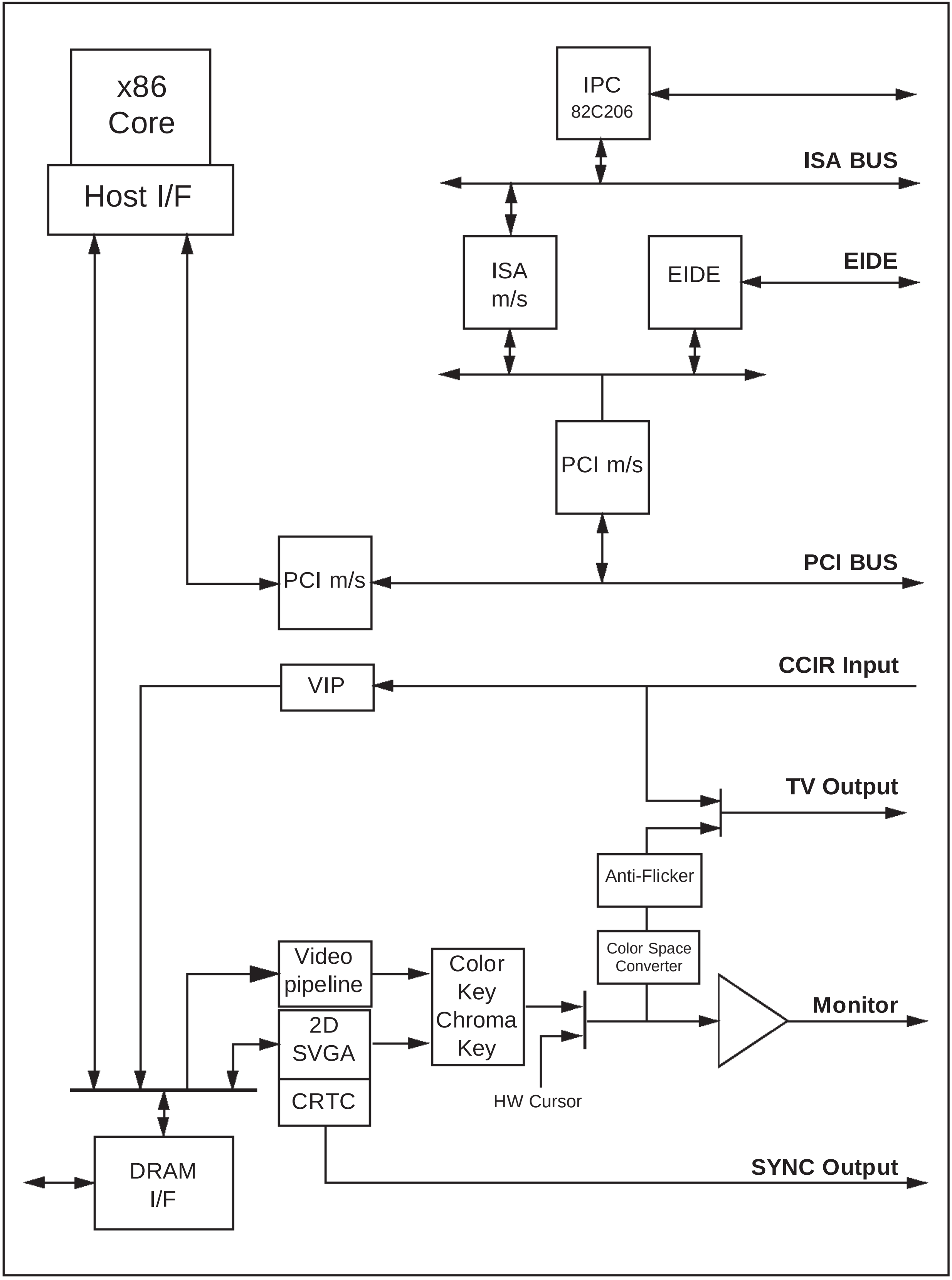


Figure 5. Pictorial Block Diagram

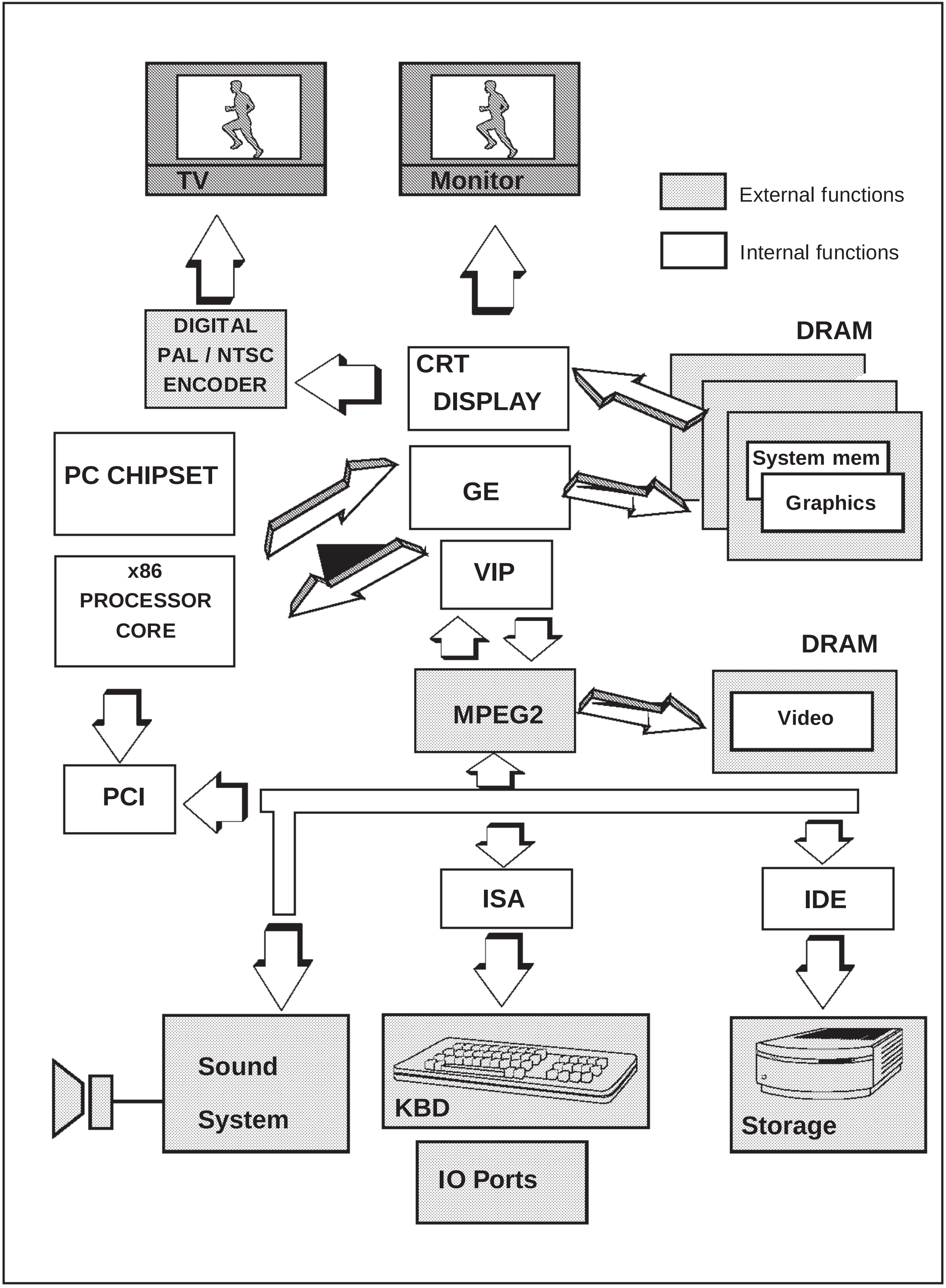
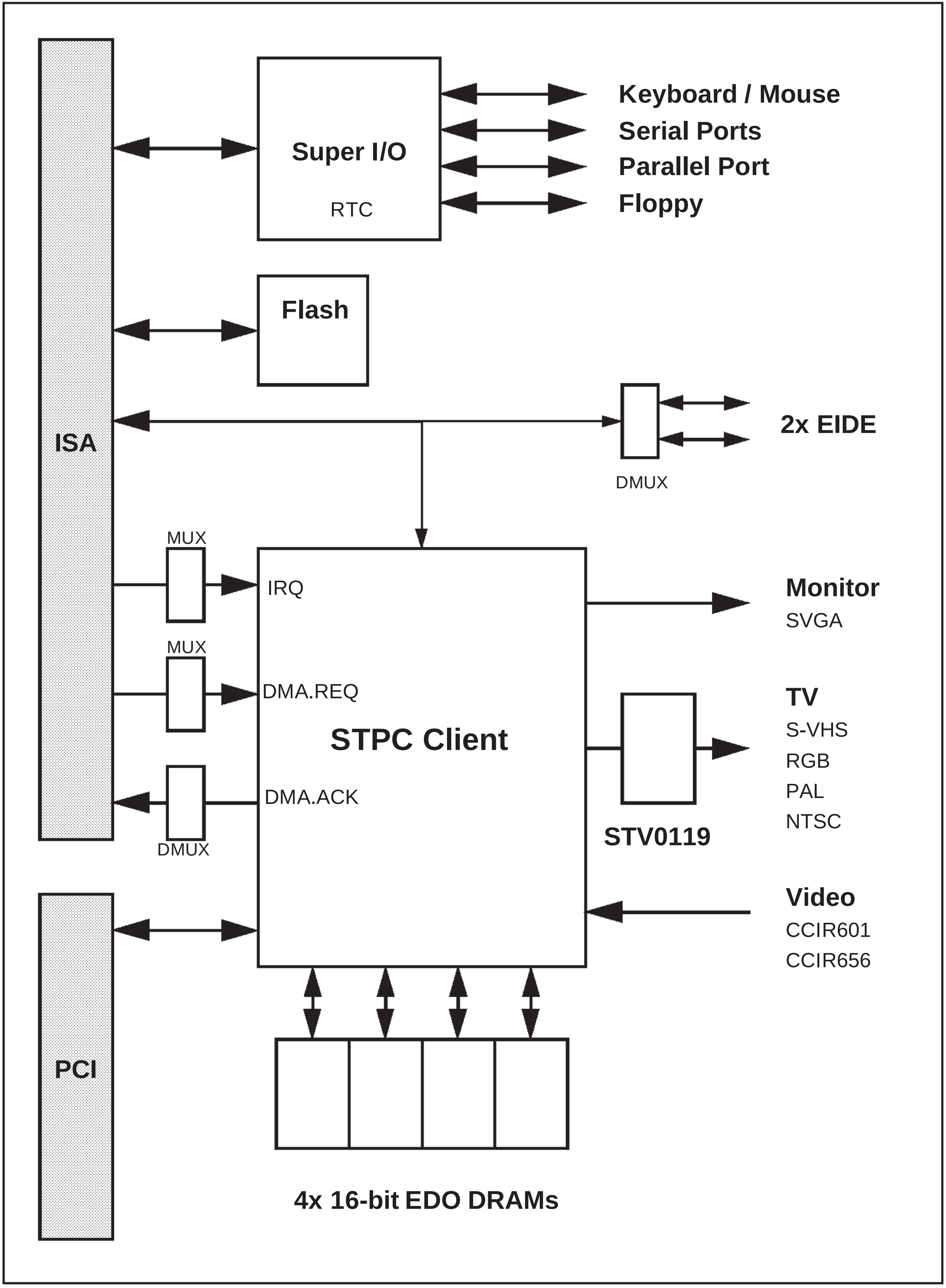


Figure 6. Typical Application



2. HOW TO USE THIS MANUAL

2.1. INTRODUCTION

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

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

2.2. SPECIFIC NOTES

2.2.1. RESERVED BITS

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.

2.2.2. 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.

2.2.3. HEX 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 7830A hexadecimal will be written 783Ah.

2.2.4. 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

3 PIN DESCRIPTION

3.1 INTRODUCTION

The STPC Client integrates most of the functionalities of the PC architecture. As a result, many of the traditional interconnections between the host PC microprocessor and the peripheral devices are totally internal to the STPC Client. This offers improved performance due to the tight coupling of the processor core and these peripherals. As a result many of the external pin connections are made directly to the on-chip peripheralfunctions.

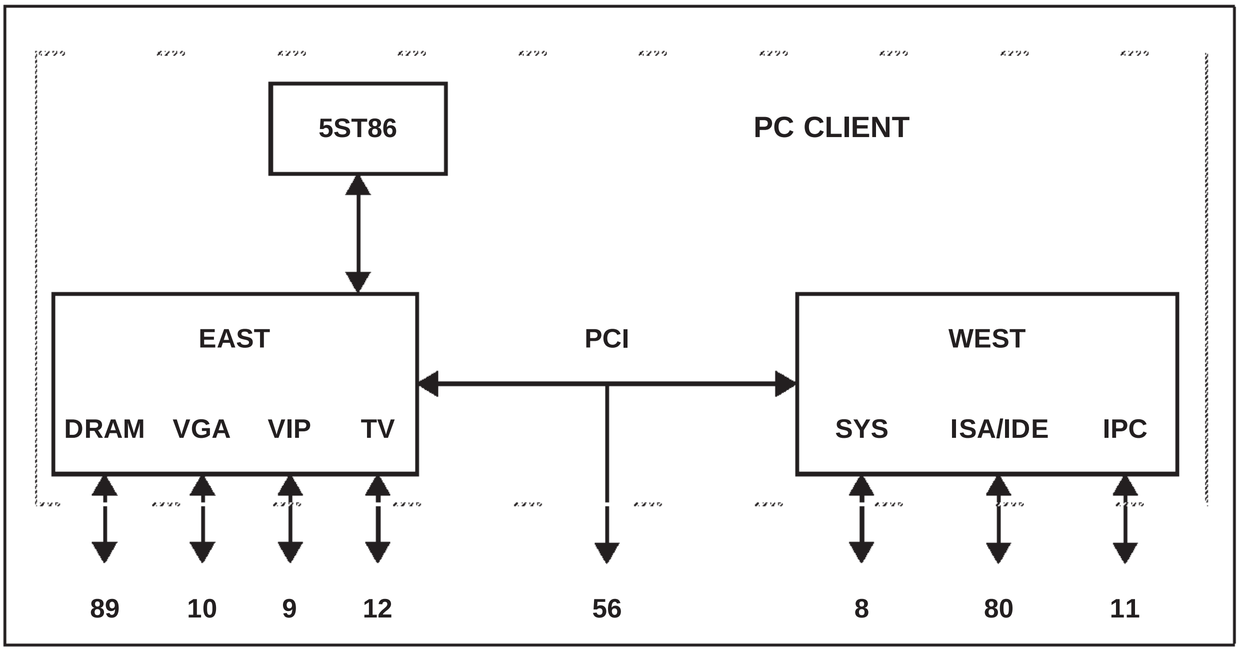
Figure 7 shows the STPC Client's external interfaces. It defines the main busses and their function. Table 1 describes the physical implementation listing signals type and their functionality. Table 2 provides a full pin listing and description of pins. Table 3 provides a full listing of pin locations of the STPC Client package by physical connection. Please refer to the pin allocation drawing for reference.

Table 1. Signal Description

Group name	Qty
Basic Clocks reset & Xtal	7
Memory Interface	89
PCI interface	56
ISA / IDE / IPC combined interface	88
Video Input	9
Video Output	10
VGA Monitor interface	10
Grounds	69
V_{DD}	24
Analog specific V_{CC}/V_{DD}	16
Reserved	10
Total Pin Count	388

Note: Several interface pins are multiplexed with other functions, refer to the Pin Description section for further details

Figure 7. PC Client External Interfaces



PIN DESCRIPTION

Table 2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
BASIC CLOCKS AND RESETS			
PWERGD	I	System Reset / Power good	1
XTALI	I	14.3MHz Crystal Input	1
XTALO	I/O	14.3MHz Crystal Output - External Oscillator Input	1
HCLK	O	Host Clock (Test)	1
DEV_CLK	O	24MHz Peripheral Clock (floppy drive)	1
GCLK2X	I/O	80MHz Graphics Clock	1
DCLK	I/O	135MHz Dot Clock	1
DCLK _DIR	I	Dot Clock Direction	1
MEMORY INTERFACE			
MA[11:0]	I/O	Memory Address	12
RAS#[3:0]	O	Row Address Strobe	4
CAS#[7:0]	O	Column Address Strobe	8
MWE#	O	Write Enable	1
MD[63:0]	I/O	Memory Data	64
PCI INTERFACE			
PCI_CLKI	I	33MHz PCI Input Clock	1
PCI_CLKO	O	33MHz PCI Output Clock (from internal PLL)	1
AD[31:0]	I/O	PCI Address / Data	32
CBE[3:0]	I/O	Bus Commands / Byte Enables	4
FRAME#	I/O	Cycle Frame	1
TRDY#	I/O	Target Ready	1
IRDY#	I/O	Initiator Ready	1
STOP#	I/O	Stop Transaction	1
DEVSEL#	I/O	Device Select	1
PAR	I/O	Parity Signal Transactions	1
SERR#	O	System Error	1
LOCK#	I	PCI Lock	1
PCIREQ#[2:0]	I	PCI Request	3
PCIGNT#[2:0]	O	PCI Grant	3
PCI_INT[3:0]	I	PCI Interrupt Request	4
ISA AND IDE COMBINED ADDRESS/DATA			
LA[23:22] / SCS3#,SCS1#	I/O	Unlatched Address (ISA) / Secondary Chip Select (IDE)	2
LA[21:20] / PCS3#,PCS1#	I/O	Unlatched Address (ISA) / Primary Chip Select (IDE)	2
LA[19:17] / DA[2:0]	O	Unlatched Address (ISA) / Address (IDE)	3
RMRTCCS# / DD[15]	I/O	ROM/RTC Chip Select / Data Bus bit 15 (IDE)	1
KBCS# / DD[14]	I/O	Keyboard Chip Select / Data Bus bit 14 (IDE)	1
RTCRW# / DD[13]	I/O	RTC Read/Write / Data Bus bit 13 (IDE)	1
RTCDS# / DD[12]	I/O	RTC Data Strobe / Data Bus bit 12 (IDE)	1
SA[19:8] / DD[11:0]	I/O	Latched Address (ISA) / Data Bus (IDE)	16
SA[7:0]	I/O	Latched Address (IDE)	4
SD[15:0]	I/O	Data Bus (ISA)	16
ISA/IDE COMBINED CONTROL			
IOCHRDY / DIORDY	I/O	I/O Channel Ready (ISA) - Busy/Ready (IDE)	1

Table 2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
ISA CONTROL			
SYSRSTO#	O	Reset Output to System	1
ISA_CLK	O	ISA Clock Output - Multiplexer Select Line For IPC	1
ISA_CLK2X	O	ISA Clock x 2 Output - Multiplexer Select Line For IPC	1
OSC14M	O	ISA bus synchronisation clock	1
ALE	O	Address Latch Enable	1
BHE#	I/O	System Bus High Enable	1
MEMR#, MEMW#	I/O	Memory Read and Memory Write	2
SMEMR#, SMEMW#	O	System Memory Read and Memory Write	2
IOR#, IOW#	I/O	I/O Read and Write	2
MASTER#	I	Add On Card Owns Bus	1
MCS16#, IOCS16#	I	Memory/IO Chip Select16	2
REF#	O	Refresh Cycle.	1
AEN	O	Address Enable	1
IOCHCK#	I	I/O Channel Check.	1
ISAOE#	O	Bidirectional OE Control	1
GPIOCS#	I/O	General Purpose Chip Select	1
IDE CONTROL			
PIRQ	I	Primary Interrupt Request	1
SIRQ	I	Secondary Interrupt Request	1
PDRQ	I	Primary DMA Request	1
SDRQ	I	Secondary DMA Request	1
PDACK#	O	Primary DMA Acknowledge	1
SDACK#	O	Secondary DMA Acknowledge	1
PIOR#	I/O	Primary I/O Read	1
PIOW#	O	Primary I/O Write	1
SIOR#	I/O	Secondary I/O Read	1
SIOW#	O	Secondary I/O Write	1
IPC			
IRQ_MUX[3:0]	I	Multiplexed Interrupt Request	4
DREQ_MUX[1:0]	I	Multiplexed DMA Request	2
DACK_ENC[2:0]	O	DMA Acknowledge	3
TC	O	ISA Terminal Count	1
MONITOR INTERFACE			
RED, GREEN, BLUE	O	Red, Green, Blue	3
VSYNC	O	Vertical Sync	1
HSYNC	O	Horizontal Sync	1
VREF_DAC	I	DAC Voltage reference	1
RSET	I	Resistor Set	1
COMP	I	Compensation	1
DDC[1:0]	I/O	Display Data Channel Serial Link	2
VIDEO INPUT			
VCLK	I	Pixel Clock	1
VIN	I	YUV Video Data Input CCIR 601 or 656	8

Table 2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
DIGITAL TV OUTPUT			
TV_YUV[7:0]	O	Digital Video Outputs	8
VTV_BT#	O	Frame Synchronisation	1
VTV_HSYNC	O	Horizontal Line Synchronisation	1
MISCELLANEOUS			
ST[6:0]	I/O	Reserved (Test/Misc pins)	7
CLKDEL[2:0]	I/O	Reserved (Test/Misc pins)	3

3.2 SIGNAL DESCRIPTIONS

3.2.1 BASIC CLOCKS AND RESETS

PWGD *System Reset/Power good.* This input is low when the the reset switch is depressed. Otherwise, it reflects the power supply's power good signal. PWGD is asynchronous to all clocks, and acts as a negative active reset. The reset circuit initiates a hard reset on the rising edge of PWGD.

SYSRSTO# *Reset Output to System.* This is the system reset signal and is used to reset the rest of the components (not on Host bus) in the system. The ISA bus reset is an externally inverted buffered version of this output and the PCI bus reset is an externally buffered version of this output.

XTALI *14.3MHz Crystal Input*
XTALO *14.3MHz Crystal Output.* These pins are the 14.318 MHz crystal input; This clock is used as the reference clock for the internal frequency synthesizer to generate the HCLK, CLK24M, GCLK2X and DCLK clocks.
A 14.318 MHz Series Cut Quartz Crystal should be connected between these two pins. Balance capacitors of 15 pF should also be added. In the event of an external oscillator providing the master clock signal to the STPC Client device, the TTL signal should be provided on XTALO.

HCLK *Host Clock.* This is the host 1X clock. Its frequency can vary from 25 to 75 MHz. All host transactions and PCI transactions are synchronized to this clock. The DRAM controller to execute the host transactions is also driven by this clock. In normal mode, this output clock is generated by the internal pll.

PCI_CLKI *33MHz PCI Input Clock*

This signal is the PCI bus clock input and should be driven from the PCI_CLKO pin.

PCI_CLKO *33MHz PCI Output Clock.* This is the master PCI bus clock output.

GCLK2X *80MHz Graphics Clock.* This is the Graphics 2X clock, which drives the graphics engine and the the DRAM controller to execute the graphics and display cycles.
Normally GCLK2X is generated by the internal frequency synthesizer, and this pin is an output. By setting a bit in Strap Register 2, this pin can be made an input so that an external clock can replace the internal frequency synthesizer.

DCLK *135MHz Dot Clock.* This is the dot clock, which drives graphics display cycles. Its frequency can be as high as 135 MHz, and it is required to have a worst case duty cycle of 60-40.
This signal is either driven by the internal pll (VGA) either by an external 27MHz oscillator (TV). The direction can be controlled by a strap option or an internal register bit.

DCLK_DIR *Dot Clock Direction.* Specify if DCLK is an input (0) or an output (1).

ISA_CLK *ISA Clock Output (also Multiplexer Select Line For IPC).* This pin produces the Clock signal for the ISA bus. It is also used with ISA_CLK2X as the multiplexor control lines for the Interrupt Controller Interrupt input lines. This is divided down version of either the PCICLK or OSC14M.

ISA_CLKX2 *ISA Clock Output (also Multiplexer Select Line For IPC).* This pin produces a signal at twice the frequency of the Clock signal for the ISA bus. It is also used with ISA_CLK as the multiplexor control lines for the Interrupt Controller Interrupt input lines.

DEV_CLK *24MHz Peripheral Clock Output.* This 24MHZ signal is provided as a convenience for the system integration of a Floppy Disk driver function in an external chip.

OSC14M *ISA bus synchronisation clock Output.* This is the buffered 14.318 Mhz clock to the ISA bus.

3.2.2 MEMORY INTERFACE

MA[11:0] *Memory Address Output.* These 12 multiplexed memory address pins support external DRAM with up to 4K refresh. These include all 16M x N and some 4M x N DRAM modules. The address signals must be externally buffered to support more than 16 DRAM chips. The timing of these signals can be adjusted by software to match the timings of most DRAM modules.

MD[63:0] *Memory Data I/O.* This is the 64-bit memory data bus. If only half of a bank is populated, MD63-32 is pulled high, data is on MD31-0. MD[40-0] are read by the device strap option registers during rising edge of PWGD.

RAS#[3:0] *Row Address Strobe Output.* There are 4 active low row address strobe outputs, one each for each bank of the memory. Each bank contains 4 or 8-bytes of data. The memory controller allows half of a bank (4-bytes) to be populated to enable memory upgrade at finer granularity.
The RAS# signals drive the SIMMs directly without any external buffering. These pins are always outputs, but they can also simultaneously be inputs, to allow the memory controller to monitor the value of the RAS# signals at the pins.

CAS#[7:0] *Column Address Strobe Output.*There are 8 active low column address strobe outputs, one each for each byte of the memory.
The CAS# signals drive the SIMMs either directly or through external buffers.
These pins are always outputs, but they can also simultaneously be inputs, to allow the memory controller to monitor the value of the CAS# signals at the pins.

MWE# *Write Enable Output.* Write enable specifies whether the memory access is a read (MWE# = H) or a write (MWE# = L). This single write enable controls all DRAMs. It can be externally buffered to boost the maximum number of loads (DRAM chips) supported.
The MWE# signals drive the SIMMs directly without any external buffering.

3.2.3 VIDEO INTERFACE

VIDEO_CLK *Pixel Clock Input.*

VIDEO_D[7:0] *YUV Video Data Input CCIR 601 or 656.* Time multiplexed 4:2:2 luminance and chrominance data as defined in ITU-R Rec601-2 and Rec656 (except for TTL input levels). This bus interfaces with an MPEG video decoder output port and typically carries a stream of Cb,Y,Cr,Y digital video at VCLK frequency, clocked on the rising edge (by default) of VCLK. A 54-Mbit/s ‘double’ Cb, Y, Cr, Y input multiplex is supported for double encoding application (rising and falling edge of CKREF are operating).

3.2.4 TV OUTPUT

TV_YUV[7:0] *Digital video outputs.*

VTV_BT# *Frame Synchronisation*.

VTV_HSYNC *Horizontal Line Synchronisation*.

3.2.5 PCI INTERFACE

AD[31:0] *PCI Address/Data.* This is the 32-bit multiplexed address and data bus of the PCI. This bus is driven by the master during the address phase and data phase of write transactions. It is driven by the target during data phase of read transactions.

CBE#[3:0] *Bus Commands/Byte Enables.* These are the multiplexed command and byte enable signals of the PCI bus. During the address phase they define the command and during the data phase they carry the byte enable information. These pins are inputs when a PCI master other than the STPC Client owns the bus and outputs when the STPC Client owns the bus.

FRAME# *Cycle Frame.* This is the frame signal of the PCI bus. It is an input when a PCI master owns the bus and is an output when STPC Client owns the PCI bus.

TRDY# *Target Ready.* This is the target ready signal of the PCI bus. It is driven as an output when the STPC Client is the target of the current bus transaction. It is used as an input when STPC Client initiates a cycle on the PCI bus.

IRDY# *Initiator Ready.* This is the initiator ready signal of the PCI bus. It is used as an output when the STPC Client initiates a bus cycle on the PCI bus. It is used as an input during the PCI cycles targeted to the STPC Client to determine when the current PCI master is ready to complete the current transaction.

STOP# *Stop Transaction.* Stop is used to implement the disconnect, retry and abort protocol of the PCI bus. It is used as an input for the bus cycles initiated by the STPC Client and is used as an output when a PCI master cycle is targeted to the STPC Client.

DEVSEL# *I/O Device Select.* This signal is used as an input when the STPC Client initiates a bus cycle on the PCI bus to determine if a PCI slave device has decoded itself to be the target of the current transaction. It is asserted as an output either when the STPC Client is the target of the current PCI transaction or when no other device asserts DEVSEL# prior to the subtractive decode phase of the current PCI transaction.

PAR *Parity Signal Transactions.* This is the parity signal of the PCI bus. This signal is used to guarantee even parity across AD[31:0], CBE#[3:0], and PAR. This signal is driven by the master during the address phase and data phase of write transactions. It is driven by the target during data phase of read transactions. (Its assertion is identical to that of the AD bus delayed by one PCI clock cycle)

SERR# *System Error.* This is the system error signal of the PCI bus. It may, if enabled, be asserted for one PCI clock cycle if target aborts a STPC Client initiated PCI transaction. Its assertion by either the STPC Client or by another PCI bus agent will trigger the assertion of NMI to the host CPU. This is an open drain output.

LOCK# *PCI Lock.* This is the lock signal of the PCI bus and is used to implement the exclusive bus operations when acting as a PCI target agent.

PCIREQ#[2:0] *PCI Request.* This pin are the three external PCI master request pins. They indicates to the PCI arbiter that the external agents desire use of the bus.

PCIGNT#[2:0] *PCI Grant.* These pins indicate that the PCI bus has been granted to the master requesting it on its PCIREQ#.

3.2.6 ISA/IDE COMBINED ADDRESS/DATA

LA[23]/SCS3# *Unlatched Address (ISA)/Secondary Chip Select (IDE).* This pin has two functions, depending on whether the ISA bus is active or the IDE bus is active.

When the ISA bus is active, this pins is ISA Bus unlatched address bit 23 for 16-bit devices. When ISA bus is accessed by any cycle initiated from PCI bus, this pin is in output mode. When an ISA bus master owns the bus, this pins is in input mode.

When the IDE bus is active, this signals is used as the active high secondary slave IDE chip select signal. This signal is to be externally NANDed with the ISAOE# signal before driving the IDE devices to guarantee it is active only when ISA bus is idle.

LA[22]/SCS1# *Unlatched Address (ISA)/Secondary Chip Select (IDE)*

This pin has two functions, depending on whether the ISA bus is active or the IDE bus is active.

When the ISA bus is active, this pins is ISA Bus unlatched address bit 22 for 16-bit devices. When ISA bus is accessed by any cycle initiated from PCI bus, this pin is in output mode. When an ISA bus master owns the bus, this pins is in input mode.

When the IDE bus is active, this signals is used as the active high secondary slave IDE chip select signal. This signal is to be externally ANDed with the ISAOE# signal before driving the IDE devices to guarantee it is active only when ISA bus is idle.

LA[21]/PCS3# *Unlatched Address (ISA)/Primary Chip Select (IDE).* This pin has two functions, depending on whether the ISA bus is active or the IDE bus is active.

When the ISA bus is active, this pins is ISA Bus unlatched address bit 21 for 16-bit devices. When ISA bus is accessed by any cycle initiated from PCI bus, this pin is in output mode. When an ISA-bus master owns the bus, this pins is in input mode.

When the IDE bus is active, this signals is used as the active high primary slave IDE chip select signal. This signal is to be externally NANDed with the ISAOE# signal before driving the IDE devices to guarantee it is active only when ISA bus is idle.

LA[20]/PCS1# *Unlatched Address (ISA)/Primary Chip Select (IDE).* This pin has two functions, depending on whether the ISA bus is active or the IDE bus is active.

When the ISA bus is active, this pins is ISA Bus unlatched address bit 20 for 16-bit devices. When ISA bus is accessed by any cycle initiated from PCI bus, this pin is in output mode. When an ISA bus master owns the bus, this pins is in input mode.

When the IDE bus is active, this signals is used as the active high primary slave IDE chip select signal. This signal is to be externally NANDed with the ISAOE# signal before driving the IDE devices to guarantee it is active only when ISA bus is idle.

LA[19:17]/DA[2:0] *Unlatched Address (ISA)/Address (IDE).* These pins are multi-function pins. They are used as the ISA bus unlatched address bits [19:17] for ISA bus or the three address bits for the IDE bus devices.

When used by the ISA bus, these pins are ISA Bus unlatched address bits 19-17 on 16-bit devices. When ISA bus is accessed by any cycle initiated from the PCI bus, these pins are in output mode. When an ISA bus master owns the bus, these pins are tristated.

For IDE devices, these signals are used as the DA[2:0] and are connected to DA[2:0] of IDE devices directly or through a buffer. If the toggling of signals are to be masked during ISA bus cycles, they can be externally ORed before being connected to the IDE devices.

SA[19:8]/DD[11:0] *Unlatched Address (ISA)/Data Bus (IDE).* These are multifunction pins. When the ISA bus is active, they are used as the ISA bus system address bits 19-8. When the IDE bus is active, they serve as IDE signals DD[11:0].

These pins are used as an input when an ISA bus master owns the bus and are outputs at all other times.

IDE devices are connected to SA[19:8] directlyand ISA bus is connected to these pins through two LS245 transceivers. The $\overline{\text{OE}}$ of the transceivers are connected to ISAOE# and DIR is connected to MASTER#. A bus signals of the transceivers are connected to CPC and IDE DD bus and B bus signals are connected to ISA SA bus.

DD[15:12] *Databus (IDE).* The high 4 bits of the IDE databus are combined with several of the X-bus lines. Refer to the following section for X-bus pins for further information.

SA[7:0] *ISA Bus address bits [7:0].*These are the 8 low bits of the system address bus of ISA on 8-bit slot. These pins are used as an input when an ISA bus master owns the bus and are outputs at all other times.

SD[15:0] *I/O Data Bus (ISA).* These pins are the external databus to the ISA bus.

3.2.7 ISA/IDE COMBINED CONTROL

IOCHRDY/DIORDY *Channel Ready (ISA)/Busy/ Ready (IDE).* This is a multi-function pin. When the ISA bus is active, this pin is IOCHRDY. When the IDE bus is active, this serves as IDE signal DIORDY.
IOCHRDY is the IO channel ready signal of the ISA bus and is driven as an output in response to an ISA master cycle targeted to the host bus or an internal register of the STPC Client. The STPC Client monitors this signal as an input when performing an ISA cycle on behalf of the host CPU, DMA master or refresh.
ISA masters which do not monitor IOCHRDY are not guaranteed to work with the STPC Client since the access to the system memory can be considerably delayed due to CRT refresh or a write back cycle.

3.2.8 ISA CONTROL

ALE *Address Latch Enable.* This is the address latch enable output of the ISA bus and is asserted by the STPC Client to indicate that LA23-17, SA19-0, AEN and SBHE# signals are valid. The ALE is driven high during refresh, DMA master or an ISA master cycles by the STPC Client.
ALE is driven low after reset.

BHE# *System Bus High Enable.*This signal, when asserted, indicates that a data byte is being transferred on SD15-8 lines. It is used as an input when an ISA master owns the bus and is an output at all other times.

MEMR# *Memory Read.* This is the memory read command signal of the ISA bus. It is used as an input when an ISA master owns the bus and is an output at all other times.
The MEMR# signal is active during refresh.

MEMW# *Memory Write.* This is the memory write command signal of the ISA bus. It is used as an input when an ISA master owns the bus and is an output at all other times.

SMEMR# *System Memory Read.* The STPC Client generates SMEMR# signal of the ISA bus only when the address is below one megabyte or the cycle is a refresh cycle.

SMEMW# *System Memory Write.* The STPC Client generates SMEMW# signal of the ISA bus only when the address is below one megabyte.

IOR# *I/O Read.* This is the IO read command signal of the ISA bus. It is an input when an ISA master owns the bus and is an output at all other times.

IOW# *I/O Write.* This is the IO write command signal of the ISA bus. It is an input when an ISA master owns the bus and is an output at all other times.

MASTER# *Add On Card Owns Bus.*This signal is active when an ISA device has been granted bus ownership.

MCS16# *Memory Chip Select16.* This is the decode of LA23-17 address pins of the ISA address bus without any qualification of the command signal lines. MCS16# is always an input. The STPC Client ignores this signal during IO and refresh cycles.

IOCS16# *IO Chip Select16.* This signal is the decode of SA15-0 address pins of the ISA address bus without any qualification of the command signals. The STPC Client does not drive IOCS16# (similar to PC-AT design). An ISA master access to an internal register of the STPC Client is executed as an extended 8-bit IO cycle.

REF# *Refresh Cycle.*This is the refresh command signal of the ISA bus. It is driven as an output when the STPC Client performs a refresh cycle on the ISA bus. It is used as an input when an ISA master owns the bus and is used to trigger a refresh cycle.
The STPC Client performs a pseudo hidden refresh. It requests the host bus for two host clocks to drive the refresh address and capture it in external buffers. The host bus is then relinquished while the refresh cycle continues on the ISA bus.

AEN *Address Enable.* Address Enable is enabled when the DMA controller is the bus owner to indicate that a DMA transfer will occur. The enabling of the signal indicates to IO devices to ignore the IOR#/IOW# signal during DMA transfers.

IOCHCK# *IO Channel Check.* IO Channel Check is enabled by any ISA device to signal an error condition that can not be corrected. NMI signal becomes active upon seeing IOCHCK# active if the corresponding bit in Port B is enabled.

ISAOE# *Bidirectional OE Control.* This signal controls the $\overline{OE}$ signal of the external transceiver that connects the IDE DD bus and ISA SA bus.

GPIOCS# *I/O General Purpose Chip Select 1.* This output signal is used by the external latch on ISA bus to latch the data on the SD[7:0] bus. The latch can be use by PMU unit to control the external peripheral devices to power down or any other desired function.
This pin is also serves as a strap input during reset.

3.2.9 IDE CONTROL

PIRQ *Primary Interrupt Request.* Interrupt request from primary IDE channel.

SIRQ *Secondary Interrupt Request.* Interrupt request from secondary IDE channel.

PDRQ *Primary DMA Request.* DMA request from primary IDE channel.

SDRQ *Secondary DMA Request.* DMA request from secondary IDE channel.

PDACK# *Primary DMA Acknowledge.* DMA acknoledge to primary IDE channel.

SDACK# *Secondary DMA Acknowledge.* DMA acknoledge to secondary IDE channel.

PIOR# *Primary I/O Read.* Primary channel read. Active low output.

PIOW# *Primary I/O Write*. Primary channel write. Active low output.

SIOR# *Secondary I/O Read* Secondary channel read. Active low output.

SIOW# *Secondary I/O Write* Secondary channel write. Active low output.

3.2.10 X-Bus Interface pins / IDE Data

RMRTCCS# / DD[15] *ROM/Real Time clock chip select.* This pin is a multi-function pin. When ISAOE# is active, this signal is used as RMRTCCS#. This signal is asserted if a ROM access is decoded during a memory cycle. It should be combined with MEMR# or MEMW# signals to properly access the ROM. During a IO cycle, this signal is asserted if access to the Real Time Clock (RTC) is decoded. It should be combined with IOR or IOW# signals to properly access the real time clock.
When ISAOE# is inactive, this signal is used as IDE DD[15] signal.
This signal must be ORed externally with ISAOE# and is then connected to ROM and RTC. An LS244 or equivalent function can be used if OE# is connected to ISAOE# and the output is provided with a weak pull-up resistor.

KBCS# / DD[14] *Keyboard Chip Select.* This pin is a multi-function pin. When ISAOE# is active, this signal is used as KBCS#. This signal is asserted if a keyboard access is decoded during a I/O cycle.
When ISAOE# is inactive, this signal is used as IDE DD[14] signal.
This signal must be ORed externally with ISAOE# and is then connected to keyboard. An LS244 or equivalent function can be used if OE# is connected to ISAOE# and the output is provided with a weak pull-up resistor.

RTCRW# / DD[13] *Real Time Clock R$\overline{W}$.* This pin is a multi-function pin. When ISAOE# is active, this signal is used as RTCRW#. This signal is asserted for any I/O write to port 71H.
When ISAOE# is inactive, this signal is used as IDE DD[13] signal.
This signal must be ORed externally with ISAOE# and then connected to the RTC. An LS244 or equivalent function can be used if OE is connected to ISAOE# and the output is provided with a weak pull-up resistor.

RTCDS# / DD[12] *Real Time Clock DS*. This pin is a multi-function pin. When ISAOE# is active, this signal is used as RTCDS. This signal is asserted for any I/O read to port 71H.
When ISAOE# is inactive, this signal is used as IDE DD[12] signal.
This signal must be ORed externally with ISAOE# and is then connected to RTC. An LS244 or equivalent function can be used if OE# is connected to ISAOE# and the output is provided with a weak pull-up resistor.

3.2.11 IPC

IRQ_MUX[3:0] *Multiplexed Interrupt Request.* These are the ISA bus interrupt signals. They are to be encoded before connection to the STPC Client using ISACLK and ISACLKX2 as the input selection strobes.
Note that IRQ8B, which by convention is connected to the RTC, is inverted before being sent to the interrupt controller, so that it may be connected directly to the $\overline{\text{IRQ}}$ pin of the RTC.

PCI_INT[3:0] *PCI Interrupt Request.* These are the PCI bus interrupt signals. They are to be encoded before connection to the STPC Client using ISACLK and ISACLKX2 as the input selection strobes.

DREQ_MUX[1:0] *ISA Bus Multiplexed DMA Request.* These are the ISA bus DMA request signals. They are to be encoded before connection to the STPC Client using ISACLK and ISACLKX2 as the input selection strobes.

DACK_ENC[2:0] *DMA Acknowledge.* These are the ISA bus DMA acknowledge signals. They are encoded by the STPC Client before output and should be decoded externally using ISACLK and ISACLKX2 as the control strobes.

TC *ISA Terminal Count.* This is the terminal count output of the DMA controller and is connected to the TC line of the ISA bus. It is asserted during the last DMA transfer, when the byte count expires.

3.2.12 Monitor Interface

RED, GREEN, BLUE *RGB Video Outputs.* These are the 3 analog color outputs from the RAMDACs

VSYNC *Vertical Synchronisation Pulse.* This is the vertical synchronization signal from the VGA controller.

HSYNC *Horizontal Synchronisation Pulse.* This is the horizontal synchronization signal from the VGA controller.

VREF_DAC *DAC Voltage reference.* An external voltage reference is connected to this pin to bias the DAC.

RSET *Resistor Current Set.* This is reference current input to the RAMDAC is used to set the full-scale output of the RAMDAC.

COMP *Compensation.* This is the RAMDAC compensation pin. Normally, an external capacitor (typically 10nF) is connected between this pin and V_{DD} to damp oscillations.

DDC[1:0] *Direct Data Channel Serial Link.* These bidirectional pins are connected to CRTC register 3Fh to implement DDC capabilities. They conform to I^2C electrical specifications, they have open-collector output drivers which are internally connected to V_{DD} through pull-up resistors.

3.2.13 MISCELLANEOUS

ST[6:0] *Reserved*. The pins are reserved for Test and Miscellaneous functions)

CLKDEL[2:0] *Reserved*. The pins are reserved for Test and Miscellaneous functions)

Table 3. Pinout.

Pin #	Pin name
AF3	PWERGD
AF15	XTALI
AE16	XTALO
G23	HCLK
F25	DEV_CLK
AC5	GCLK2X
AD5	DCLK
AF5	DCLK_DIR
AD15	MA[0]
AF16	MA[1]
AC15	MA[2]
AE17	MA[3]
AD16	MA[4]
AF17	MA[5]
AC17	MA[6]
AE18	MA[7]
AD17	MA[8]
AF18	MA[9]
AE19	MA[10]
AF19	MA[11]
AD18	RAS#[0]
AE20	RAS#[1]
AC19	RAS#[2]
AF20	RAS#[3]
AE21	CAS#[0]
AC20	CAS#[1]
AF21	CAS#[2]
AD20	CAS#[3]
AE22	CAS#[4]
AF22	CAS#[5]
AD21	CAS#[6]
AE23	CAS#[7]
AC22	MWE#
AF23	MD[0]
AE24	MD[1]
AF24	MD[2]
AD25	MD[3]
AC25	MD[4]
AC26	MD[5]
AB24	MD[6]
AA25	MD[7]
AA24	MD[8]
Y25	MD[9]
Y24	MD[10]
V23	MD[11]
W24	MD[12]
V26	MD[13]
V24	MD[14]
U23	MD[15]
U24	MD[16]
R26	MD[17]
P25	MD[18]
P26	MD[19]
N25	MD[20]
N26	MD[21]
M25	MD[22]
M26	MD[23]
M24	MD[24]
M23	MD[25]
L24	MD[26]
J25	MD[27]
J26	MD[28]
H26	MD[29]
G25	MD[30]
G26	MD[31]
AD22	MD[32]
AD23	MD[33]
AE26	MD[34]
AD26	MD[35]
AC24	MD[36]
AB25	MD[37]
AB26	MD[38]
Y23	MD[39]
AA26	MD[40]
Y26	MD[41]
W25	MD[42]
W26	MD[43]
V25	MD[44]
U25	MD[45]
U26	MD[46]
T25	MD[47]
R25	MD[48]
T24	MD[49]
R23	MD[50]
R24	MD[51]
N23	MD[52]
P24	MD[53]
N24	MD[54]
L25	MD[55]
L26	MD[56]
K25	MD[57]
K26	MD[58]
K24	MD[59]
H25	MD[60]
J24	MD[61]
H23	MD[62]
H24	MD[63]
F24	PCI_CLKI
D25	PCI_CLKO
A20	AD[0]
C20	AD[1]
B19	AD[2]
A19	AD[3]
C19	AD[4]
B18	AD[5]
A18	AD[6]
B17	AD[7]
C18	AD[8]
A17	AD[9]
D17	AD[10]
B16	AD[11]
C17	AD[12]
B15	AD[13]
A15	AD[14]
C16	AD[15]
D15	AD[16]
A14	AD[17]
C15	AD[18]
B13	AD[19]
D13	AD[20]
A13	AD[21]
C14	AD[22]
C13	AD[23]
A12	AD[24]
B11	AD[25]
C12	AD[26]
A11	AD[27]
D12	AD[28]
B10	AD[29]
C11	AD[30]
A10	AD[31]
D10	CBE[0]
C10	CBE[1]
A9	CBE[2]
B8	CBE[3]
A8	FRAME#
B7	TRDY#
D8	IRDY#
A7	STOP#
C8	DEVSEL#
B6	PAR
D7	SERR#
A6	LOCK#
C21	PCI_REQ#[0]
A21	PCI_REQ#[1]
B20	PCI_REQ#[2]

Pin #	Pin name
C22	PCI_GNT#[0]
B21	PCI_GNT#[1]
D20	PCI_GNT#[2]
D24	PCI_INT[0]
C26	PCI_INT[1]
A25	PCI_INT[2]
B24	PCI_INT[3]
F2	LA[17]/DA[0]
G4	LA[18]/DA[1]
F3	LA[19]/DA[2]
F1	LA[20]/PCS1#
G2	LA[21]/PCS3#
G3	LA[22]/SCS1#
H2	LA[23]/SCS3#
J4	SA[0]
H1	SA[1]
H3	SA[2]
J2	SA[3]
J1	SA[4]
K2	SA[5]
J3	SA[6]
K1	SA[7]
K4	SA[8]/DD[0]
L2	SA[9]/DD[1]
K3	SA[10]/DD[2]
L1	SA[11]/DD[3]
M2	SA[12] / DD[4]
M1	SA[13] / DD[5]
L3	SA[14] / DD[6]
N2	SA[15] / DD[7]
M4	SA[16] / DD[8]
N1	SA[17] / DD[9]
M3	SA[18] / DD[10]
P4	SA[19] / DD[11]
P3	RTCDS / DD[12]
R2	RTCRW# / DD[13]
N3	KBCS# / DD[14]
P1	RMRTCCS# / DD[15]
R1	SD[0]
T2	SD[1]
R3	SD[2]
T1	SD[3]
R4	SD[4]
U2	SD[5]
T3	SD[6]
U1	SD[7]
U4	SD[8]
V2	SD[9]

Pin #	Pin name
U3	SD[10]
V1	SD[11]
W2	SD[12]
W1	SD[13]
V3	SD[14]
Y2	SD[15]
AE4	SYSRSETOx
AD4	ISA_CLK
AE5	ISA_CLK2X
C6	OSC14M
W3	ALE
AA2	BHE#
Y4	MEMR#
AA1	MEMW#
Y3	SMEMR#
AB2	SMEMW#
AA3	IOR#
AC2	IOW#
AB4	MASTER#
AC1	MCS16#
AB3	IOCS16#
AD2	REF#
AC3	AEN
AD1	IOCHCK#
AF2	ISAOE#
AE3	GPIOCS#
Y1	IOCHRDY
B1	PIRQ
C2	SIRQ
C1	PDRQ
D2	SDRQ
D3	PDACK#
D1	SDACK#
E2	PIOR#
E4	PIOW#
E3	SIOR#
E1	SIOW#
E23	IRQ_MUX[0]
D26	IRQ_MUX[1]
E24	IRQ_MUX[2]
C25	IRQ_MUX[3]
A24	DREQ_MUX[0]
B23	DREQ_MUX[1]
C23	DACK_ENC[0]
A23	DACK_ENC[1]
B22	DACK_ENC[2]
D22	TC

Pin #	Pin name
AE6	RED
AD6	GREEN
AF6	BLUE
AE9	VSYNC
AF9	HSYNC
AD7	VREF
AE8	RSET
AC9	COMP
AF8	DDC[1]
AD8	DDC[0]
AD14	VIDEO_CLK
AE13	VIDEO_D[0]
AC12	VIDEO_D[1]
AD12	VIDEO_D[2]
AE14	VIDEO_D[3]
AC14	VIDEO_D[4]
AF14	VIDEO_D[5]
AD13	VIDEO_D[6]
AE15	VIDEO_D[7]
AF10	VTV_YUV[0]
AC10	VTV_YUV[1]
AE11	VTV_YUV[2]
AD10	VTV_YUV[3]
AF11	VTV_YUV[4]
AE12	VTV_YUV[5]
AF12	VTV_YUV[6]
AD11	VTV_YUV[7]
AE10	VTV_HSYNC
AD9	VTV_BT#
B4	ST[0]
D5	ST[1]
A4	ST[2]
C5	ST[3]
B3	ST[4]
C4	ST[5]
A3	ST[6]
C7	CLKDEL[0]
B5	CLKDEL[1]
A5	CLKDEL[2]
AC7	VDD_DAC1
AF4	VDD_DAC2
W4	VDD_GCLK_PLL
AB1	VDD_DCLK_PLL
F26	VDD_HCLK_PLL
G24	VDD_DEVCLK_PLL

Pin #	Pin name
AF13	VDDE
AD19	VDDE
A16	VDD5
B12	VDD5
B9	VDD5
D18	VDD5
A22	VDD
B14	VDD
C9	VDD
D6	VDD
D11	VDD
D16	VDD
D21	VDD
F4	VDD
F23	VDD
G1	VDD
K23	VDD
L4	VDD
L23	VDD
P2	VDD
T4	VDD
T23	VDD
T26	VDD
AA4	VDD
AA23	VDD
AB23	VDD
AC6	VDD
AC11	VDD
AC16	VDD
AC21	VDD
AE7	VSS_DAC1
AF7	VSS_DAC2
E25	VSS_DLL
E26	VSS_DLL
A1:2	VSS
A26	VSS
B2	VSS
B25:26	VSS
C3	VSS
C24	VSS
D4	VSS
D9	VSS
D14	VSS
D19	VSS
D23	VSS
H4	VSS
J23	VSS
L11:16	VSS

Pin #	Pin name
M11:16	VSS
N4	VSS
N11:16	VSS
P11:16	VSS
P23	VSS
R11:16	VSS
T11:16	VSS
V4	VSS
W23	VSS
AC4	VSS
AC8	VSS
AC13	VSS
AC18	VSS
AC23	VSS
AD3	VSS
AD24	VSS
AE1:2	VSS
AE25	VSS
AF1	VSS
AF25	VSS
AF26	VSS

4 MEMORY

4.1 INTRODUCTION

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

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

4.2 MEMORY CONTROLLER

The STPC handles the memory data (DATA) bus directly, controlling from 2 to 128 MBytes. This main memory is supported using 4 SIMM sockets (Banks 1 to 4) which can be populated with either single or double sided 36-bit (4 bit parity) or 32-bit data SIMMs. Parity is not supported. Four DRAM densities are supported: 1M (256KX4), 4M (512KX4), 16M (4MX4) and 64M (16MX4). Bank 0 SIMMs must be used in pairs (64 bits wide) since the Graphics Controller does not support 32 bit banks. Banks 1,2 and 3 can be 32 or 64 bits wide.

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

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

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

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

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

4.3 ADDRESS MAP

4.3.1 Memory Address Map:

4.3.1.1 00000000h-0009FFFFh (640K)

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

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

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

4.3.1.2 000A0000h-000BFFFFh (128K)

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

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

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

This segment is never cacheable.

4.3.1.3 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 mapping. 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.

4.3.1.4 000C4000h-000C7FFFh (16K)

This range has the same characteristics as that of 000C0000h-000C3FFFh segment as described above. 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.

4.3.1.5 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.

4.3.1.6 000CC000h-000CFFFFh (16K)

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

4.3.1.7 000D0000h-000DFFFFh (64K)

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

4.3.1.8 000E0000h-000EFFFFh (64K)

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

4.3.1.9 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 the ISA bus will always result in ROMCS# assertion. The cacheability of this address segment is controlled via Shadow Control register 3.

4.3.1.10 00100000h (1M) - Top of addressable DRAM memory

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

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

4.3.1.11 Top of addressable DRAM memory - FFFEFFFFh (4G-64K)

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

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

PCI master access in this range if not claimed by a PCI slave will be forwarded to the ISA bus.

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

4.3.1.12 FFFF0000 - FFFFFFFFh (4G-64K)

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

4.3.1.13 Extended Graphics segment

A 16M segment of memory anywhere between Top of addressable DRAM memory and 256M can optionally be 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 accesses in this region are absorbed by the STPC and are either consumed internally or initiate a frame buffer memory access.

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

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

This address segment is always considered non-cacheable.

4.3.1.14 Memory Hole

The Memory Hole register allows creating 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 the same as the a cycle above the addressable memory range as described above.

4.3.1.15 SMM Memory

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

4.3.1.16 Addressable DRAM Memory

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

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

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

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

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

The addressable DRAM memory =

TOPM - TOGM + MHOLE_SIZE + REMAP_SIZE

4.3.1.17 CPU address to DRAM address mapping:

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

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

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

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

For example:

Total populated DRAM = 4 Mbyte

Frame buffer size = 256 Kbyte

Memory hole size = 1 Mbyte

Memory hole starting address = 200000h

Shadow feature for D0000h to EFFFFh = disabled

The total CPU memory = 4 Mbyte - 256 Kbyte + 1 Mbyte + 128 Kbyte = 4 Mbyte plus 896 Kbyte

Since the frame buffer is 256 Kbyte, the system memory is reduced by 256 Kbyte and becomes 3 Mbyte plus 768 Kbyte. Since a 1 Mbyte memory hole exists, the CPU address space is increased by 1 Mbyte and becomes 4 Mbyte plus 768 Kbyte. The CPU address between 3 Mbyte plus 768 Kbyte and 1 Mbyte 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 Kbyte to make use of this DRAM space that no device accesses. The total CPU memory then becomes 4 Mbyte plus 896 Kbyte.

4.4 IO Address Map:

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

These register occupy the following map in the IO space
.

IO address	Description	Notes
0000h-000Fh	8237 DMA controller 1 registers.	1
0010h-001Ch	Local Bus configuration registers	
0020h-0021h	8259 Interrupt controller 1 registers.	
0022h	STPC specific configuration registers index port	
0023h	STPC specific configuration registers data port	
0040h-0043h	8254 Timer/Counter registers.	1
0060h-0064h	Keyboard shadow registers.	1
0070h-0071h	NMI Mask control registers.	1
0080h-008Fh	DMA Page registers.	
0094h	Mother-board VGA enable.	2
00A0h-00A1h	8259 Interrupt controller 2 registers.	1
00B1h	ISA standard Port B.	1
00C0h-00DFh	8237 DMA controller 2 registers.	1
0102h	VGA setup register.	
03B4h,03B5h,03BAh	VGA registers.	
03D4h,03D5h,03DAh		
03C0h-03CFh		
0CF8h	PCI configuration Address register.	
0CFCh-0CFFh	PCI configuration Data register.	
46E8h	VGA add-in mode enable register.	2

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 STPC is strapped to look like a mother-board VGA.

4.4.1 PCI Configuration Address Map:

The STPC occupies Device number 0 slot on the PCI bus and implements a number of registers in PCI configuration address space. These registers occupy the following map:

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

4.4.2 Cache related registers

4.4.2.1 Cache architecture register 0 C.I. 20

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.

4.4.2.2 Cache architecture register 1 C.I.21

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	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 always work.

Bit 3	Bit 2	Start next read...
0	0	only after complete 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 to an address different to posted writes

Bit 0 *Reserved*.

This register defaults to 00h at reset.

4.4.2.3 Cache architecture register 2 C.I. 22

Bit 7 **Local Bus Disable.**

Bit 7	Local Bus
0	No Local Bus
1	Local Bus enabled, decode CPU addresses

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 async SRAMs only. Must be ‘0’ for sync 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.

4.4.3 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).

4.4.3.1 Memory Hole Control register - C.I. 24

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. This memory hole is also non-cacheable.

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 starting address of the memory hole. Starting address of the memory hole must be aligned to the size of the hole.

This register defaults to 00h at reset.

4.4.3.2 Shadow Control register 0 - C. I. 25

This 8-bit register controls the read/write attributes of the memory located at C0000h-CFFFFh. Each 16k of the whole 64k is controlled by 2 bits, one for read and one for write. 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/write to this area are directed to the system memory. Else the cycles is forwarded to the expansion busses.

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 C0000h-C3FFFFh.** 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.

4.4.3.3 Shadow Control register 1 - C. I. 26

Similar to Shadow Control Register 0, this 8-bit register controls the read/write attributes of the memory located at D0000h-DFFFFh. This entire 64K segment has the cacheability bit hardwired to ‘0’ (disabled).

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.

4.4.3.4 Shadow Control register 2 - C. I. 27

Similar to Shadow Control Register 0, this 8-bit register controls the read/write attributes of the memory located at E0000h-EFFFFh. This entire 64K segment has the cacheability bit hardwired to ‘0’ (disabled).

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.

4.4.3.5 Shadow Control register 3 - C. I. 28

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

Bit 7 **SMRAM 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 STPC allows for 128K bytes of SMRAM. Physically this memory is located at the system memory behind the above 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 go 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 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 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.

4.4.3.6 VGA decode register - C.I. 29

This 8-bit register controls address decode for the internal VGA. It is presented here in the sequence of the Configuation Index. The function is as follows:

Bits 7-6 *Reserved*

Bit 5 stop g-clock.

Bit 4 stop p-clock.

Bit 3 stop d-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.

4.4.4 DRAM CONTROLLER REGISTERS

The STPC manages 4 SIMM sockets which can be populated with either single or double sided 36-bit (4 bit parity) or 32-bit data SIMMs. Four DRAM densities are supported: 1M (256KX4), 4M (1MX4), 16M (4MX4) and 64M (16MX1). Although system DRAM data bus is 64-bit wide, 32-bit DRAM bank is also supported by not populating the upper Dword SIMM module for that particular bank. However Bank 0 must always be populated to 64-bit when the integrated Graphics Controller is enabled.

Index 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.

4.4.4.1 DRAM Bank 0 register - C. I. 30

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

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

This register defaults to 07h.

Example 1:

SIMM0 = 4MB
SIMM1 = 4MB

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

Bank 1, 2, 3 Top Address = 07h

Example 2:

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

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

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

Bank 2, 3 Top Address = 3Fh

Example 3:

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

SIMM2 = 16MB
SIMM3 = 16MB

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

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

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

Bank 3 Top Address = 5Fh

4.4.4.2 DRAM Bank 1 register - C. I. 31

This register controls the top address of DRAM bank 1.

4.4.4.3 DRAM Bank 2 register - C.I. 32

This register controls the top address of DRAM bank 2.

4.4.4.4 DRAM Bank 3 register - C. I. 33

This register controls the top address of DRAM bank 3.

4.4.4.5 Memory Bank Width - C. I. 34

Each memory bank can have one or two 32 bit SIMMs 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 Bank Width
0	64 bits (default)
1	32 bits

This register defaults to 00h at reset.

4.4.4.6 Bank 0 Timing Parameter - C. I. 35

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

Bit 7 **Main RAS Active.** This bit controls if RAS is kept active after the current DRAM access. It applies to all banks in main memory space.

0 = keep RAS# active

1 = deassert RAS#

Bits 6 *Reserved*

Bits 5-4 **Bank 0 RAS Precharge.** RAS precharge timing for Bank 0

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

Bits 3-2 **Bank 0 RAS/CAS Delay.** RAS to CAS delay timing for Bank 0

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

Bits 1-0 Bank 0 CAS Pulse Width. These bits control the CAS pulse width

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

This register defaults to 80h.

4.4.4.7 Bank 1 Timing Parameter - C. I. 38

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

Bits 7-6 *Reserved*

Bits 5-4 **Bank 1 RAS Precharge.** These bits control the RAS precharge timing.

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

Bits 3-2 **Bank 1 RAS/CAS Delay.** These bits control the RAS to CAS delay timing.

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

Bits 1-0 **Bank 1 CAS Pulse Width.** These bits control the CAS pulse width

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

This register defaults to 00h.

4.4.5 Bank 2 Timing Parameter - C. I. 3A

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

Bits 7-6 *Reserved*

Bits 5-4 **Bank 2 RAS Precharge.** These bits control the RAS precharge timing.

01	Reserved
10	RAS# precharge time 2 cycles
11	RAS# precharge time 3 cycles
00	RAS# precharge time 4 cycles

Bits 3-2 **Bank 2 RAS/CAS Delay.** These bits control the RAS to CAS delay timing.

01	Reserved
10	RAS# to CAS# delay 2 cycles
11	RAS# to CAS# delay 3 cycles
00	RAS# to CAS# delay 4 cycles

Bits 1-0 **Bank 2 CAS Pulse Width.** These bits control the CAS pulse width

01	CAS# low pulse width 1 cycles
10	CAS# low pulse width 2 cycles
11	CAS# low pulse width 3 cycles
00	CAS# low pulse width 4 cycles

This register defaults to 00h.

4.4.6 Bank 3 Timing Parameter - C. I. 3B

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

Bits 7-6 *Reserved*

Bits 5-4 **Bank 3 RAS Precharge.** These bits control the RAS precharge timing.

01	Reserved
10	RAS# precharge time 2 cycles
11	RAS# precharge time 3 cycles
00	RAS# precharge time 4 cycles

Bits 3-2 **Bank 3 RAS/CAS Delay.** These bits control the RAS to CAS delay timing.

01	Reserved
10	RAS# to CAS# delay 2 cycles
11	RAS# to CAS# delay 3 cycles
00	RAS# to CAS# delay 4 cycles

Bits 1-0 **Bank 3 CAS Pulse Width.** These bits control the CAS pulse width

01	CAS# low pulse width 1 cycles
10	CAS# low pulse width 2 cycles
11	CAS# low pulse width 3 cycles
00	CAS# low pulse width 4 cycles

This register defaults to 00h.

4.4.6.1 Graphics memory size register - C. I. 36

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

Bit 7 **Graphics RAS Active.** This bit controls if RAS is kept active after the current framebuffer DRAM 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 necessary.

This register defaults to 04h.

4.4.6.2 Memory Type register - C. I. 37

Bits 7-6 Bank 3 type code

Bits 5-4 Bank 2 type code

Bits 3-2 Bank 1 type code

Bits 1-0 Bank 0 type code

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

This register defaults to 00h after reset.

4.4.6.3 DRAM Refresh - C. I. 39

Refresh disable bit. This register also contains the number of host clocks for the DRAM refresh interval.

Bit 7 **Refresh Enable.** This bit must be programmed to ‘0’ for normal operation

Bits 6-0 **Refresh Cycle.** (HCLK frequency in MHz * 15.6us) >> 4

Examples: (rounded down to nearest integer)

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

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

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

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

This register defaults to 30h.

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

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

4.5 DRAM INTERFACE

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

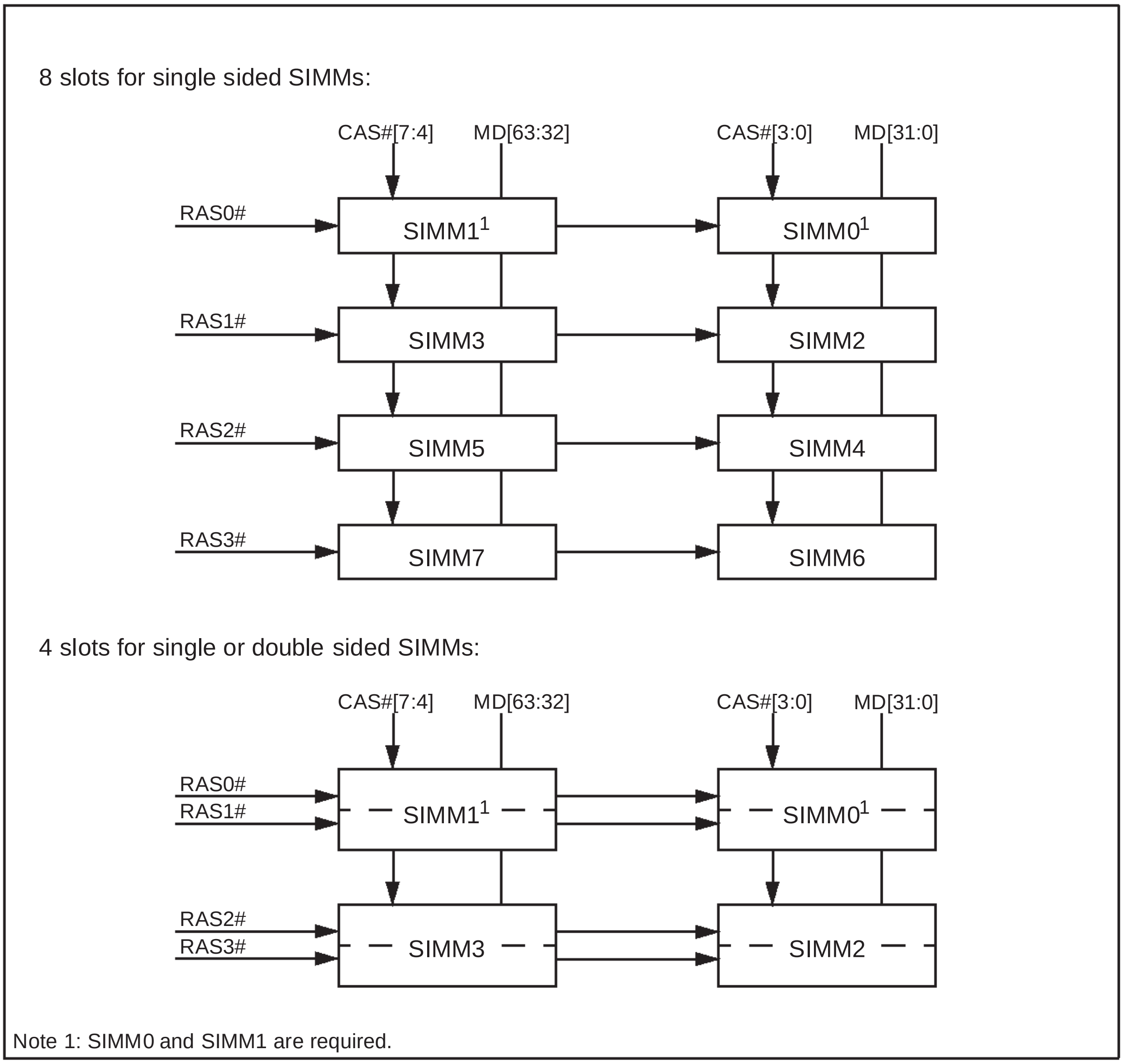
The following SIMMs are supported:

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

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

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

Figure 8. DRAM Organisation



Note 1: SIMM0 and SIMM1 are required.

The STPC DRAM interface consists of a host clock domain DRAM controller and a graphics clock domain DRAM controller. The arbitration between these two different clock domain controllers are described in section 5.15, DRAM arbitration. The host clock domain DRAM controller processes host initiated read and write cycles as well as DRAM refresh cycles. Graphics, CRTC and Video Scaler read and write cycles are processed by the graphics domain DRAM controller (see section 5.7).

4.5.1 Fast Page Mode (FPM) DRAM

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

4.5.2 EDO DRAM

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

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

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

4.5.3 Host Address to MA bus Mapping

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

The bank attributes can be retrieved from a lookup table to select the final DRAM row and column address mappings. (Table 4)

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

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

Bank Height is determined by subtracting adjacent bank top addresses to get the bank size, then dividing by 4 bytes for a 32-bit bank or dividing by 8 bytes for a 64-bit bank. The bank width is important to distinguish between two 8MB SIMMs with 2Mx8 DRAMs from a single 16MB SIMM with 4Mx4 DRAMs.

Table 4. Host Address to MA Bus Mapping

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

A = Corresponding remapped host address bit

MA = memory address

X = don’t care

DRAM row address

4.5.4 DRAM Module Presence and Type Detect

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

Density	Configuration	PD0	PD1	PD4	row adrs bits	col adrs bits
1MB	256Kx32/36	S	O	O	9	9
2MB	2x256Kx32/36	O	S	O	9	9
2MB	512Kx32/36	O	S	S	10	9
4MB	2x512Kx32/36	S	S	S	10	9
4MB	1Mx32/36	S	S	O	10	10
8MB	2x1Mx32/36	O	O	O	10	10
8MB	2Mx32/36	O	O	S	11	10
16MB	2x2Mx32/36	S	O	S	11	10
16MB	4Mx32/36	S	O	O	11/12	11/10
32MB	2x4Mx32/36	O	S	O	11/12	11/10
32MB	8Mx32/36	O	S	S	12	11
64MB	2x8Mx32/36	S	S	S	12	11
64MB	16Mx32/36	S	S	O	12	12

S = short to Ground

O = Open circuit

The configuration codes are unique within the to 32MB density range. Software probing of memory locations is necessary to support a wider range of SIMM configurations.

Table 5. DRAM Page Mode Detect

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

Table 6. DRAM Speed Detect

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

4.6 DRAM ARBITRATION:

Following agents compete for the system DRAM memory:

– CPU

– PCI masters

– ISA masters

– Graphics engine

– CRT controller

– Video Output

– Video Input Port

– Refresh controller

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

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

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

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

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

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

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

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

5 PCI CONTROLLER

5.1 Introduction

The STPC internally translates appropriate host bus IO and Memory cycles onto the PCI bus. The PCI bus is the main data communication link to the STPC chip. The device contains a PCI arbitration function for three external PCI devices.

Two PCI devices are present internal to the STPC, the East Bridge and the West Bridge. The East Bridge translates appropriate host bus IO and Memory cycles onto the PCI bus. It is a PCI bus agent (host bridge class) and fully complies with PCI specification 2.1. The East Bridge also implements the PCI mandatory header registers in Type 0 PCI configuration space for easy porting of PCI-aware system BIOS.

The West Bridge controller responds to PCI configuration read and write transactions as a PCI bus agent (expansion bridge class). The West bridge implements two PCI functions: Function 0, PCI to ISA bridge and Function 1, Bus master IDE controller.

5.2 THE EAST BRIDGE

The STPC East Bridge PCI controller 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 IO cycles in the address range CFCh-CFFh to configuration cycles on the PCI bus. The PCI controller is assigned the Device Number Bh, which corresponds to IDSEL on AD11 signal. PCI configuration registers of the controller are accessible by the Type 0 PCI configuration cycles generated at Device number Bh.

5.2.1 EAST BRIDGE PCI RELATED REGISTERS

5.2.1.1 Configuration Address register IO CF8h

This is a 32-bit register accessible only via double-word IO read and write cycles. A non double-word read or write cycle in CF8h-CCBh range will not affect this register and will be passed on to the expansion busses.

Bit 31 **IO Configuration.** 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 address range pass through as normal IO cycles on the PCI bus.

Bits 31-24 *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 PCI Controller. A configuration cycle with bus number 0 will run as Type 0 configuration cycle. A non-zero bus number will result in Type 1 configuration cycle initiated on the PCI bus. During Type 1 configuration cycle, 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 appropriate IDSEL line. The PCI Controller is assigned Device Number Bh. The PCI configuration space registers can be accessed by running a Type-0 configuration cycle with this field set to Bh. During a Type-1 configuration cycle, this field is driven on bits15-11 of the AD bus during the address phase.

Bits 10-8 **Function Number.** During a PCI configuration cycle, this field is driven on bits 10-8 of the AD bus of the PCI during the address phase.

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.

5.2.1.2 Configuration Data register IO CFCh-CFFh

A host IO cycle in this address range, if enabled via bit 31 of the Configuration Address register, is translated into a configuration cycle on the PCI bus. Otherwise if bit 31 of the Configuration Address register is ‘0’, the IO cycle is passed on as a normal IO cycle to the expansion busses. The IO access may be 1 to 4 bytes wide. The byte enables will be accurately reflected on the PCI bus.

5.2.1.3 E.B. Vendor Identification register PCI Config. Offset 0h-1h

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

Bits 15-8 These bits are hardwired to 10h.

Bits 7-0 These bits are hardwired to 4Ah.

Writes to this register have no effect.

5.2.1.4 E.B. Device Identification register PCI Config. Offset 2h-3h

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

Bits 15-8 These bits are hardwired to 02h

Bits 7-0 These bits are hardwired to 09h

Writes to this register have no effect.

5.2.1.5 E.B. PCI Command register PCI Config. Offset 4h-5h

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 East Bridge may assert SERR# upon detecting a target abort in response to an East bridge initiated PCI transaction, upon being forced to end a non-configuration space transaction with a master abort, or if a parity error on the PCI bus is detected. If this bit is set to '0', the East 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 **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.

5.2.1.6 E.B. PCI Status register PCI Config. Offset 6h-7h

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 East Bridge. Writing a ‘1’ to this bit will clear it.

Bit 13 **Signaled Master Abort.** This bit is set to a 1 when the East 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 East 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 East 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.

5.2.1.7 E.B. PCI Revision ID register PCI Config. Offset 8h

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

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

5.2.1.8 E.B. Device Class Code register PCI Config. Offset 9h-Bh

This is a 24 bit read only register implemented at configuration space offset 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 (9h) These bits are hardwired to 00h.

5.2.1.9 E. B. PCI Control register PCI Config. Offset 50h

Bit 31-22 *Reserved.* Hardwired to ‘0’.

Bit 21 **PCI to Host Read Prefetch Enable.** If this bit is set to ‘1’, all qword aligned burst reads from a PCI master addressed to the East Bridge system memory will use prefetch. If set to ‘0’, memory read cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst read attempts will be disconnected on the PCI bus.

Bit 20 **PCI to Host Write Posting Enable.** If this bit is set to ‘1’, all burst writes from a PCI master addressed to the East Bridge system memory will be posted. If it is set to ‘0’, all memory write cycles from PCI to host are allowed to complete before the PCI cycle is terminated and all burst write attempts will be disconnected on the PCI bus.

Bit 19-0 *Reserved.* Hardwired to ‘0’.

5.2.1.10 E.B. PCI Error Status register PCI Config. Offset 54h

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 error as a result of 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 error as a result of received target abort. This bit is set to ‘1’ when SERR# was asserted as a result of recieving target abort. Writing a ‘1’ will clear it.

5.3 THE WEST BRIDGE

The STPC's West Bridge (W.B.) controller responds to PCI configuration read and write transactions. The West bridge, as a PCI bus agent (expansion bridge class, fully complies with the PCI specification. The West Bridge PCI interface itself is assigned the Device Number 0Ch, which corresponds to IDSEL on AD12 signal. PCI configuration registers are accessible by the Type 0 PCI configuration cycles.

The West bridge implements two PCI functions:

- Function 0: PCI to ISA bridge

- Function 1: Bus master IDE controller

Following the PCI specification, the West bridge will responds to both function 0 and function 1 configuration cycles directed to configuration slot 12 (AD12).

West Bridge PCI Interface:

The PCI interface of the West Bridge consists of two major functional blocks; the master interface and the target interface. Transactions from IDE or ISA to PCI are handled by the PCI master interface, transactions from PCI to West Bridge resources are handled by the target interface. In addition, there is error handling logic that takes care of detecting and logging bus anomolies.

West Bridge PCI Master interface:

The IDE or ISA busses communicate with the PCI bus via the PCI master interface. Either of these may initiate a PCI transaction by setting up the address, the cycle type, the byte enables then asserting the address strobe for one PCI CLK cycle. This will cause a request to be sent to the PCI arbiter. When ISA or IDE grants is asserted, meaning that the PCI arbiter in the West Bridge is currently asserting a grant to ISA or IDE, then themaster interface will begin the PCI cycle immediately. The West Bridge master interface can initiate PCI burst transactions if requested by the IDE interface, the ISA bus cannot generate PCI burst cycles.

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, shutdown and halt respectively, are snooped and passed on to the ISA bus. Byte enables and address bits 0 and 4 are passed from PCI to ISA as well to support decode of the special cycle by ISA. All other PCI special cycles are ignored by the West Bridge.

PCI Cycle Termination:

PCI cycles initiated by the IDE controller can be terminated in one of three ways. Assertion of an IDE ready condition indicates to the IDE controller that the PCI cycle finished correctly and, in the case of a read, that read data is available on the read data bus. If an IDE stop condition is asserted it means at least one data phase of a burst was completed but the current PCI cycle was terminated with a retry. If the first data phase ends in retry then the PCI controller takes care of re-running the transaction until it completes successfully. If an IDE error condition is asserted it indicates that the cycle was terminated on the PCI bus with either a master abort (meaning that no device responded to the PCI cycle), or target abort (meaning that the target aborted the PCI cycle) and the cycle should not be repeated by the IDE.

PCI cycles initiated by the ISA controller can be terminated in one of two ways. Assertion of ISA ready condition indicates to the ISA controller that the PCI cycle finished correctly and, in the case of a read, that read data is available on the read data bus. If an ISA unclaimed state is asserted it means that the PCI cycle was either not claimed by any other PCI device (master aborted) or the the target terminated the transaction with a target abort, in either case the transaction should be completed back on the ISA bus. If the PCI cycle ends in retry, the PCI master interface takes care of rerunning the transaction until it terminates normally.

PCI Target Interface:

PCI bus mastering devices can communicate with West Bridge resources (IDE PIO registers or ISA based devices) via the PCI target interface. The target interface also contains the West Bridge PCI configuration register set, accesses to these registers are handled locally to the target interface.

PCI Address Decode:

The only positive decode done by the West Bridge is for the IDE controllerPIO 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.

PCI Bus Latency Controls:

The PCI spec rev 2.1 introduced some guidelines regarding bus latency and bandwidth sharing/fairness. In order to meet these guidelines the following algorithm is used: when the West Bridge decodes and claims a PCI cycle and begins the process of forwarding it to the either the IDE or the ISA controllers a counter is preloaded with the count of 8. This counter is decremented every PCI clock in which the IDE or ISA controller is not yet finished with the transaction. When the count reaches 0 a retry is signaled on the PCI bus, but the IDE or ISA controller continues to process the transaction. Eventually the IDE or ISA controller finishes the transaction and the original PCI master regains the PCI bus and reruns the same transaction. At this point the connection is reestablished and the cycle is terminated successfully. This mechanism satisfies the requirement that the first PCI data phase complete with 16 PCI clocks or disconnect.

PCI Error Handling:

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

- ISA initiated transaction ending in target abort
- IDE initiated transaction ending in target abort

PCI Arbitor:

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

The IDE and ISA requests are generated by the PCI master interface whenever a cycle is being requested by one of these. Handshake with the arbiter is also done by the master interface.

5.3.1 WEST BRIDGE PCI FUNCTION 0 CONFIGURATION REGISTERS

This section describes Function 0 (F#0) configuration registers, including the PCI to ISA bridge control.

5.3.1.1 W.B. Vendor Identification register PCI Config. F#0 Offset 00h-01h

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

Bits 15-8 These bits are hardwired to 10h.

Bits 7-0 These bits are hardwired to 4Ah.

Writes to this register have no effect.

5.3.1.2 W.B. Device Identification register PCI Config. F#0 Offset 02h-03h

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

Bits 15-8 These bits are hardwired to 02h

Bits 7-0 These bits are hardwired to 09h

Writes to this register have no effect.

5.3.1.3 W.B. PCI Command register PCI Config. F#0 Offset 04h-05h

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 West Bridge may assert SERR# upon detecting a target abort in response to a West Bridge initiated PCI transaction on behalf of an ISA master. If this bit is set to ‘0’, the West Bridge will not assert SERR#.

Bit 7-4 *Reserved.* These bits are hardwired to a ‘0’. Writes to it have no effect.

Bit 3 **Enable Special Cycles.** This bit is hardwired to a ‘1’. The West Bridge writes to it have no effect. The West 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.

5.3.1.4 W.B. PCI Status register
PCI Config. F#0 Offset 06h-07h

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 West Bridge on behalf of an ISA master cycle. Writing a ‘1’ to this bit will clear it. SERR# is asserted in response to a target abort during an ISA master cycle on PCI bus and if bit-8 of the F#0 PCI command register is set to a ‘1’ to enable SERR# signaling.

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

Bit 12 **Received Target Abort.** This bit is set to a ‘1’ when PCI transaction initiated by the West 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 West Bridge terminates a PCI transaction with a target abort. Writing a ‘1’ to this bit will clear it. The West 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 West 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.

5.3.1.5 W.B. PCI Revision ID register
PCI Config. F#0 Offset 08h

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

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

5.3.1.6 W.B. Device Class Code register
PCI Config. Offset 09h-0Bh

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

Bits 31-24 (0Bh) **Base Class Code.** These bits arehardwired to 06h (bridge device).

Bits 23-16 (0Ah) **Sub Class Code.** These bits arehardwired to 01h (ISA bridge).

Bits 15-8 (09h) *Reserved.* Hardwired to 00h.

5.3.1.7 W.B. Header Type
PCI Config. F#0 Offset 0Eh

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

5.3.2 WEST BRIDGE PCI FUNCTION 1 CONFIGURATION REGISTERS

The West Bridge is a PCI bus agent and responds to PCI configuration cycles targeted to Device 12 (AD12). The West Bridge implements two functions:

- Function 0: PCI to ISA bridge

- Function 1: Bus master IDE controller

Accordingly, the West Bridge responds to both Function 0 and Function 1 configuration cycles directed to Device 12.

This section describes the Function 1 (F#1) configuration registers.

5.3.2.1 W.B. Vendor Identification register PCI Config. F#1 Offset 00h-01h

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

Bits 15-8 These bits are hardwired to 10h.

Bits 7-0 These bits are hardwired to 4Ah.

Writes to this register have no effect.

5.3.2.2 W.B. Device Identification register PCI Config. F#1 Offset 02h-03h

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

Bits 15-8 These bits are hardwired to 02h

Bits 7-0 These bits are hardwired to 09h

Writes to this register have no effect.

5.3.2.3 W.B. PCI Command register PCI Config. F#1 Offset 04h-05h

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 West Bridge may assert SERR# upon detecting a master or target abort in response to a the West Bridge initiated PCI transaction on behalf of IDE master. If this bit is set to ‘0’, the West Bridge will not assert SERR#.

Bit 7-3 *Reserved.* These bits are hardwired to a ‘0’. Writes to it have no effect.

Bit 2 **Bus Master Enabled.** Setting this bit to ‘1’ enables the bus mastering and DMA function of the IDE. Setting it to ‘0’, disables bus mastering and IDE can be used only in PIO mode.

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 ports.

This register defaults to 0000h after reset.

5.3.2.4 W.B. PCI Status Register PCI Config. F#1 Offset 06h-07h

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 West Bridge on behalf of an IDE master cycle. Writing a ‘1’ to this bit will clear it. SERR# is asserted in response to master or target abort during an IDE master cycle on the PCI bus and if bit 8 of the F#1 PCI command register is set to a ‘1’.

Bit 13 **Signaled Master Abort.** This bit is set to a ‘1’ when the West Bridge terminates a PCI transaction initiated on behalf of the IDE master with a master abort. The West Bridge master aborts an IDE master cycle if no target responds to this cycle.

Bit 12 **Received Target Abort.** This bit is set to a ‘1’ when a PCI transaction initiated by the West Bridge on behalf of the IDE master is terminated with a target abort. Writing a ‘1’ to this bit will clear it.

Bit 11 *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.

5.3.2.5 W.B. Revision ID Register PCI Config. F#1 Offset 08h

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.

5.3.2.6 W.B. Programming Interface Register PCI Config. F#1 Offset 09h

Bit 7 This bit is hardwired to ‘1’ indicating that the device supports master IDE. 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 either 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 primary 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.

5.3.2.7 W.B. Sub-class code register PCI Config. F#1 Offset 0Ah

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

5.3.2.8 W.B. Base-class code register PCI Config. F#1 Offset 0Bh

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

5.3.2.9 W.B. Latency timer control register PCI Config. F#1 Offset 0Dh

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

This register defaults to 00h after reset.

5.3.2.10 W.B. Header Type PCI Config. F#1 Offset 0Eh

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

5.3.2.11 W.B. IDE Base Address 0 register PCI Config. F#1 Offset 10h-13h

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 0x00000001 at reset.

5.3.2.12 W.B. IDE Base Address 1 register PCI Config. F#1 Offset 14h-17h

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.

5.3.2.13 W.B. IDE Base Address 2 register PCI Config. F#1 Offset 18h-1Bh

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

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 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.

5.3.2.14 WB IDE Base Address 3 register PCI Config. F#1 Offset 1Ch-1Fh

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

5.3.2.15 W.B. IDE Base Address 4 register PCI Config. F#1 Offset 20h-23h

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

Bits 31-4 **Base Address.** This field specifies the 16-bytes IO address range where the Bus master control and status registers are located.

Bits 3-2 Hardwired to '0' to indicate that this base address occupies 16-bytes in IO space.

Bit 1 *Reserved.* Hardwired to '0'.

Bit 0 **Memory space indicator.** This bit is hardwired to '1' to indicate IO space.

This register defaults to 00000001h at reset.

5.3.2.16 W.B. Primary IDE Timing register PCI Config. F#1 Offset 40h-43h

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 **DMA Speed Mode Select.** These bits, along with bits 29-26, determine width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.

00 - Fast mode
01 - Medium fast mode
10 - Medium slow (Default)
11 - Slow mode

The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.

Bits 29-28 **DMA Recovery Time.** These bits, along with bits 31-30, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.

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

Bits 27-26 **DMA Active Time.** These bits, along with bits 31-30, determine the duration of the active time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.

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

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 the table below to determine the number of clocks for recovery times of the read/write signals:

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 a active time of 10 clocks.

Refer to the table below to determine the number of clocks for active times of the read/write signals:

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. After initial startup latency, the address setup time is as follows :

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 for ATAPI commands (when 0xA0 is written into the command register - Register offset 0x07 for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices.

Prefetch for the IDE controller is as follows :

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

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

Bits 15-14 **DMA Speed Mode Select.** These bits, along with bits 13-10, determine width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.

00 - Fast mode
01 - Medium fast mode
10 - Medium slow
11 - Slow mode (Default)

The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.

Bits 13-12 **DMA Recovery Time.** These bits, along with bits 15-14, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.

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

Bits 11-10 **DMA Active Time.** These bits, along with bits 15-14, determine the duration of the active time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.

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

Bits 9-7 **PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers.

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

Refer to the table below to determine the number of clocks for recovery times of the read/write signals:

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 the table below to determine the number of clocks for active times of the read/write signals:

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 as follows :

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 ever 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 to0, prefetch is completely disabled.

Bit 0 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when 0xA0 is written into the command register - Register offset 0x07 for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices.

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

This register defaults to 0x97609760 at reset.

5.3.2.17 W.B. Secondary IDE Timing register PCI Config. F#1 Offset 44h-47h

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 **DMA Speed Mode Select.** These bits, along with bits 29-26, determine width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.

00 - Fast mode
01 - Medium fast mode
10 - Medium slow
11 - Slow mode (Default)

The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.

Bits 29-28 **DMA Recovery Time.** These bits, along with bits 31-30, determine the duration of the recovery (inactive) time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.

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

Bits 27-26 **DMA Active Time.** These bits, along with bits 31-30, determine the duration of the active time of the read/write signals during DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.

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

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 the table below to determine the number of clocks for recovery times of the read/write signals:

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 a active time of 10 clocks.

Refer to the table below to determine the number of clocks for active times of the read/write signals:

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. After initial startup latency, the address setup time is as follows :

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 IOCHRDYsampling 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 thatthe 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 to0, prefetch is completely disabled.

Bit 16 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when 0xA0 is written into the command register - Register offset 0x07 for the device). When set to 0, prefetching during ATAPI commands is disabled. This isto accommodate non-512 boundary data transfers that are supported by ATAPI devices.

Prefetch for the IDE controller is as follows :

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 **DMA Speed Mode Select.** These bits, along with bits 13-10, determine width of the read and write signals during DMA transfers. Refer to the table below to determine the number of clocks for active and recovery times for the read/write signals.

00 - Fast mode
01 - Medium fast mode
10 - Medium slow
11 - Slow mode (Default)

The slower modes are normally used for single word DMA modes and the faster modes are used for double word DMA modes.

Bits 13-12 **DMA Recovery Time.** These bits, along with bits 15-14, determine theduration of the recovery(inactive) time of the read/write signals during DMA transfers.

Refer to the table below to determine the number of clocks for recovery times of the read/write signals. Default is 01h.

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

Bits 11-10 **DMA Active Time.** These bits, along with bits 15-14, determine theduration of the active time of the read/write signalsduring DMA transfers. Refer to the table below to determine the number of clocks for active times of the read/write signals. Default is 01h.

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

Bits 9-7 **PIO Recovery Time.** These bits determine the duration of the recovery (inactive) time of the read/write signals during PIO data transfers.

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

Refer to the table below to determine the number of clocks for recovery times of the read/write signals:

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 the table below to determine the number of clocks for active times of the read/write signals:

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 as follows :

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 ever 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 to0, prefetch is completely disabled.

Bit 0 **Enable Prefetch for ATAPI commands.** When set to 1, this bit enables prefetching for ATAPI commands (when 0xA0 is written into the command register - Register offset 0x07 for the device). When set to 0, prefetching during ATAPI commands is disabled. This is to accommodate non-512 boundary data transfers that are supported by ATAPI devices.

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

This register defaults to 0x97609760 at reset.

5.3.2.18 W.B. Miscellaneous Register PCI Config. F#1 Offset 48h

This register contains some miscellaneous informations.

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

Bits 6-4 *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 0x00 at reset.

6 ISA INTERFACE

7.1 Introduction

The East 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 write that target East 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 East Bridge also routes PCI reads and writes to main memory and its internal registers. The West 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 West Bridge will claim all PCI cycles which were initiated outside the West 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 West Bridge's PCI controller. All other PCI operations, including reads and writes to the West 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, stop grant and halt are also sent to the ISA bus controller. These cycles do not create ISA bus cycles, but they use the same state machines for timing and arbitration as reads and writes. Interrupt acknowledge is covered in this section, shutdown, stop grant and halt are covered in section 5.16 (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.

Every cycle initiated on the ISA bus is tried on the PCI bus. If the the cycle is claimed by some PCI target, then data is read from or written to that target. If the PCI cycle is not claimed, and the cycle targets a West Bridge internal register, then that register is read from or written to. Otherwise, the target is expected to be on the ISA bus.

7.2 PCI / ISA Cycles

7.2.1 PCI to ISA read and write

The PCI transfers data four bytes at a time, with byte enables for each byte. The West 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.

7.2.2 PCI to internal register read and write

All West 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 West 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 West Bridge drives the ISA data bus with the contents of the register, and a 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 East Bridge and one on the West Bridge.

Writes to IO address 22h go to both copies of the register. Reads from IO address 22h normally come from the East 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 West 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 East Bridge alone, some in the West Bridge alone, and some are duplicated and implemented in both. Whether an index registe is implemented in the East Bridge, West Bridge or both is indicated in the description of that register in this document.

For index registers that are implemented in the East 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 West 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 East Bridge and the West 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 East 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 21. In this case, the West Bridge copy of the register is read, using a PCI read cycle.

7.2.3 Interrupt Acknowledge Cycle

When an interrupt is requested, the interrupt controller in the West 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 East Bridge handles both of the cycles identically, converting them to PCI interrupt acknowledge cycles.

Outside of the interrupt controller, the West 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 50.

7.2.4 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 4.12.3).

7.2.5 ISA to PCI read and write

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

7.2.6 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 a ISA initiated read of a host memory address has been detected by the West Bridge. The West 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 West Bridge onto the ISA bus to finish the ISA read cycle.

The buffer will be refilled if the data requested by the current read is not in the buffer. Also, to avoid stale data, the buffer will be refilled for the first host memory read after an ISA bus master gets ownership of the bus, for the first host memory read after any ISA bus cycle which is not a host memory read, and for any ISA read of a byte in the buffer which has already been read since the buffer was last filled.

If a host memory read can be fulfilled without refilling the buffer, no PCI cycle is generated.

7.2.7 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 West Bridge to generate writes with discontiguous byte enables if posted writes are enabled.

7.2.8 ISA to register read and write

ISA initiated cycles which target West 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 West Bridge to pull IOCHRDY low for at least the number of cycles programmed into Index Register 01. Reads and writes to the West 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.

7.3 XBUS read and write

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

7.3.1 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#, RW#, DS) are controlled directly by the STPC. The MOT pin of the RTC must be tied low. The registers in the RTC are accessed indirectly, by first writing the register number to IO port 70h, and then reading or writing the register at IO port 71h.

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

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

7.3.2 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 West Bridge to be reads and writes to the keyboard controller. Writes to the keyboard controller may be intercepted by West Bridge for keyboard controller emulation (see Section 7.3.2). In this case, neither IOW# or KBCS# will be asserted. For writes to the keyboard controller that are not intercepted, both IOW# and KBCS# will be asserted (low) during the write. Similarly, for any reads from the keyboard controller, both IOR# and KBCS# will be asserted (low) during the read.

7.3.3 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.

7.4 Fast CPU Reset and Fast Gate A20:

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

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

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

7.5 ISA STANDARD REGISTERS

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

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

7.5.1 DMA 1 registers

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 CPC pin DRQ3B.

There are 16 DMA 1 registers. They are as shown in Table 7

Table 7. DMA 1 Registers

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

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

Detailed description of the 8237 DMA controller operation can be obtained from Intel Peripheral Components Data book.

7.5.2 Interrupt Controller 1 registers

Interrupt controller 1 is the master interrupt controller. Interrupt controller 1 input IR0 is connected pin IRQ0B, IR1 to IRQ1B, IR2 to interrupt out from interrupt controller 2, IR3 to IRQ3B, IR4 to IRQ4B, IR5 to IRQ5B, IR6 to IRQ6B, and IR7 to IRQ7B.

There are two Interrupt controller 1 registers. They are as shown in Table 8

Table 8. Interrupt Controller 1 registers

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

Note that not all bits of the address are used.

Detailed description of the 8259 Interrupt Controller operation can be obtained from Intel Peripheral Components Data book.

7.5.3 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 4.12.4).

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

Table 9. Interval Timer registers

IO address bits 15-0	Reset Value	Register Name
0000 0000 010x xx00	xxxx xxxx	Counter 0 count
0000 0000 010x xx01	xxxx xxxx	Counter 1 count
0000 0000 010x xx10	xxxx xxxx	Counter 2 count
0000 0000 010x xx11	1111 1111	Command Mode register

Note that not all bits of the address are decoded.

Detailed description of the 8254 Timer/Counter operation can be obtained from Intel Peripheral Components Data book.

7.5.4 Port B

This is the ISA compatible 8-bit Port B register located at 0000 0000 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 clear the F/F and then set to a ‘0’ to enable further IOCHK# assertions. This bit is read/write and cleared to a ‘0’ by ISA reset.

Bit 2 *Reserved.* Although this bit is read/write, it iscleared 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.

7.5.5 Port 60 and 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 STPC uses these ports to generate HA20M# and Fast CPU reset.

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

Fast host CPU only reset is generated by two methods:

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

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

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

7.5.6 Port 70

This 8-bit write-only register contains the NMI enable bit and is located at 0000 0000 0111 0xx1 IO address. Writing to this address also sets the address register in the Real Time Clock (RTC, not part of the STPC, it is normally connected via the ISA interface).

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.

7.5.7 Interrupt Controller 2 registers

Interrupt controller 2 is the slave interrupt controller. Interrupt controller 2 input IR2 is connected to IRQ9B, IR2 to IRQ10B, IR3 to IRQ11B, IR4 to IRQ12B, IR6 to IRQ14, IR7 to IRQ15. IR0 driven by IRQ8B inverted. IR5 is driven by an internally generated floating point error interrupt request.

Interrupt controller 2 occupies two register locations. They are as shown inTable 10:

Table 10. Interrupt Controller 2 registers

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

Note that not all address bits are decoded.

For a detailed description of the operation of the 8259 interrupt controller, refer to Intel Peripherals Components Data book.

7.5.8 DMA Controller 2 registers

There are 16 DMA 2 registers. They are as shown in Table 11

.

Table 11. DMA Controller 2 registers

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

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

Detailed description of the 8237 DMA controller operation can be obtained from Intel Peripheral Components Data book.

7.5.9 DMA Page registers

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

There are 16 DMA page registers. They are as shown in Table 12.

Table 12. DMA Page registers

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

7.6 ISA Configuration registers

7.6.1 Miscellaneous Control Register 0 - C.I.50

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 is disabled. The source of warm reset indication is from 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.

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.

7.6.2 Miscellaneous Control register 1 - C.I.51

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 generation 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 extended bus. If set to '0', write to extended bus ROM BIOS is allowed. Note: This bit can not disable the write to shadowed BIOS in DRAM since after shadow is enabled, all 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 FLASH 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 FLASH memory with F0000h-FFFFFh segment.

0 = sharing disabled
1 = sharing enabled

This register defaults to 00h at reset.

7.6.3 PIRQA Routing control register 0 - C. I.52

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:

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.

7.6.4 PIRQB Routing control register 0 - C.I. 53

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:

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.

7.6.5 PIRQC Routing control register 0 - C. I. 54

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:

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.

7.6.6 PIRQD Routing control register 0 - C.I. 55

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 enables the routing of PCI interrupt, otherwise the PCI interrupt D# is unconnected.

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

Bit 3-0 **Routing Control D#.** These bits route the PCI interrupt D# as follows:

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.

7.6.7 Interrupt Level Control Register 0 - C. I. 56

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.

7.6.8 Interrupt Level Control Register 1C. I. 57

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 to 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 outone value from IPC. When low will force the outone to low, else the outone is driven by IPC.

7.6.9 IPC Configuration register - C.I. 01

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

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

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 specify the number of wait states in 8 bit DMA cycles

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

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.

7.6.10 VMI IRQ Routing control register - C.I. 58

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:

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.

7.6.11 ISA synchronizer bypass - C.I. 59

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.

7 IDE CONTROLLER

7.1 Introduction

The IDE controller provides 2 IDE channels, primary and secondary, for interfacing 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.

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

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

7.2 PRD Table Entry

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

Bits 62-48 *Reserved.*

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

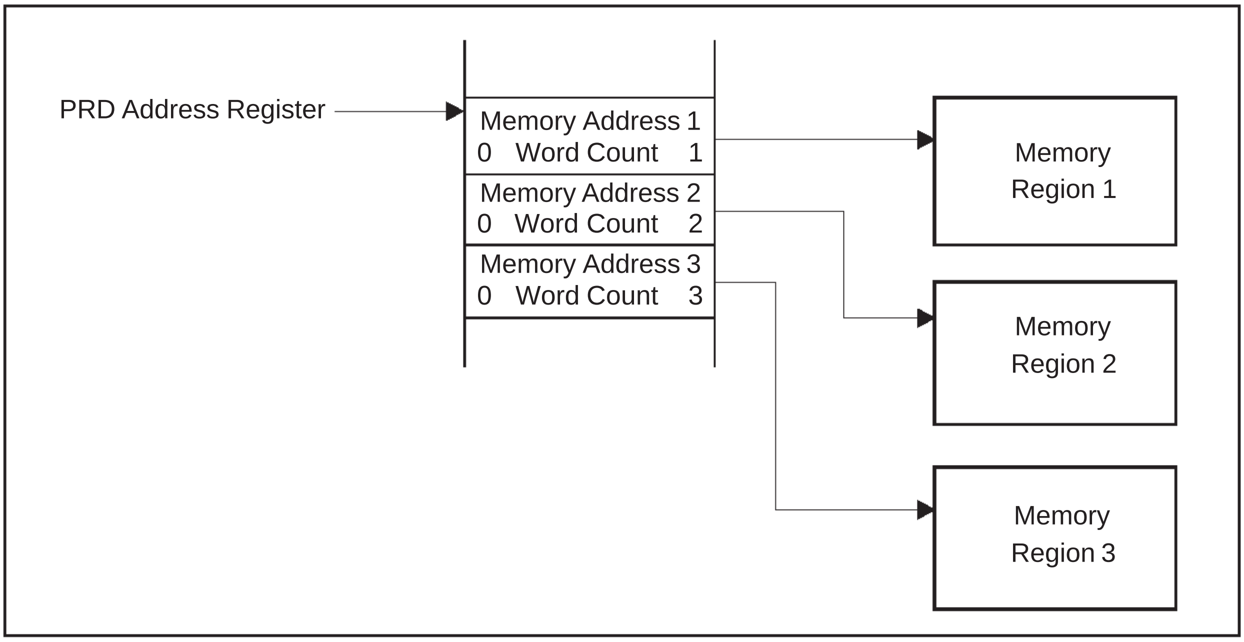
Bit 32 *Ignored.* The byte count must be even.

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

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

The table must be aligned on a 4 byte boundary and should not cross a 64k boundary. A memory region also should not cross a 64k boundary. An example of a PRD table is shown inFigure 9..

Figure 9. PRD Table Entry Example



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 East Bridge arbitrates between the IDE controller and the ISA bus bridge to select which has control of the shared pins.

7.3 IDE Bus Master Registers

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

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

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

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

7.3.1 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 number of Physical Region Descriptors (PRD) which describe areas of memory that are involved in the data transfer. The descriptor table must be aligned on a 4 byte boundary and the table cannot cross a 64K boundary in memory.

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

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

7.4 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 follows:

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

7.4.1 Bus Master IDE Command Register

7.4.1.1 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.

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

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

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

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

Bit 2 **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 encounters an error in transferring data to/from memory. The exact error condition is bus specific and can be determined in a bus specific manner. This bit is cleared when a ‘1’ is written to it by software.

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

Reset value is 00h.

7.4.1.3 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 7.2

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

7.5 Operation

7.5.1 Standard Programming Sequence

To initiate a bus master transfer between memory and an IDE DMA slave 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.

7.6 Data Synchronization

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

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

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

* Flush all buffered data

* Set the Interrupt bit in the controller Status register

* Guarantee that a read to the controller Status register does not complete until all buffered data has been written to memory.

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

7.6.1 Status Bit Interpretation

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

Interrupt	Active	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.

7.7 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.

7.8 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 (top two bytes have the value 0x0101) 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 Error Detected	Any time bit 8 of the Config Space Status register is set.

8 VGA CONTROLLER

8.1 Introduction

The STPC integrates a full VGA Controller with Extended Functions together with an output Color Digital to Analog (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 streamfrom the video buffer and the appropriate horizontal and vertical synchronisation pulses. The Frame video buffer exists in the main DRAM memory in the first 4M of the memory space. This Frame buffer area is selected upon configuration of the VGA video output and, once selected thefunction of the Frame buffer area can not be changed back to normal DRAM memory program/data functions until the next reset cycle.

The on-chip triple RAMDAC runs at up to 135MHz, using and 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.

The Graphics Engine provides a high performance 64-bit windows accelerator. It is completely backward compatible with the VGA and SVGA standards. Hardware acceleration is provided for BITBLTs, transparent BLTs and fills and text (generalised bit map expansion).

The Graphics Display Controller is fully integrated with the input and display of video signals (see Next Chapter).

8.2 VGA Controller

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

Resolutions of up to 1280x1024 and color depths of 8, 16, 24 and 32 bits per pixel are supported. The integrated RAMDAC supports digital to analog conversion rates of up to 135 MHz. This along with the peak video bandwidth of 320 MBytes/sec (using EDO DRAMs) enable the VGA Controller to support 1280x1024x16, 1024x768x24 and 800x600x32 resolutions at 75 Hz refresh rate.

The RAMDAC supports internal or external voltage reference source as well as DDC monitor sense functionality.

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 locating the frame buffer at any double word boundary within this 4Mb of memory. This frame buffer sits within the 16Mbyte 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.

8.3 VGA registers

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

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

8.3.1 General Registers

8.3.1.1 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 not altered by drawing operations.

The contents of this register are 28h after reset.

8.3.1.2 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.

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

This register is defined to be 00h after reset.

8.3.1.3 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 affected by this bit. The VGA continues to display video data while disabled.

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

This register is defined to be 00h after reset.

8.3.1.4 Video Subsystem Enable 2 Reg

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 accessible to allow enabling of the VGA. Port 94h is also not affected by this bit. The VGA continues to display video data while disabled.

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

This register is defined to be 00h after reset.

8.3.1.5 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:

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.** These register bits drive two clock select output pins of the DPC chip. The pins are named CLOCK0 and CLOCK1.

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:

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

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

This register is defined to be 00h after reset.

8.3.1.6 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.

8.3.1.7 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 follows:

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.

8.3.1.8 Reserved Register
3CAh/3XAh (R/W)

Bits 7-0 *Reserved.* These bits read as zero.

8.3.2 Sequencer Registers

8.3.2.1 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.2.2 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 altered by drawing operations.

This register is defined to be 00h after reset.

8.3.2.3 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 follows:

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 input dot clock is divided by 2 for 320 and 360 horizontal pixel modes 0, 1, 4, 5, D and 13.

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.

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

This register is defined to be 00h after reset.

8.3.2.4 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:

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

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

This register is defined to be 00h after reset.

8.3.2.5 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**

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 following tables:

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

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

Note that when the frame buffer width is only 32 bits and high speed text mode is entered, bits 3 and 1 must be programmed identically otherwise the screen will be garbage.

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

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

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

8.3.2.6 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 ignored 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 four display planes for host transfers. HA0 = ‘0’ selects planes 0 and 2, and HA1 = ‘1’ selects planes 1 and 3. Selected planes are ANDed with the Plane Mask register (SR2) to generate the plane write enables during write transfers. Read transfers use Map Select bit 1 from GR4 along with HA0 to select one of the 4 bytes to be returned to the host. Read Map select bit 0 is not used when odd/even addressing mode is enabled.

Bit 1 **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’.

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

This register is defined to be 04h after reset.

8.3.2.7 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’.

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

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

8.3.3 Graphics Controller Registers

8.3.3.1 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.3.2 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.3.3 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.3.4 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.3.5 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:

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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.3.6 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 altered by drawing operations.

The contents of this register are not defined after reset.

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

Bit 7 *Reserved,* read as ‘0’.

Bits 6-5 **Shift mode**

Bit6	Bit5	Shift Register Behavior
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 Register (GR4). This is also called Read Mode 0.
When this bit is set to ‘1’, a host read transfer returns the result of the logical comparison between the data in the four planes selected by the Color Don’t Care Register (GR7) and the contents of the Color Compare Register (GR2). This is also called Read Mode 1.

Bit 2 *Reserved,* reads as ‘0’.

Bits 1-0 **WriteMode.** These bits select the write mode as follows
:

Bit1	Bit0	Write Behavior
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.

The contents of this register are not defined after reset.

8.3.3.8 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:

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 duplicated in AR10[0].

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

The contents of this register are not defined after reset.

8.3.3.9 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.3.10 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 altered by drawing operations.

The contents of this register are not defined after reset.

8.3.4 Attribute Controller Registers

8.3.4.1 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.

8.3.4.2 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.**

8.3.4.3 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.

8.3.4.4 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.

8.3.4.5 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:

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 input into the Palette. If any of these bits are zero, the corresponding plane from the frame buffer will be masked out of the Palette look up .

The bits in this register reset to zero.

8.3.4.6 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.

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.

8.3.4.7 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 color 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.

8.3.5 CRT Controller Registers

The STPC implements an extension of the VGA CRTC controller. The CRTC controller supports up to 1280x1024 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.

8.3.5.1 Index register CRX 3X4h (RW)

The CRTC Index register points to an internal register of the CRT controller. The five least significant bits determine which register will be pointed to in the next register read/write operation to IO port 3B5/3D5.

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

Bits 5-0 **CRTC Index.** Points to the CRTC register that will be accessed by an IO cycle at 03B5h/ 03D5h.

This register resets to zero.

8.3.5.2 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.

8.3.5.3 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.

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

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

This register is in unknown state after reset.

8.3.5.5 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 operation. It resets to one.

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

Bits 4-0 **Horizontal Blanking End Value Bits 4-0.** These bits specify the least significant 5-bits of the 6-bit wide Horizontal Blanking End value. The sixth bit is located in CRTC Horizontal Retrace End register. This field is in unknown state after reset.

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.

8.3.5.6 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.

8.3.5.7 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.

8.3.5.8 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.

8.3.5.9 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.

8.3.5.10 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:

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 the character cell height value programmed in CR9. If this field is programmed to a value larger than the character cell height, the row scan counter will count up to 1Fh before rolling over. A ‘0’ in this field means no scrolling.

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

8.3.5.11 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.

8.3.5.12 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 register is in unknown state after reset.

8.3.5.13 Cursor End register CRB 3X5h Index B (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 character 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.

This register is in unknown state after reset.

8.3.5.14 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 CRD contains the lower 8-bits and the CRTC extended register CR19 contains the upper 4 bits.

If split screen mode is in effect, this address is the start address of the first of the two (the top one) screens (Screen A). The start address of Screen B (the bottom one) is always 0. The starting scan line for the Screen B is determined by the line compare register (CR18).

8.3.5.15 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.

8.3.5.16 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.

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

This register acts as 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.

8.3.5.18 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.

8.3.5.19 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 writeable.

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 determines 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 reset.

8.3.5.20 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 overflow register and the bit-10 in the Repaint Control Register 4.

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

8.3.5.21 Offset register CR13 3X5h Index 13 (RW)

This register defines bits 7-0 of the 10-bit wide logical width of the line displayed on the screen. Extended register CR19 contains the upper two bits. The first scan line displayed on the screen, starts at the address specified in registers CRC, CRD and extended register CR19. The starting address of the next scan line is computed as the byte starting address of the current row + 2*offset.

This register is undefined after reset.

8.3.5.22 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.

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.

This register is undefined after reset.

8.3.5.23 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.

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

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

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.

This register is undefined after reset.

8.3.5.25 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 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.

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 memory address bit. If set to a ‘1’, it propagates bit iA15 instead. iA13 should be used if total display buffer memory is 64K and iA15 should be used if total memory is 256K. It is expected that the VGA Controller will always be used with 256K or larger memory. Therefore this bit should be programmed to a ‘1’.

Bit 4 *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.

8.3.5.26 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 horizontally with Screen A or not.

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

This register is undefined after reset.

8.3.5.27 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 register (GR4) specify which latch is read.

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

The contents of this register is not defined after reset.

8.3.5.28 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.

8.3.5.29 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 Controller Index register.

8.3.6 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.

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

Bit 7 **"Turbo Mode" Enable.** Programming this bit to '1' enables enhanced performance mode. The effect is to increase system performance by reducing the bandwidth required by the display graphics datapath thus giving more bandwidth back to the CPU. This bit resets to zero (performance enhancement disabled).
Note that after "turbo mode" is turned on, any software that alters CRTC registers which "move the screen" (such as start address, display offset etc.) should also program bit 6 to a one. Alternatively, the software can do a dummy host write to the VGA frame buffer.

Bit 6 **Tag Invalidate.** Programming this bit to '1' invalidates all the "turbo mode" tags forcing them to be recomputed. This bit resets itself to zero when the operation is complete.

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.

8.3.6.2 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).

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).

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 zeroed).
Set this bit to zero for SVGA modes.

This bit has no effect on CPU reads or writes to the frame buffer - only CRTC accesses.

This register resets to 3Fh.

8.3.6.3 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. Also, do not set this register to a value greater than that programmed into the high water mark (register 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 underflows. 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 water mark value) reset this bit to zero.

This register defaults to 20h at reset.

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

Bit 7 **Enhanced Text Font Load.** This bit should be set to one prior toloading 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 zero.

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

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 software should subtract half page for passing on the address.

When this bit is ‘1’, the memory address bits MA17 and MA18 are changed as follows for Normal, Odd/Even and Chain-4 cases:

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:

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	

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

This register is defined to be 00h after reset.

8.3.6.5 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.

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

The contents of this register is not defined after reset.

8.3.6.6 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.

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

The contents of this register is not defined after reset.

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

Bit 7 **Exen.** Writing a ‘1’ in this bit enables the GE extended functionality. Writing a ‘0’ disables it. After reset, this bit is set to ‘0’.

Bits 6-0 *Reserved,* read as ‘0’s

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

This register is defined to be 00h after reset.

8.3.6.8 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.

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

This register is defined to be 00h after reset.

8.3.6.9 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 MByte 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 contents of this register are not altered by drawing operations.

This register is defined to be FFh after reset.

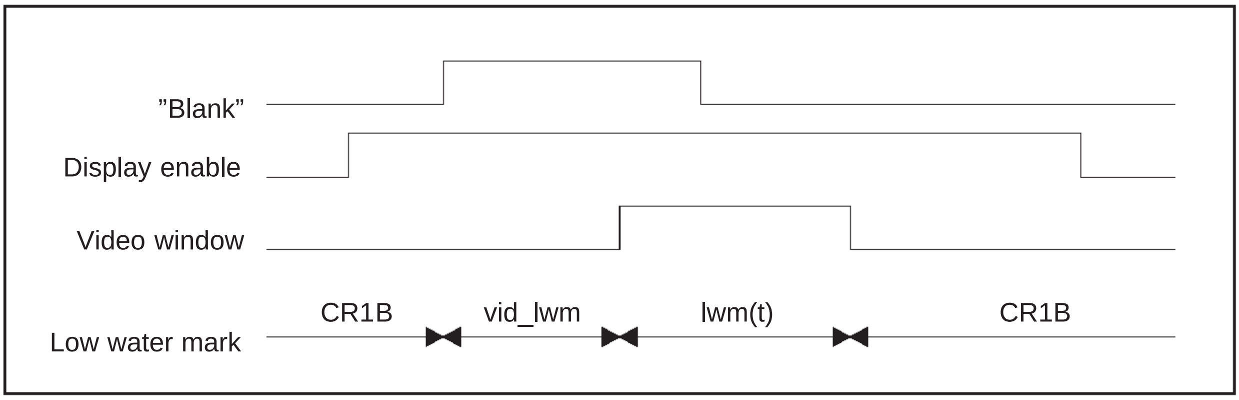
8.3.6.10 Repaint Control Register 4 CR25 3X5h Index 25 (RW)

Bits 7-6 *Reserved,* read as ‘0’s

Bit 5 **Blank With Display Enable.** When this bit is set to one use display enable to generate the blank signal. This eliminates the border but allows the blank timing generator to be used for sophisticated low water mark control when video in a window is displayed.

Consider the following diagram inFigure 9-1:

Figure 9-1. Low Watermark timing



In this scenario, the CRTC low water mark is determined as follows:1

Disp Enable	Video Window	Blank	Low Water Mark
0	X	X	CR1B
1	0	0	CR1B
1	0	1	crtc_lwm
1	1	X	lwm(t)

Where:

crtc_lwm is the contents of the CRTC_Burst_length register, bits [23-16] and lwm(t) is the variable low water mark described in the same register.

If crtc_lwm is set to 255 bytes, then the above operation can be used to guarantee CRTC ownership of the bus at the beginning of the video window: Setting crtc_lwm to 255 will cause the CRTC to request ownership of the DRAM at the rising edge of “blank”. If “blank” is high prior to the active video window for approximately 23 GCLKs or more, the CPU or any other potential owner will be guaranteed by design to relinquish the bus in time for the start of the video window.

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 register.**

This register is set to 00h after reset.

8.3.6.11 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 shouldonly ever be set by the BIOS during mode switches. Do not set this register to a value greater than the default value of D0h - nothing will work.

Bits 2-0 *Reserved*, read as ‘0’.

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

8.3.6.12 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 power down.

Bit 4 **Enable On Chip Voltage Reference.** When this bit is set to one, the on-chip analog voltage reference is used for the analog to digital converters. In this case the VREF pin should be a no-connect. When this bit is zero, the on-chip reference is disabled and an external reference must be supplied - connected to the VREF pin. This bit defaults to zero after reset

Bit 3 **LUT Bypass.** Setting this bit to one bypasses the RAMDAC look up table (LUT) and allows pixels to drive the DACs directly. When this bit is set to zero the look up table is used to compute final pixel colors. This provides palette functionality for 8 bit and other low color modes and gamma correction (non-linearity compensation) for the 24 and 32 bit true color modes.

Bits 2-0 **Pixel Format.** These 3-bits specify the pixel color depth and are encoded as follows:

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

For 15-bit and 16-bit pixels, the 5 or 6 bits per color are shifted left by 3 or 2 bits and then presented to the 8-bit DACs or LUT address (depending on bit 3 above). The least significant bits are zeroed.

The following resolutions are supported at 75 Hz refresh rate for each of the above color depths when a 64 bit bank of DRAM is used for the frame buffer:

Pixel Format	Maximum Resolution
VGA (other than mode 13)	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

These are the resolutions supported at 75 Hz when only a 32 bit wide frame buffer is available:

Pixel Format	Maximum Resolution
VGA (other than mode 13)	1280 x 1024
VGA (mode 13)	320 x 200
8 bit (non-VGA)	1024 x 768
15 bit	800 x 600
16 bit	800 x 600
24 bit	640 x 480
32 bit	-

Interlaced monitors and timings are supported.

This register resets to 08h after reset.

8.3.6.13 Cursor Height Register
CR29 3X5h Index 29 (RW)

Bit 7 **Cursor XOR Pre/Post Look Up Table.** When this bit is set to one, the graphics cursor XOR operation is performed before the look up table. The default behavior, when this bit is set to zero, is for the XOR operation to happen after the look up table. This is correct for 15, 16, 24 bit per pixel modes but not 8bpp.

Bits 6-0 **Cursor height.** This field represents the vertical extent of the graphics cursor in scan lines. Setting this to zero effectively turns the graphics cursor off. Values greater than 40h (decimal 64) are meaningless and produce unpredictable results.

Note: there is no cursor width register - the width is always 64 pixels. If a narrower cursor is required, pad the bitmap on the right with transparent cursor color (pad the AND plane with ‘1’s on the right and the XOR plane with ‘0’s).

This register is set to 00h after reset.

8.3.6.14 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.

8.3.6.15 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.

8.3.6.16 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.

8.3.6.17 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.

8.3.6.18 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.

8.3.6.19 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.

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

Bits 7-1 **Cursor AND Address Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the cursor's AND mask. The cursor's XOR mask begins at this address + 512. This memory must be aligned on a 1 KByte boundary. For a discussion of DRAM linear addresses, see section tbc.

Note that the cursor bitmap is ordered such that the top left hand corner of the cursor is represented by bit 7 of the byte addressed by this field (AND) and bit 7 of the byte at 512 plus this address (XOR plane). The next pixel right is represented by bit 6 of these bytes and so on until the bottom right hand pixel is represented by bit 0 of the byte locat ed at this address plus 511 (AND) and bit 0 of the byte at 1023 plus this address.

Bit 0 *Reserved.* This bit should be written as zero.

This register is undefined after reset.

8.3.6.21 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.

8.3.6.22 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.

8.3.6.23 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.

This register is typically programmed to be around 23 GCLKs before the start of the CRTC window.

8.3.6.24 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.

8.3.6.25 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.

8.3.6.26 Housekeeping Address 0 CR36 3X5h Index 36 (RW)

Bits 7-1 **Housekeeping Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the “turbo mode” housekeeping region. See description of CR19 bit 7.

8.3.6.27 4.7.6.25 Housekeeping Address 1 CR37 3X5h Index 37 (RW)

Bits 7-0 **Housekeeping Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the “turbo mode” housekeeping region.

8.3.6.28 Housekeeping Address 2 CR38 3X5h Index 38 (RW)

Bits 7-3 *Reserved.* These bits should be written as zero.

Bits 2-0 **Housekeeping Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the “turbo mode” housekeeping region.

This register is undefined after reset.

8.3.6.29 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).

8.3.6.30 Implementation Number Register CR3A 3X5h Index 3A (R)

Bits 7-0 **Implementation Number.** Indicates the hardware implementation number for the graphics drawing and display subsystem. The table below describes the interpretation of each value.

VALUE	IMPLEMENTATION
01h	STPC Implementation

8.3.6.31 Graphics Version Register CR3B 3X5h Index 3B (R)

Bits 7-0 **Graphics Version Number.** Indicates the hardware version number for the graphics drawing and display subsystem. The table below describes the interpretation of each value.

VALUE	IMPLEMENTATION
01h	STPC Implementation

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

See section 4.4.1 for the description of this register

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

See section 4.4.2 for the description of this register

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

See section 4.4.3 for the description of this register

8.3.6.35 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 only when CR19[6] is set to one. If CR19[6] is zero, writes have no effect.
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.

8.3.6.36 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 **MPEG Enable** (active low). This bit multiplexes the TV output port between the graphics pipeline (bit 5 = ‘1’) and the MPEG 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:

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 **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:

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 the following table:

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

8.3.6.37 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.

8.3.6.38 TV Horizontal Active Video Start B CR42 3X5h Index 42 (RW)

Bit 7 *Reserved.* This bit must be written as a ‘0’.

Bits 6-3 *Reserved.* These read/write bits should be written as ‘0’s to ensure future compatibility.

Bits 2-0 These represent bits 10:8 of the eleven bit wide horizontal active video start field. See section 4.7.6.34.

8.3.6.39 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.

8.3.6.40 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 separately 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.

8.4 Additional Modes

8.4.1 Fast 132 Character Wide Text Mode.

To meet the high bandwidth requirements of 132 column text mode, VGA Controller supports a special high speed text mode. For column widths of 96 characters and greater, bit 7 of extended register CR1C - the Repaint Control Register #3 must be set to one prior to loading the font tables into frame buffer plane two. Fonts may then be loaded in the standard VGA manner one byte at a time at the end of which bit 7 of CR1C should be reset to zero.

Setting bit 7 of CR1C to one performs an address warping such that standard VGA font load cycles actually store fonts into plane two the following way:

Byte 0: Character Set 0, Font (ASCII) 0, Line 0
Byte 1: Character Set 0, Font (ASCII) 1, Line 0
...
Byte 255: Character Set 0, Font 255, Line 0
Byte 256: Character Set 1, Font (ASCII) 0, Line 0
...
Byte 511: Character Set 1, Font 255, Line 0
Byte 512: Character Set 2, Font (ASCII) 0, Line 0
...
Byte 2047: Character Set 7, Font 255, Line 0
Byte 2048: Character Set 0, Font (ASCII) 0, Line 1

Applications which load their own fonts independent of the motherboard BIOS will not be supported in 132 column modes because of the above requirements.

Note that the above organization of font data will ensure that 132 column mode bandwidth requirements are low enough to be satisfied by 64 bit wide DRAM frame buffers only. If the frame buffer is 32 bits wide, then the primary and secondary character map selects (SR3) should only ever be programmed such that both of the primary and secondary fonts are in the range 0-3 or both are in the range 4-7. Failure to observe this requirement will result in a garbaged screen.

8.4.2 Turbo Graphics Mode.

Proprietary techniques are used to reduce the bandwidth required for the screen repaint in extended graphics modes. This mode is totally user transparent and is entered simply by programming bit 7 of extended register CR19 to a one. The only requirement made by this mode is that 32K of frame buffer space must be made available for exclusive use by VGA Controller for house-keeping functions.

8.5 CRTC-VIDEO Arbitration

The VGA Controller is capable of supporting 352x16bpp unscaled video. To satisfy both CRTC and video refresh requirements, the arbitration between the two for memory accesses have been modified to include a variable low water mark. If video is to be displayed, then the following conditions need to be satisfied first.

1. CR25[5] should be set to 1.7

2. Horizontal Blank start register should be programmed to be at least 23 gclks earlier than the rising edge of h_win (or the X coordinate of video window).

3. Horizontal Blank End register value should be later than the rising edge of h_win and earlier than the falling edge of h_win (The value should be within the video window). This can require up to 7 bits. CR3[4-0] specify the least significant 5-bits of the 7-bit wide Horizantal Blank End value.

The sixth bit is located in CRTC Horizantal Retrace End register CR5[7], and the 7th bit is located in CRTC Extended register CR25[4].

Programming CR25[5] will disable the display of overscan color by ignoring horizontal and vertical blank signal, and instead use the invert of horizontal and vertical display enable as the blank signals. CR2 and CR3 (blank start and blank end registers) are then used to generate a *new* blank signal used in the arbitration scheme.

8.6 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.

8.7 RAMDAC registers

8.7.1 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.

8.7.2 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.

8.7.3 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 address 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 address of the last IO write to ports 3C7h or 3C8h. ‘00’ indicate that the last write was to port 3C8h, ‘11’ indicate 3C7h.

8.7.4 Palette Write Index register 3C8h (RW)

This register is similar to the Read index register and contains the index value for the write access to the 256 entries of the color palette. Each entry is 24-bit wide and are written as a sequence of 3-bytes. After writing the index of the entry to be modified, the data values may be written to the DAC Data port (3C9h) in the sequence: 1) Red, 2) Green, and 3) Blue. This 3-byte write sequence is aborted and new one is started if either the Read or Write Index register is written before writing the third byte.

After the third byte of the sequence is written, the index register is automatically incremented to point to the next entry of the look-up table. If the index is FFh, it rolls over to 00h.

Both the Read and the Write index registers, physically map to a single index register. However only the Write Index register can be read. Reads from the Read Index register return the contents of Palette state register.

8.7.5 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 register and writes to this port modify the content of the look-up table entry pointed to by the Write Index register. Each look-up table entry is 24-bit wide and is read or written as a sequence of 3 bytes. The read or write sequence is always Red, Green and Blue. The normal procedure for accessing the look-up table is to initialize one of the Index registers and follow it with an uninterruptible sequence of 3 reads/writes from this register.

For VGA backward compatibility, when bits 2-0 of CR28 are programmed to 000 (as they are after reset), the palette look up table is treated as if each entry was only 18 bits wide. In this case, writes to port 3C9h map data such that bits 5-0 of host data are written into bits 7-2 of the look-up table while bits 1-0 are zeroed out. Similarly, reads return bits 7-2 of look-up table data onto host bits 5-0 and zero out bits 7-6.

When bit 5 of register CR3E is set to one, reads and writes to this port access red, green and blue signature data instead of look-up table data.

To minimize the sparkle while accessing the look-up table, all 3-bytes are read or written in a single video clock interrupting the screen repaint for one clock only. The interrupted pixel is painted with the same color as the previous pixel.

This register is not initialized by reset.

9 GRAPHICS ENGINE

9.1 Introduction

The Graphics Engine (GE) performs limited graphics drawing operations. The results of these operations change 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.

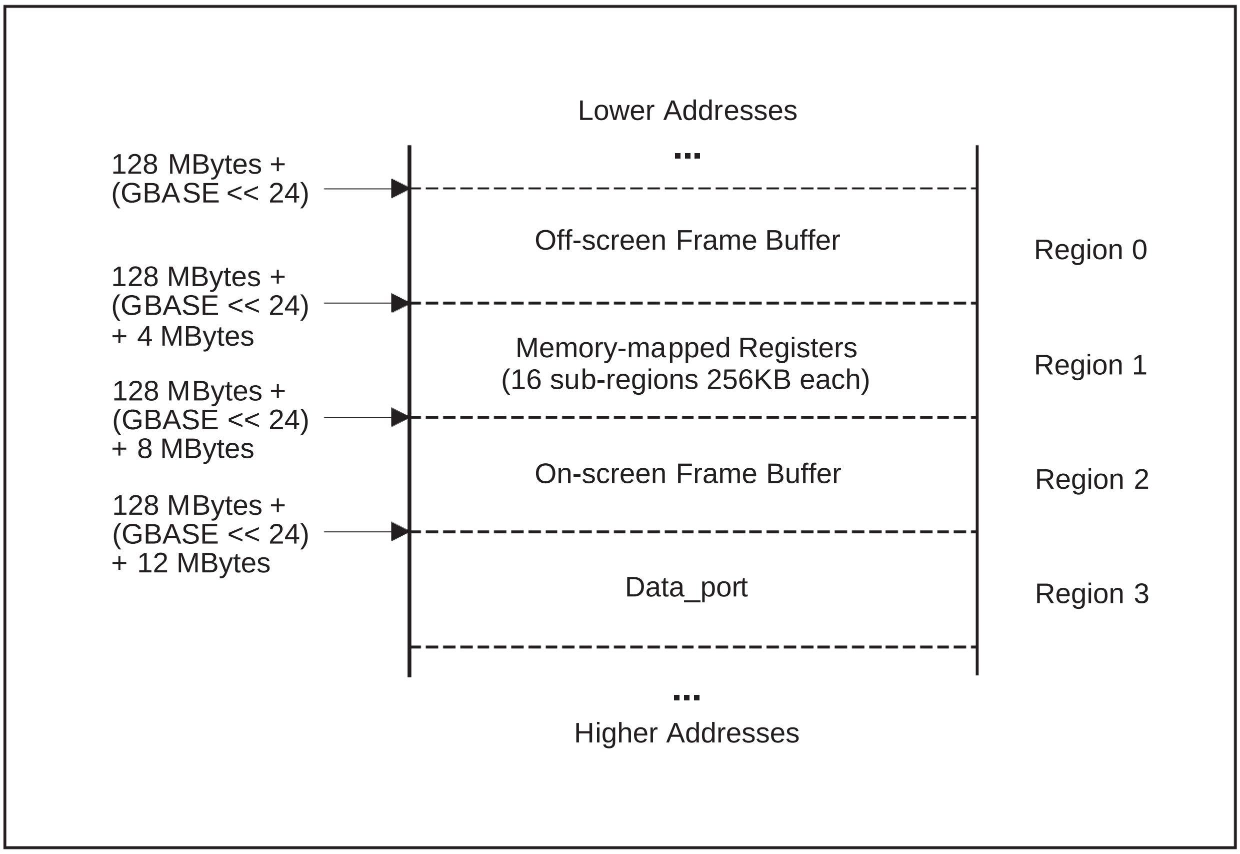
9.2 Memory address space:

The extended (non-VGA) graphics and video functions of Graphics Engine occupy 16 MBytes of memory address space. This space can located anywhere in the memory on any 16 Mbyte boundary above 128 MBytes. The 16 MByte region is divided into four parts as shown inFigure 10:

In this figure 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 access 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 accessing extended functionality in real mode.

Figure 10. GE memory Map


Lower Addresses
...
128 MBytes + (GBASE << 24)
Off-screen Frame Buffer
Region 0
128 MBytes + (GBASE << 24) + 4 MBytes
Memory-mapped Registers (16 sub-regions 256KB each)
Region 1
128 MBytes + (GBASE << 24) + 8 MBytes
On-screen Frame Buffer
Region 2
128 MBytes + (GBASE << 24) + 12 MBytes
Data_port
Region 3
...
Higher Addresses

Two 4-Mbyte regions are dedicated to frame buffer. Frame Buffer reads from either: 128Mbytes + (GBASE<<24) or 12Mbytes + (GBASE<<24) + 8MBytes are identical.

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 draw using an off-screen area address to a portion of the Frame Buffer that is currently being displayed.

The Frame Buffer addresses loaded into GE registers are from 0 to 4 MB. The source, pattern or destination of a GE operation can be located anywhere in physical 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 as follows:

Sub-region	Region function
0	2-D Graphics Engine registers
1	Extended CRTC registers
2	Video 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. Except for the VGA registers, Writes, must be done using double-word (32-bit) transfers. VGA registers may be written to via byte, word or double-word accesses.

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.

9.3 Dumb Frame Buffer Access

The CPU can access the frame buffer memory as ordinary memory. It can read from or write to the frame buffer using any memory access instruction, and any data width. Thus, the CPU can access the frame buffer as if it were an unaccelerated display subsystem. This access by the CPU is permitted regardless of the GE’s busy status. Therefore, software must be careful to avoid race conditions or clashes if writing to the frame buffer when the GE is busy.

9.4 Addressing

The GE's frame buffer and extended registers may be accessed by the CPU via two methods: extended addresses, or A0000h-AFFFFh addresses. The former method allows direct access to the 16 MByte GE address range via 32-bit addresses. The A0000h-AFFFFh addressing method maps a 64K window of the 16 MByte GE address space into the address range A0000h-AFFFFh. For additional detail on the A0000h-AFFFFh addressing, see CR21 Graphics Extended Aperture Register.

The Frame Buffer address used by the host also informs the GE whether the access is being made to an on-screen or off-screen drawing area. It is important that the software never draw 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 OS address translation and GE host address remapping is done. The Frame Buffer addresses loaded into GE registers are from 0 to 4 MB. The source, pattern or destination of a GE operation can be located anywhere in the Frame Buffer space.

9.5 VGA Operations

The GE fully supports VGA operations. The standard two read modes and four write modes of the VGA frame buffer are implemented. While, there is no “GE mode” bit that places the drawing operations in a VGA mode, the Foreground and Background color registers must be set to all ‘1’s and all ‘0’s, respectively, before issuing standard VGA write operations.

9.5.1 Addressing for reads and writes

Host addresses are mapped into display memory addresses according to the VGA addressing mode selected.

Multiple byte reads and writes are broken into single byte operations by the GE. For multi-byte reads, all of the bytes are sequenced through by the GE which performs each read and store the resulting byte in correct byte of an VGA read register. After the last read is done, the GE will provide the CPU with the requested data.

For multi-byte writes, all the GE sequences through the CPU data bytes, performing a 32-bit write to memory for each byte. No optimization is done by the GE to merge 32-bit Dword writes. (Such optimization would only be possible in the Chain-4 addressing modes.)

9.5.1.1 Read Mode 0

Read data is read into the GE. The byte selected by the read-map-select register is replicated into all four byte positions to form the resulting Dword, which is presented to the CPU.

9.5.1.2 Read Mode 1

Read data is read into the GE. The color-compare assembler takes the appropriate 4 bit sets from the output and groups them into one byte. This byte is then replicated into all four byte positions to form the resulting Dword, which is presented to the CPU.

9.5.1.3 Write Mode 0

One of the bytes of the write data, depending upon the byte address, is selected and rotated 0-7 bits. The resulting byte is then output as all four bytes of the result. In parallel, the four set/reset mask bits are each expanded to a byte. The four set/reset enable bits select either the rotated write data or the expanded set/reset bytes, on a byte-by-byte basis. The VGA ROP is then applied to the selected data and the latched read data. A bit mask controls which bits of all bytes are altered and a map mask controls which bytes are written.

9.5.1.4 Write Mode 1

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

9.5.1.5 Write Mode 2

Write mode 2 is similar to write mode 0, except that only data from the bit-alignment is used. In the bit alignment unit, the four least significant bits of the current byte are selected. Since the bit ordering is different for this operation than Windows bitmaps, the stutterer/selector must reverse the order of the four bits. Bit-expander expands the bits into the FG or BG, which software will have set to all ‘1’s or ‘0’s, respectively.

The bit-mask, VGA logic-operation, and map-masks are implemented in the same way as Write Mode 0.

9.5.1.6 Write Mode 3

Write mode 3 is similar to write mode 0, except that the CPU data byte is rotated and anded with the contents of the bit mask register to form the bit mask. The set/reset mask bits are expanded to one byte each and used regardless of the state of the set/reset enable bits.

9.5.2 Operand Sources

The GE operates on data which can originate in one of three possible areas:

1) 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) host-supplied data

3) on-chip color registers

9.5.2.1 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) Source, 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) Pattern, must originate from the frame buffer

3) Destination, must originate from the frame buffer

One or more of these operands are the input 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 to prevent wasted performance due to needless operand fetching.

If the ROP requires destination data reads, then the "Dst" field of the ROP register must be set to '1'. If destination reads are not required, then this field should be set to '0'.

If the ROP requires pattern data or uses color transparent mode, then the "Pat" field of the ROP register must be set to '1'. If no pattern or color transparency is being used in operation, then this field should be set to '0'.

9.5.2.2 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; and

2) Pattern transparency, where any Pattern bytes that are '0' suppresses writing to the corresponding destination pixel bytes.

3) Source transparency, where any Source pixels that match the value of the source transparency register are not drawn.

4) Destination transparency, where any Destination pixels that match the value of the source transparency register are not drawn.

These modes are controlled by fields in the ROP register.

9.5.3 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 offsets from this base address, and a pitch for that region. The pitch is the byte distance between two pixels that 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. A 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 of 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 depends on the X_dir and Y_dir fields of the Pixel_depth 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

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:

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).

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

9.5.3.1 Command Initiation

The destination coordinate register, Dst_XY, appears multiple times in the address space. Reading from any of these appearances, or aliases, are equivalent. Writing to most of these aliases also has the side-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 in the following manner:

4-MByte region 1 (memory mapped regs)

256KByte sub-region 0

3 2	2 2	22	2 1	1 1	1 1	1	
1 7	6 4	32	1 8	7 6	5 4	3 2	1 0
00001	<GBASE>	01	0000	01	<Cmd>	<Count>	10

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. Cmd is one of the following:

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)

When the CPU writes to the Dst_XY register using one of these operation aliases, the register write is completed and then the associated operation is begun. Thus, to perform an operation, the CPU writes to all but the Destination register. Then the last write is done to the Dst_XY register using one of the above aliases.

The ROP register also has fields that control the Source data (Screen, Host or Foreground color register), enables/disables bitmap expansion, determines if the drawing is done in one of the transparent modes.

Screen-to-screen BitBlts are done as a BitBlt with the Source set as the Screen and bitmap expansion disabled.

Host-to-screen BitBlts are done as a BitBlt with the Source set as Host and the bitmap expansion disabled.

Rectangular fills are done as a BitBlt with the Source set to the Foreground color register and the bitmap expansion disabled.

Text drawing using bit-packed font data provided by the Host is done as BitBlt with the Source set as Host and the bitmap expansion enabled. A transparent mode may also be specified.

Lines are not generally supported by the GE, but horizontal and vertical line segments can be quickly implemented as Width-specified BitBlts with the Height register set to 0 (to indicate a single pixel high BitBlt), or as Height-specified BitBlts with the Width register set to one less than the pixel depth.

9.5.4 Drawing Engine Registers

The software controls graphics 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.

Except for the VGA registers, Writes, must be done using double-word (32-bit) transfers. There are a few extended registers for which only the lower 16 bits are defined. For these, 16-bit writes are acceptable. VGA registers may be written to via byte, word or double-word accesses.

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.

9.5.5 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:

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>	01	0000	000000	<Regoffset>

Where “Regoffset” 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 Regoffset will always be ‘00’. The following sections will list the “regoffset” 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, accepting the short possible periods of being held.

The GE Status register may be accessed at any time.

9.5.5.1 Data Port Access

The CPU writes Host data to the GE through the Data Port for Text and Host-to-screen BitBlt operations. The Data Port appears as one of the registers, as discussed in the previous section. Behind the Data Port the Data FIFO buffers incoming data from the CPU. The Data Port is also repeatedly aliased in the upper 4 MBytes of the GE’s 16 MByte address space.

In normal operation for text drawing (done as a bitmap expanded Host-to-screen BitBlts), the CPU-writes enough data to the Data Port for the current character, starts on the next text character by writing to the Dst_XY register, and then 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 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), then the GE forces a CPU retry via the backoff (BOFF) mechanism.

9.5.6 REGISTER SPECIFICATION

The GE registers are listed in alphabetical order and defined below.

9.5.6.1 Background Regoffset = 004h

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 contents of this register are not altered by drawing operations.
The contents of this register are defined to be zero after reset.

9.5.6.2 Cursor_XY Regoffset = 11Ch

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.

Note: 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.
The contents of this register are not defined after reset.

9.5.6.3 Data_Port Regoffset = 804h

This write-only register is the port through which the CPU provides Host data.

Bits 31-0 **Top of Data FIFO.**

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.

9.5.6.4 Dst_base Regoffset = 018h

This register specifies the starting DRAM linear address of the destination operand (aligned to a 32 byte boundary).

Bits 31-21 *Reserved.*

Bits 21-5 **DstOp_Base.** Base DRAM linear address of the destination operand.

Bits 4-0 *Reserved.* These bits are set to ‘0’.

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

9.5.6.5 Dst_pitch Regoffset = 028h

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.

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 the following table for the multiplication values that this field can specify.

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

9.5.6.6 Dst_XY

This register contains the coordinate address of the starting corner of the destination operand. Which corner is the “starting” corner is controlled by the X_dir and Y_dst_dir fields of the Pixel_depth register.

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 specifically to accelerate the Text and Line Segment 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 address of this register is described in Section 9.5.5.

See Section 9.5.5 for a description of these fields.

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

9.5.6.7 Foreground Regoffset = 034h

This register contains the full-color value(s) that a “1” bit is expanded to. 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.

Bits 31-0 **Foreground Color.** This is the color to be used as the foreground when expanding bitmap ‘1’ values

The contents of this register are not altered by drawing operations.
The contents of this register are defined to be FFFFFFFFh.

9.5.6.8 Height Regoffset = 048h

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.

This register can also 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 Section 4.8.1, “Command Initiation”.

This register is double-buffered. While the GE is busy executing one operation, a new value may be safely written to this register for the next operation. If the Dst_XY double-buffered register is full, then the CPU will be held off. This feature is specifically to accelerate the Text and Line Segment operations.

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.

This register can be accessed via 32-bit or 16-bit transfers.

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

9.5.6.9 Pattern Regoffset = 058h

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.

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 9.5.8.1, “Pattern Data”.

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 bitsmust be set to:

(Dst_X / 8) % 3

where Dst_X is the byte address of the first 24-bit pixel in the destination row. For all other pixel depths, must be set to “00”.

Bits 2-0 *Reserved.*

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

9.5.6.10 Pixel_depth Regoffset = 07Ch

This register contains the number of bytes in a pixel, and bits that control the direction of Screen-to-screen Blts.

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 addresses to larger linear addresses). Setting this bit to ‘1’ reverses the direction of BitBlt source operations. This bitshould 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 linear addresses 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’ if pixels advance from left to right, and to ‘1’ if 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:

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

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.

The contents of this register are not defined after reset.

9.5.6.11 ROP Regoffset = 08Ch

This register contains the ROP code to be applied during processing a pixel, enables bitmap expansion, selects transparent modes, and controls the source operand.

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:

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 Op requires no SRC operand (such as in inverting the destination as well as constant fills). Failure to set this field correctly can result in a degradation of performance.

Note that CPU writes to the Data Port will complete without error, and the data will be ignored unless the Source field is set to HOST.

Bit 29 **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 destination data is to be read during the operation.

Bit 27 **Diagnostic Mode,** for normal operation 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 buffer vs. the DST transparency compare register. The results of the comparison are interpreted based upon the DST transparency match bit. Note that the DST present bit must also be set to ‘1’ when this bit is set. This mode is not valid with 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 this 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 transparency compare register. The results of the comparison are interpreted based upon the SRC transparency match bit. This mode is only meaningful when using non-bitmap screen or host data as the source. Transparency for bitmap source data should not use this mode, but rather the SRC bitmap transparency mode. This mode is not valid with 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’, itmap expansion will be enabled and source data from screen or host is assumed to be bitmap data. If set to ‘0’, ource 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 altered by drawing operations.
The contents of this register are not defined after reset.

9.5.6.12 Src_base Regoffset = 098h

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 altered by drawing operations.

The contents of this register are not defined after reset.

9.5.6.13 Src_pitch Regoffset = 0ACh

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:

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

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 below table for the multiplication values that this field can specify.

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.

This register can be accessed via 32-bit or 16-bit transfers.

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

The contents of this register are not defined after reset.

9.5.6.14 Src_XY Regoffset = 0BCh

This register contains the coordinate address of the starting corner of the source operand. The corner which is the “starting” corner is controlled by the X_dir and Y_src_dir fields of the Pixel_depth register.

This register is double-buffered. While the GE is busy executing one operation, a new value may be safely written to this register for the next operation. If the Dst_XY double-buffered register is full, then the CPU will be held off.

The Y field and all but the lower 3 bits (5 when bitmap expansion enabled) of the X field are ignored during Host-to-screen BitBlts.

Bits 32-16 **Src_Y.** The unsigned Y coordinate of the starting corner of the source operand.

Bits 15-0 **Src_X.** The unsigned X location of the starting corner of the source operand. When bitmap expansion is not enabled, this is a byte address. When in bitmap expansion is enabled, this is a bit address.

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

The contents of this register are not defined after reset.

9.5.6.15 Status Regoffset = 908h

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. GE register writes that are done to registers when this bit is set may result in the CPU being held for the duration of the current operation. GE register reads always return without holding the CPU, but the 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.

9.5.6.16 Width Regoffset = 0C8h

This register contains the one less than the number of bytes across 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 inSection 9.5.3.1, “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 bytes specified in this field will be modified.

This register can be accessed via 32-bit or 16-bit transfers.

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

9.5.6.17 Xtra Regoffset = 0D4h

This register contains 32 bits of data that software can read and write.

This register has no effect on any drawing operation or display.

Bits 31-0 **Data for user software use.**

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

The contents of this register are not defined after reset.

9.5.6.18 SRC Transparency Compare Regoffset = ECh

This 32-bit register contains the pixel value used for comparison in SRC transparency mode. For pixels depths of 1 byte per pixel, the pixel value must be replicated in all four bytes of this register. For pixel depths of 2 bytes per pixel, the pixel value must be replicated in the upper and lower 16 bits of this register.

9.5.6.19 DST Transparency CompareRegoffset = FCh

This 32-bit register contains the pixel value used for comparison in DST transparency mode. For pixels depths of 1 byte per pixel, the pixel value must be replicated in all four bytes of this register. For pixel depths of 2 bytes per pixel, the pixel value must be replicated in the upper and lower 16 bits of this register.

9.5.7 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.

9.5.8 GE Operations

9.5.8.1 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) % 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.

9.5.8.2 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 significant. The first bit after those skipped will then be aligned to the first destination pixel.

For bitmap expansion, the X_dir must be ‘0’. The result for a bitmap expansion BitBlt with X_dir set to ‘1’ is not defined.

With the X_dir field of the Pixel_depth register set to ‘0’, the bitmap is considered to start at the least significant end of the first quad-word and continues towards the most significant end of the quad-word and then to higher memory addresses. The first bit of a quad-word is bit 7 of byte 0 and the last bit is bit 0 of byte 7.

9.5.9 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

9.5.9.1 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 raster operation. 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 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 color 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 register 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 is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address. 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 buffer at its full bandwidth. Thus, result pixels are computed in groups of 32 bits, to allow one 64-bit result every 2 video domain clock cycles. As the memory subsystem supports separate byte-write enables, the first and last 64-bit write in each scan line can be performed without requiring a read-modify-write cycle.

9.5.9.2 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)) -> (Destination)

If these rectangular areas are overlapping, then the direction of the BitBlt must be carefully selected:

* *Source region is below the destination:*

> BitBlt should be done from the upper-left-hand corner and progress downwards,

> the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register should be “0”

* *Source region is above the destination:*

> BitBlt should be done from the lower-right-hand corner and progress upwards,

> the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register should be “1”

The Source, destination and pattern data can be anywhere in the frame buffer. ROP may be any 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 is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address. This write starts the BitBlt.

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

The implementation of this operation is performance optimized and can drive the DRAM buffer at its full bandwidth during constant fills. Thus, result pixels are computed in groups of 32 bits, to allow one 64-bit result every 2 graphics clock domain cycles. As the memory subsystem supports separate byte-write enables, the first and last 64-bit write in each scan line can be performed without requiring a read-modify-write cycle.

9.5.9.3 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 CPU has built a rectangle in the frame buffer memory area with the Host data, then the Screen-to-screen BitBlt operation can be used instead of this operation.

The pixel depth of the Host data must match that of the Destination region, unless it is a bitmap (if bitmap expansion is specified).

The function performed during the BitBlt is:

ROP ((Host), (Pattern), (Destination)) -> (Destination)

The host area data is supplied by the CPU, which writes its data into the Data Port. The destination and pattern data can be anywhere in the frame buffer.

ROP may be any of the 256 standard raster operations.

The CPU specifies the number of least significant bytes of the first double-word that should be discarded, via the least significant 3 bits of the X field in the Src_XY register. The GE then merges bytes of two double-words at a time, in order to build a double-word to operate on. For example, if the X field was set to 3, then the last byte of the first double-word and the first three bytes of the second double-word would be combined to form the first Host data double-word:

	Double Word 1				Double Word 0			
Byte:	7	6	5	4	3	2	1	0

Byte:	6	5	4	3

The CPU must provide the number of words required for Height * Width pixels. At the end of a scan line, the GE will discard the excess Host bytes or bits that may be left in the last double-word and advance to the next scan line, unless Src_pitch is set to 0. In this case, data for adjacent scan lines are contiguous in the host data stream.

To perform a Host-to-screen BitBlt (except for the last write, order is unimportant):

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

4) Set Src_pitch to 0 if packed source data or to a non-zero value, otherwise

5) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to ‘0’

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 HOST

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) Set Src_XY to indicate alignment of the Host data (only the least 3 significant bits are important)

14) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address, starting the BitBlt.

15) 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.

9.5.9.4 Packed Text

The Packed Text operation is used to efficiently expand packed bitmap fonts into full color representations in the frame buffer (either on-screen or off-screen areas). This operation is implemented as a Host-to-screen BitBlt with bitmap expansion and packed source data. The next section discusses how to handle Microsoft Font Text operations.

The function performed during the Packed Text operation is:

ROP ((Host), (Pattern), (Destination)) -> (Destination)

The Host packed bitmap data is supplied by the CPU via writes to the Data Port and is 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 of the 256 standard raster operations.

In a standard bitmap, the start of each scan line is aligned to a pitch-specified boundary. This is acceptable for wide bitmaps, however text font bitmaps are usually not very wide. To increase the amount of information provided to the GE per Host write, the Text operands are bit-packed. Each 32-bit write contains only useful font data, except possibly for the trailing bits of the last write. For example, question mark character might appear in a fictitious font as:

```
 -----                                -----
| *** |                              |01110|
|*   *|                              |10001|
|   * |                              |00010|
|  *  |   Or, in binary form:        |00100|
|     |                              |00000|
|  *  |                              |00100|
 -----                                -----
```

This would appear in memory as:

top line					bottom line
01110	10001	00010	00100	00000	00100

Increasing memory addresses ------>

Breaking this up into bytes:

	01110100	01000100	01000000	000100XX
Byte	0	1	2	3

Breaking this up into a double-word:

	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

4) Set the ROP.Packed to 1 for packed source data

5) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, Pixel_depth.Y_dst_dir to ‘0’

6) Set Height to the height of the destination character, in scan lines

7) Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values

8) Set ROP.Source to HOST

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) 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 is written to using the Width-specified address alias and the width of the destination character (in bytes) in the Count field of the address, starting the BitBlt.

14) 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.

9.5.9.5 Microsoft Font Text

Microsoft fonts (consisting of 8-bit stripes of a character) can be handled as simple 8-pixel-wide Host-to-screen BitBlts with bitmap expansion, but no packed data. The last stripe of a character is handled in a different manner. The background color for the last stripe is first filled into its rectangular area. Then the stripe data is drawn in transparent mode with the unused bits filled with zeros.

9.5.9.6 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 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 the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register must be set to “0”.

The Background color register should be set to the color of the line segment to be drawn.

9.5.10 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:

AND Value	XOR Value	Displayed Pixel
0	0	Cursor_color0
0	0	Cursor_color1
1	1	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 + 63)

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.

10 VIDEO CONTROLLER

10.1 Introduction

The STPC Client controls video signal inout, 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.

10.1.1 The Video Input Port

The Video Input Port interfaces to external video in several digital formats, it is also compatible with VIP 1.0 compatible video streams.

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.

Two DMA channels allow sliced Vertical Blanking Data (VBI) data and compressed audio/video bitstreams to be automatically transferred between System Memory and the VIP Host Port. Both channels operate from a linked list of instructions in system memory allowing data transfers between non-contiguous memory blocks (scatter and gather operations). Each channel has a 32-byte FIFO to optimize transfers to system memory while insuring adequate bandwidth for VIP transfers.
Both channels support Hardware Polling with a programmable delay period to reduce polling when the selected target FIFO is not ready. Hardware polling minimizes transfer startup time for the DMA operations.

Arbitration between DMA Channels and host accesses can be round-robin or priority based. Round-robin arbitration and maximum burst length controls allow the maximum latency to be calculated and controlled. Priority based arbitration insures maximum bandwidth for critical tasks. The time-out period is programmable to accommodate devices with long access times.

10.1.2 The Digital Video Output Port

……..……

10.1.3 The Video Accelerator Registers

The video input and display is controlled through the Video Accelerator registers. These registers provide the settings for the display buffer areas, filter control, color mixing and color space mixing.

10.2 Video Input Port

10.2.1 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.

10.2.2 Digital Video Input Formats

The video input port can be programmed to decode one of several video formats. The following sections discuss this functionality in more detail.

Existing video input formats include:

Input Format	Description
VIP 1.0 (ITU-R 656)	Lock internal timing generator to EAV codes
STi3520A compatibility mode	8-bit multiplexed CCIR 601
Passthrough	Video stream is passed directly through the Video Ports without processing

Note: VIP 1.0 support replaces ITU-R 656 mode. The primary difference between the two implementations is that the existing mode locks the internal timing generator to the EAV codes and uses it to generate the horizontal and vertical active windows. In a VIP 1.0 environment this information must be extracted from the SAV and EAV codes since the VIP Target is only required to send SAV and EAV codes during active video.

10.2.2.1 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 and VBI data must be based solely on the SAV and EAV codes embedded in the video stream. The VBI data is digitized and captured at video frequencies and must be post processed by the driver to recover the desired information.

This implementation includes:

– The most significant bit of the SAV/EAV code may be used to differentiate between VBI data and video or between two video streams.

– The Video Timing Generator may be used independently in this mode, to generate the system timing signals, HSYNC# and B/T#.

– Horizontal and vertical active window based on SAV and EAV codes only

– Byte swapping may be disabled based on the Task Bit from the SAV code.

– Invalid pixel detection when the value is 0x00. When a pixel data value of 0x00 is encountered during an active line, the data is not written to the frame buffer and the pointer is not incremented. This allows re-sampled video to be output without changing the PIXCLK frequency.

10.2.2.2 STi3520A Compatibility Mode (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 STi3520A outputs video data in 4:2:2 format, multiplexed to 8 bit data words in Cb, Y, Cr, Y format. The STi3520A uses input signals field (B/T#) and horizontal sync (HSYNC#) to generate video timing. The STPC Client Video Controller can be configured to generate video timing (driving HSYNC# and B/T#) or lock to these signals when generated by an external video timing source. The OSD signal is not supported.

10.2.2.3 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 MPEG decoder 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.

10.2.2.4 Standard ITU-R-656 Format

This mode is replaced by VIP 1.0 compatible mode. See above.

10.2.3 VIP Specifications Not Supported

10.2.3.1 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. .

10.2.3.2 DMA Channel Restrictions

DMA channel 1 is provided and optimized for transferring compressed audio and video data to an MPEG device. It reads from System Memory and writes to VIP Host FIFO Space.

DMA Channel 2 is provided and optimized for the transfer of sliced VBI data. It reads from VIP Host FIFO space and writes to System Memory.

The DMA controllers are only capable of accessing Host FIFO Space. They are not capable of accessing Host Register Space.

10.2.3.3 Chroma Mask

There is no support provided for chroma key mixing since the SIP CRTC supports both color and chroma based video mixing.

10.3 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.

Table 13. Video Input Module Address Space

AD[31:28]	[27:24]	[23:20]	[19:18]	[17:15]	Description
1xxx	GBASE	0110	00	000	VIP Host Target Address Space - VIPCLK
1xxx	GBASE	0110	00	001	VIP Host Port/DMA Registers - VIPCLK
1xxx	GBASE	0110	00	01x	Reserved
1xxx	GBASE	0110	00	100	Video Input Port Registers - PIXCLK domain
1xxx	GBASE	0110	00	11x	Reserved
1xxx	GBASE	0110	00	1x1	Reserved

10.3.1 VIP Host Target Address Space

The VIP Host Address Space is subdivided into eight regions, providing Register and FIFO address space for four different VIP Target devices as shown in Table 14. The least significant 12-bits of the System Byte Address are mapped directly to the VIP Target Address (A[11:0]). For Register accesses, all 12 of the address bits are sent to the Target. During FIFO transfers the upper four address bits (A[11:8]) are sent as the FIFO address.

The start of VIP Host Target Address Space is at (GBASE << 24) + 600000h.

Table 14. VIP Address Space

AD[31:28]	[27:24]	[23:20]	[21:16]	[15:12]	Description
1xxx	GBASE	0110	0000	0000	VIP Device 0 Register Space
1xxx	GBASE	0110	0000	0001	VIP Device 0 FIFO Space
1xxx	GBASE	0110	0000	0010	VIP Device 1 Register Space
1xxx	GBASE	0110	0000	0011	VIP Device 1 FIFO Space
1xxx	GBASE	0110	0000	0100	VIP Device 2 Register Space
1xxx	GBASE	0110	0000	0101	VIP Device 2 FIFO Space
1xxx	GBASE	0110	0000	0110	VIP Device 3 Register Space
1xxx	GBASE	0110	0000	0111	VIP Device 3 FIFO Space

10.3.2 VIP Host Port and DMA Registers

The Host Port and DMA Channel Registers are listed in Table 15.

The Host Port Configuration Register specifies the arbitration method and time-out latency used for Host Port accesses.

The DMA registers are used to control DMA transfers between System Memory and VIP Target FIFO Space. The Channel Configuration Register (ChCfg) and Channel Status Register (ChStat) are shared by both channels. In addition, each DMA channel has three dedicated registers containing the Channel Command Pointer, Current Instruction and System Memory Address.

VIP Host Control Registers start at (GBASE << 24) + 608000h. The offsets in the Figure 3 should be added to this offset.

Table 15. VIP Host Control Registers

Addr. Offset	Name	Mnemonic	Description
000h	Host Port Configuration Register	HstCfg	Configure Host Port Operation
000h	Channel Configuration Register	ChCfg	Static Configuration for DMA Controllers
004h	Channel Status Register	ChStat	Status Information for DMA Controllers
00Ch	Channel 1 Command Pointer	Ch1InstPtr	Channel 1 Command Pointer
010	Channel 1 Instruction Register	Ch1Inst	Channel 1 DMA Instruction
014	Channel 1 System Address Register	Ch1SysAdr	Channel 1 System address
02C	Channel 2 Command Pointer	Ch2InstPtr	Channel 2 Command Pointer
030	Channel 2 Instruction Register	Ch2Inst	Channel 2 DMA Instruction
034h	Channel 2 System Address Register	Ch2SysAdr	Channel 2 System address

10.3.3 Video Input Port Configuration Registers

Table 16 shows the Video Input Port Configuration Registers. Registers that are new or have modifications from the SIP implementation are shaded.

Video Input Port Registers start at (GBASE << 24) + 620000h. The offsets in the Figure 4 should be added to this offset.

Table 16. Video Input Port Configuration Registers

Addr. Offset	Name	Mnemonic	Description
000h	Frame Buffer Address Readback	fb1_adr	Current frame buffer address pointer
004h	Video Input Port Configuration	vin_cfg	Video input port configuration
008h	Video Input Port Status	vin_stat	Video input port status
00Ch	Video Input Address 0	vin_ad0	Frame buffer address 0
010h	Video Input Address 1	vin_ad1	Frame buffer address 1
014h	Video Input Destination Pitch	vin_dp	End of line address increment
028h	External Timing Configuration 1	vtg_ext1	VTG Configuration register 1
02Ch	External Timing Configuration 2	vtg_ext2	VTG Configuration register 2
030h	Horizontal Timing Generator	vtg_ht	VTG Horizontal timing
034h	Vertical Timing Generator	vtg_vg	VTG Vertical timing

10.4 VIP Host Port

The VIP Host Port is the interface between the Video Controller and VIP slave devices. It provides a 32-bit interface to the host and two DMA channels, handling the transfer protocol and parallel/serial conversion required by the VIP Host Port. It also contains time-out, arbitration and burst length control to allow automatic detection and configuration by software.

10.4.1 VIP Host Configuration Register (HstCfg)

The Host Configuration Register controls the operation of the VIP Host Port.

Bits 31-28 **Target Time-out Count, Tmo_cnt[3-0].** This field specifies the maximum number of data phases that a VIP Target Device can delay a transfer without causing a time-out.

At system startup time the driver tries to read configuration space for each of the four possible VIP target devices. Those that are present return device specific configuration information that is used to request system resources and configure the driver. When the device being addressed is not present, the transfer times out and the Host Port returns the value FFh for all of the requested bytes. The time allowed for any device to respond is specified by Tmo_cnt. Host port accesses that terminate due to a time-out are not repeated. The default value for Tmo_cnt is Fh corresponding to 60 VIPCLK periods (about 2.4µs).

Bits 27-24 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

Bit 23 **Arbitration Type, Arb_tp.** This bit controls the arbitration mode for the host accesses and DMA Channels which compete for access to the VIP Host Port (Table 17).This bit and the Arbitration Priority bit determines which will get the next access when multiple requests are pending.

In Round Robin mode, the three ports are sampled in order with the access granted to the next one in the sequence that is requesting a transfer. In this mode the maximum latency for any port is deterministic and can be calculated. This is likely to be the mode commonly used.

Table 17. Arbitration Type

Bit 23	Arbitration Type
0	Round Robin
1	Priority Based Arbitration

Bits 22-20 **Channel Priority, Arb_pri.** This field specifies the priority attached to host and DMA channels when priority based arbitration is selected

The priority is determined according to the following chart. In this mode the highest priority port is guaranteed the maximum access to the VIP port, at the potential expense of the other ports.Table 18 shows the priority granted to the three ports based on Arb_pri with 1 corresponding to the highest priority and 3 to the lowest.

Table 18. Arbitration Priority Selection

Arb_pri[2:0]	Host Priority	DMA Channel 1 Priority	DMA Channel 2 Priority
000	1	2	3
001	1	3	2
010	2	1	3
011	3	1	2
100	2	3	1
101	3	2	1
110	Undefined	Undefined	Undefined
111	Undefined	Undefined	Undefined

When priority based arbitration is used it is important to specify a non-zero value for the Target Busy Delay in the DMA Channel Configuration Register to prevent hardware polling while the target device is busy from locking out other requests.

Bits 19-14 **BurstLength, Brst_len[5-0].** This field specifies the maximum burst length in bytes per transfer in Round Robin mode. The maximum number of bytes transferred per host request is equal to Brst_len + 1.

Bits 13-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

10.5 DMA Channel Operation

Two DMA Channels are provided to handle transfers between System Memory and VIP Host FIFO Space. Channel 1 reads from System Memory and writes to Host FIFO Space. Channel 2 reads from Host FIFO Space and writes to System Memory.

10.5.1 DMA Channel Configuration Register (ChCfg)

The channel configuration register contains static configuration information common to both DMA channels. The contents of this register are set to 0 at reset.

Bit 31 **Channel 1 Interrupt Enable, Ch1IntEn.** If set to '1', channel 1 generates an interrupt on an error condition and at the end of a DMA transfer if the Interrupt on Completion bit of the instruction is asserted.

Bits 30-22 **DMA Target Busy Delay, Ch1TBDly[8-0].** Length of time in VIPCLK periods that DMA Channel 1 waits before retrying a DMA request when the VIP Target Device terminates without accepting data. (511 = 15.33µs @ 33 Mhz) This delay reduces the number of Host Port bus cycles that are wasted when the VIP Target is busy. When this field is set to zero, the DMA Channel makes a new request immediately if the DMA instruction has not been completed.

Bits 21-18 *Reserved.* This Read-Only field is reserved. When read it returns '0's.

Bits 17-16] **Transfer Threshold Ch1, Ch1WMk[1-0]**. Threshold in quad words for requesting a transfer between the Channel 1 FIFO and System Memory. When the DMA Channel is reading from System Memory, a request is made when the threshold level in quad words is larger than the number of bytes in the DMA Channel FIFO. When the DMA Channel is writing to System Memory, a request is made when the number of quad words in the FIFO is greater than or equal to the threshold level.

Bit 15 **Channel 1 Interrupt Enable, Ch2IntEn.** If set to '1', channel 1 generates an interrupt on an error condition and at the end of a DMA transfer if the Interrupt on Completion bit of the instruction is asserted.

Bits 14-6 **DMA Target Busy Delay, Ch2TBDly[8-0].** Length of time in VIPCLK periods that DMA Channel 2 waits before retrying a DMA request when the VIP Target Device terminates without accepting data. (511 = 15.33µs @ 33 Mhz)

Bits 5-2 *Reserved.* This Read-Only field is reserved. When read it returns '0's.

Bits 1-0 **Transfer Threshold Ch1, Ch2WtMk[1-0].** Threshold in quad words for requesting a transfer between the Channel 2 FIFO and System Memory. When the DMA Channel is reading from System Memory, a request is made when the threshold level in quad words is larger than the number of bytes in the DMA Channel FIFO. When the DMA Channel is writing to System Memory, a request is made when the number of quad words in the FIFO is greater than or equal to the threshold level.

10.5.2 DMA Channel Status Register (ChStat)

The channel status register contains information about the DMA operations. The currently defined fields are described below. The Target Time-out and Interrupt Pending bits are cleared by writing a zero to appropriate bit location. This register defaults to 0 at reset.

Bit 15 **Channel 1 Active, Ch1Act.** This read-only status bit is set to one when DMA channel 1 has an operation in progress. It is set to zero at the end of a DMA operation when the Instruction Fetch Enable is de-asserted and when a DMA Channel Error occurs.

Bit 14 **Channel 1 Interrupt Pending, Ch1Int.** An interrupt is generated at the end of a DMA instruction if the Channel Interrupt Enable is set to one and the Interrupt on Completion bit in the DMA instruction is set to one. When interrupts are enabled, a DMA Channel Error will also generate an interrupt since it terminates the current DMA instruction stream.

Bit 13 **Channel 1 Target Time-out, Ch1TTO.**This bit is set to one when the VIP target selected by the instruction does not respond within the number of data phases specified by the Tmo_cnt field of the HstCfg register. Target Time-out status terminates the DMA Channel operation and generates an interrupt if the Channel Interrupt Enable is asserted.
This error can occur when the VIP Device Select indicates a non-existent VIP Target or when the VIP FIFO Select points to a non-existent FIFO within the selected target.

Bits 12-8 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

Bit 7 **Channel 2 Active, Ch2Act.** This read-only status bit is set to one when DMA channel 2 has an operation in progress. It is set to zero at the end of a DMA operation when the Instruction Fetch Enable is de-asserted and when a DMA Channel Error occurs.

Bit 6 **Channel 2 Interrupt Pending, Ch2Int.** An interrupt is generated at the end of a DMA instruction if the Channel Interrupt Enable is set to one and the Interrupt on Completion bit in the DMA instruction is set to one. When interrupts are enabled, a DMA Channel Error will also generate an interrupt since it terminates the current DMA instruction stream.

Bit 5 **Channel 2 Target Time-out, Ch2TTO.**This bit is set to one when the VIP target selected by the instruction does not respond within the number of data phases specified by the Tmo_cnt field of the HstCfg register. Target Time-out status terminates the DMA Channel operation and generates an interrupt if the Channel Interrupt Enable is asserted.
This error can occur when the VIP Device Select indicates a non-existent VIP Target or when the VIP FIFO Select points to a non-existent FIFO within the selected target.

Bits 4-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

10.5.3 DMA Channel Command Pointers (Ch1CmdPtr, Ch2CmdPtr)

The Channel Command Pointer registers (Ch1CmdPtr and Ch2CmdPtr) contain the system memory address of the next Channel Command in and System Address Pointer. The Channel Command is updated from the double word located at ChCmdPtr and the System Address Register is updated from the double word at ChCmdPtr + 4. Channel Command/System Address pairs must be quad word aligned as the instruction fetch sequencer always fetches and updates both registers.

Bits 31-3 **DMA Command Pointer.**This field contains the pointer to the Quad Word Channel. These bits are undefined after reset.

Bits 2-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

10.5.4 DMA Channel Instruction Registers (Ch1Inst, Ch2Inst)

The Channel Instruction Registers contain the current or most recently executed DMA instruction. The format is identical for both channels.

Bit 31 **Channel Enable, Ch1En (Ch2En).** The Ch1En (Ch2En) bit enables operation for DMA channels 1 and 2. The associated DMA channel starts operation when the enable is set to one and terminates it when the requested number of transfers is complete and the Instruction Fetch Enable field of the Channel Instruction is zero.

Writing a 0 to the Channel Enable causes the selected DMA channel to reset immediately, terminating any transfer currently in progress. In this case, results of any currently executing channel instruction are undefined, fields will be preserved to the extent possible (for diagnostic purposes) but may be inadvertently modified as part of the reset process.

When a Host Target FIFO transfer terminates due to a Device Time-out, DMA operation is disabled and a system interrupt is generated if the Channel Interrupt Enable set to one. The Target Time-out status bit for the associated channel must be reset to zero before issuing a new DMA instruction.

Ch1En (Ch2En)	Action
0	Channel is reset
1	Enable Channel Operation

Bit 30 **Instruction Fetch Enable, Ch1IfEn (Ch2IfEn).** The Ch1IfEn (Ch2IfEn) bit enables the DMA controller to fetch the next instruction from system memory at the address specified by Channel Command Pointer. When the current instruction is complete, the instruction fetch sequencer reads a new Channel Instruction and System Address from the address pointed to by the Channel Command Pointer if the Channel Instruction Fetch Enable is set to one.

Ch1IfEn (Ch2IfEn)	Action
0	Instruction Fetching Disabled
1	Instruction Fetching Enabled

Bit 29 **Interrupt on Completion, Ch1IC (Ch2IC).** The Interrupt on Completion allows a system interrupt to be generated when the associated DMA instruction has completed, if channel interrupts are enabled.

Ch1IC (Ch2IC)	Source
0	No Interrupt Enabled
1	Interrupt on Completion

Bits 28-26 **End Byte Address, Ch1EBA (Ch2EBA).** Transfers between System Memory and the DMA Channel FIFOs occur in multiples of 8 bytes (quad words). The End Byte Address is used to determine which bytes will be transferred from the last quad word as shown inTable 19.

Bits 25-23 **Start Byte Address, Ch1SBA, (Ch2SBA).** The Start Byte Address is used to determine which bytes will be transferred from the first quad word as shown inTable 20.

Byte resolution is only provided for the first and last quad word transfers. Between the first and last quad words all bytes are transferred.

Bits 22-16 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

Bits 15-14 **DMA Target Device Select, Ch1DvSl (Ch2DvSl).** This field selects the VIP target address. The DMA Device Select field is mapped to the DEVSEL field of the first VIP command/address byte. The predefined device types are shown in the next table

.

Ch1DvSl (Ch2DvSl)	Device Type
00	MPEG Decoder
01	Video Decoder
10	User-defined
11	User-defined

Bits 13-12 **DMA Target FIFO Select, Ch1FSl (Ch2FSl).** This field selects which of the four possible FIFOs is addressed by the transfer.

Ch1FSI (Ch2FSI)	FIFO Selected
00	FIFO A
01	FIFO B
10	FIFO C
11	FIFO D

Bits 11-9 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s.

Bits 8-0 **Transfer Size in Quad Words, Ch1XfrSz (Ch2XfrSz).** The number of quad word addresses accessed in System Memory is equal to ChXfrSz + 1. The actual number of bytes transferred depends on the DMA start and end byte address fields. Up to 4 Kbytes can be transferred with each DMA channel instruction.

The Transfer Size should be set to the number of quad word addresses to be accessed, not necessarily the number of bytes divided by eight.

If the transfer starts and ends on a quad word boundary, the Transfer Size is set to the byte count divided by four.

If the transfer starts on a quad word boundary but does not end on a quad word boundary OR ends on a quad word boundary but does not start on a quad word boundary, the Transfer Size should be set to the number of quad words to be transferred plus one for the partial transfer.

When the transfer does not start or end on a quad word boundary, the Transfer Size should be set to the number of quad words to be transferred plus two for the partial transfers.

Although partial transfers are supported they should be avoided if possible since they cause additional accesses to System Memory.

Table 19. DMA End Byte Address (Ch1EBA, Ch2EBA)

Ch1EBA (Ch2EBA)	BE[7]	BE[6]	BE[5]	BE[4]	BE[3]	BE[2]	BE[1]	BE[0]
000	0	0	0	0	0	0	0	1

Table 19. DMA End Byte Address (Ch1EBA, Ch2EBA)

001	0	0	0	0	0	0	1	1
010	0	0	0	0	0	1	1	1
011	0	0	0	0	1	1	1	1
100	0	0	0	1	1	1	1	1
101	0	0	1	1	1	1	1	1
110	0	10	1	1	1	1	1	1
111	1	1	1	1	1	1	1	1

Table 20. Byte Enable Generation for the First Quad Word

Ch1SBA (Ch2SBA)	BE[7]	BE[6]	BE[5]	BE[4]	BE[3]	BE[2]	BE[1]	BE[0]
000	1	1	1	1	1	1	1	1
001	1	1	1	1	1	1	1	0
010	1	1	1	1	1	1	0	0
011	1	1	1	1	1	0	0	0
100	1	1	1	1	0	0	0	0
101	1	1	1	0	0	0	0	0
110	1	0	0	0	0	0	0	0
111	1	0	0	0	0	0	0	0

.

10.5.5 DMA System Address Registers (Ch1SysAdr, Ch2SysAdr)

The System Address Registers (Ch1SysAdr and Ch2SysAdr) contain the quad word address of the next data to be transferred between the DMA Channel FIFOs and System Memory.

The System Address Register may be written directly by the driver or updated automatically by the DMA Instruction Fetch Sequencer.

Bits 31-3 **Quad Word Address.**

Bits 2-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s

10.6 VIP Video Input Port

The video input port registers are all initialized to 0 at power up. Writes to registers marked reserved are ignored, reads always return 0.

10.7 Raw VBI Data Capture

The most significant bit of the SAV and EAV codes is unused and set to one in the ITU-R 656 specification. The VIP 1.0 specification defines this bit as the Task bit, allowing two streams of data to be differentiated on the video port.The specification recommends using task B to transfer raw VBI data. The task bit can also be used to differentiate two video streams, much like the odd and even field information.

Raw VBI appears as extra lines in the video buffer, either proceeding or following the actual video. The video portion is normally byte swapped to match the format expected by the video display unit. The task bit provides a convenient mechanism to differentiate the two streams, byte swapping only the video portion.

When VBI data is being identified by the task bit, the byte swap enable should be set to one for that task to disable byte swapping.

10.7.1 Frame Buffer Address Readback (fb1_adr) Offset 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 21-3 **Frame Buffer Address.** Read-only static readback for frame buffer address register (for test).

Bits 2-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s

10.7.2 Video Input Port Configuration Register (vin_cfg) Offset 04h

The top byte (07h) of vin_cfg is reserved for enabling and disabling interrupts.

Bits 31-28 are used to reset interrupt enables. Individual interrupts are disabled by writing a ‘1’ to the associated Reset IRQ enable field. Writing a zero to the Reset IRQ enable field preserves the existing enable status. Read values for these fields are undefined and should be masked off before comparing.

Bits 27 to 24 are the interrupt enables. Individual interrupts are enabled by writing a ‘1’ to the associated interrupt enable field. Writing a zero preserves the existing value.

Writing a ‘1’ to both the enable and reset enable field at the same time produces undefined results.

Bit 31 **Reset Buffer Full IRQ Enable, Rst_BFIEn.**

(Write-only)

0	Preserve Buffer Full IRQ enable
1	Reset Buffer Full IRQ enable

Bit 30 **Reset Field IRQ Enable, Rst_FIEn.**

(Write-only)

0	Preserve Field IRQ enable
1	Reset Field IRQ enable

Bit 29 **Reset Vertical Blank IRQ Enable, Rst_VBIEn.**

(Write-only)

0	Preserve Vertical Blank IRQ enable
1	Reset Vertical Blank IRQ enable

Video Input Port Configuration Register (vin_cfg) (Continued)

Bit 28 **Reset Buffer Overflow IRQ Enable, Rst_BOEn.**

(Write-only)

0	Preserve Buffer Overflow IRQ enable
1	Reset Buffer Overflow IRQ enable

Bit 27 **Buffer Full IRQ Enable, BF_IEn.**)

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.**)

0	Preserve existing F_IEn value
1	IRQ is generated when the internal Field bit changes

Bit 25 **Vertical Blank IRQ Enable, VB_IEn.**)

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.**

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.**

Controls which video input buffer will be filled first

0	video input buffer 0 filled first
1	video input buffer 1 filled first

Bit 20 **Decimator Enable.**

0	no decimation of input pixels
1	Enable 2:1 video decimator

Bit 19 **Auto Update.** This bit enables automatic updating of the displayed video buffer.

0	Buffer must be updated by the driver.
1	Buffer automatically switched to the most recently completed display buffer.

Video Input Port Configuration Register (vin_cfg) (Continued)

Bits 18-16 **Video Input Format.** This field controls how the video stream will be decoded.

Bits 18-16	Video Input Format
000	VIP 1.0 Video Mode
001	*Reserved*
010	Pass through
011	STi3520A Compatibility mode (Multiplexed CCIR-601)
1xx	*Reserved*

In VIP 1.0 Video Mode the timing information is recovered from SAV and EAV codes embedded in the video stream. The video timing generator may be used to generate system timing

In multiplexed CCIR-601 mode, the video input port can generate video timing or lock to an external source. Modes 0 and 1 are STi3520A compatible. Video timing generation is enabled for formats 0-3 and disabled for modes 4-7.

Vin_cfg[14] and vin_cfg[5] contain the byte swap enables, extended for the Video Controller to include task based byte swap control. This allows raw VBI data to be placed in the frame buffer without byte swapping when it is identified by the task bit of the SAV code.

Task B Byte Swap controls byte swapping in VIP 1.0 mode when the video stream is identified as task 0. In other modes, this bit is ignored.

Task A Byte Swap controls byte swapping in compatibility mode and in VIP 1.0 mode when the video stream is identified as task A (ITU-R 656 compatible).

In passthrough mode and when the task bit is being used to identify raw VBI data, task bits should be set to one to disable the automatic byte swapping during video capture. When capturing video to be displayed, the task bit should be set to zero. Task B Byte Swap is only used when the Video Input Format is set to VIP 1.0 video mode.

Bit 15 *Reserved.* This Read-Only bit is reserved. When read it returns undefined data.

Bit 14 **Task B 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.

Bit 5 **Task A 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.

Video Input Port Configuration Register (vin_cfg) (Continued)

Bits 11-10 **Frame Drop Control.**

Bits 11-10	Frame Capture/Drop
00	Capture all frames
01	Capture first frame, drop one, repeat
10	Capture first frame, drop two, repeat
11	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.**.

Bits 9-7	Video Input Format
000	*Reserved*
001	Interlaced mode, capture only odd fields
010	Interlaced mode, capture only even fields
011	Interlaced mode, capture both even and odd fields
100	*Reserved*
101	Capture only Task A
110	Capture only Task B
111	Capture both Task A and Task B

Field Capture Control determines what fields are used to generate a frame. In progressive scan mode, fields are de-interlaced by merging odd and even fields into a single video buffer. This method of de-interlacing provides the highest vertical resolution but can cause motion artifacts where there are areas of movement. When capturing interlaced video in double buffer mode, the buffer is switched at the end of each enabled field.

The ability to capture video based on the Task bit in the SAV code is added to support VIP 1.0 operation. When video capture is based on the task bit and double buffering is enabled, the buffer is switched when the task changes.

Bit 6 **Double Buffer Enable.** Double Buffer Enable allows the amount of frame buffer memory to be reduced when capturing at less than full video rates. When single buffering is selected, all captured fields are written to the frame buffer using the buffer selected by the start buffer field.

0	single buffer
1	double buffer

Bit 4 **Enable Video Capture.**

Enable Video Capture starts or stops video capture operation. Video capture starts at the first enabled field of the next frame when video is being captured based on the field bit. When both fields are enabled, frame capture starts with field 1 (the odd field) as defined by ITU-R 656. When capture is based on the task bit and both tasks are enabled, video capture starts with Task A.

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 allows the port to be reset by software. This bit should not be asserted during normal operation as it unconditionally resets the port to the default values.

0	Video input port disabled, counters/state machines initialized, capture of video stopped.
1	Video input port enabled

10.7.3 Video Input Port Status Register (vin_stat) Offset 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.

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.

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.

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 buffer full flag is set, indicating that a buffer overrun has occurred. If the corresponding interrupt enable is asserted, an interrupt is generated. This bit is cleared by writing a value of ‘1’ to vin_stat[2].

Bit 1 **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].

10.7.4 Video Input Buffer Addr 0 (vin_ad0) Offset 0Ch

Lines of video always start at a quad word boundary in the frame buffer. When the display window size is not a multiple of 8, any remaining bytes in the last quad word will be unused (and undefined). The LS 3 bits of this register are hardwired to zero to force QWORD alignment.

Bits 31-22 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s

Bits 21-3 **Video Buffer Addr 0.** Quad word frame buffer start address for video input buffer 0.

Bits 2-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s

10.7.5 Video Input Buffer Addr 1 (vin_ad1) Offset 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 21-3 **Video Buffer Addr 1.** Quad word frame buffer start address for video input buffer 1.

Bits 2-0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s

10.7.6 Video Input Dest Pitch (vin_dp) Offset 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 13-3 **Destination Pitch.** This register holds the number of bytes in the frame buffer the beginning of one video scan line to the next.

Bits 2:0 *Reserved.* This Read-Only field is reserved. When read it returns ‘0’s

10.7.7 External Timing Generator 1 (vtg_ext1) Offset 28h

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..

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..

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..

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..

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..

Bits 23-21	Genlock Mode
000	No genlocking. Video timing generator resets horizontal and vertical counters based on H_Total and V_Total. (default)
001	Genlock to B/T# and HSYNC#
010	Genlock to SAV/EAV codes
001	*Reserved*
1xx	*Reserved*

External Timing Generator 1 (vtg_ext1) Offset 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 start 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.

10.7.8 External Timing Generator 2 (vtg_ext2) Offset 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.

10.7.9 Horizontal Timing Generator (vtg_ht) Offset 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. Since the horizontal counter runs at twice the pixel rate, they are multiplied by two before being compared. Video is captured when the horizontal counter is between the multiplied values, ie. (H_Start *2) < HCOUNT <= (H_End *2). Default values are specified for ITU-R 656 (525 line) timing.

Bit 21 *Reserved.* This Read-Only field is reserved. When read it returns undefined data.

Bits 20-11 **Horizontal end of active video in pixels, 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.

10.7.10 Video Timing Generator (vtg_vt) Offset 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.

10.8 Video Accelerator Registers

The video accelerator 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 accelerator registers are located at the 256-KByte wide sub-region 2. The figure below shows the address format for the video accelerator registers.

4Mbyte Region 1

256KbyteSub-region2

31 27	26 24	23 22	21 18	17 16
00001	<GBASE>	01	0010	01

15 14	13 2	1 0
<Cmd>	<Count>	00

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.

10.8.1 Source Specification registers

10.8.1.1 Video_Src_Base
RegOffset = 00h

This register specifies the DRAM linear starting address of the video 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 21-3 Base linear address of the Video Source Image

Bits 2-0 Zeros

This register is cleared to zero after reset

10.8.1.2 Video_Src_Pitch RegOffset = 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 Color 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 10-3 Specifies the amount to add to the current address to get to the address of the corresponding pixel in the next line.

Bits 2-0 Zeros

This register is cleared to zero after reset

10.8.1.3 Vid_Src_dim RegOffset = 08h

This register contains the dimensions of the Video Source Image relative the the starting corner.

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 Source image - 1, in lines, from the starting corner to the end of the image (dependent on Y_Vid_Src_dir)

Bits 15-9 *Reserved.*

Bits 9-0 **dX,** the width in pixels, of a line BEFORE it has been loaded into the video buffer, i.e. before decimation.

10.8.1.4 CRTC_Burst_length RegOffset = 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

10.8.1.5 Vid_Burst_Length RegOffset = 10h

This register is the video counterpart of the previous 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 window starts, the video low water mark is set to vid_lwm. After that time, for every vid_dt*8 pixels' time elapsed, the low water mark will be incremented by vid_dlwm bytes. As with crtc_dlwm, above, This field is a 2's complement number.

Setting this field to zero causes the video low water mark to remain at a constant value of vid_lwm.

Bits 23-16 **vid_lwm,** the (initial) low water mark for the video FIFO in bytes. During the video window, if the video FIFO occupancy rises above the low water mark (defined as a function of time by vid_dlwm and vid_dt) and the crtc occupancy rises above crtc_lwm then ownership of the system DRAM can be given back to the CPU.

The value of this register field can only be a multiple of eight (bits 18-16 are not writable and read as zeroes).

Bits 15-11 **vid_dt, 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 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

10.8.2 Destination Specification Registers

10.8.2.1 Vid_Dst_XY RegOffset = 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

10.8.2.2 Vid_Dst_dim RegOffset = 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

10.8.3 Filter Control Registers

10.8.3.1 Horiz_Start_Phase RegOffset = 1Ch

This register contains the horizontal start phase for interpolator 0 (Y/R) and interpolators 1 and 2 (UV/GB).

This register is double buffered. The active register is only updated during vertical sync.

Bits 31-14 Reserved.

Bits 13-0 **Horizontal Start Phase 0, h0sp[13:0].** Used in YUV modes to define the position of the Y component of the first pixel, or in RGB modes to define the position of the R, G and B components of the first pixel, relative to the source line stored in the video input buffer. In YUV modes, the Y input buffer contains 4 Y pixel components per location (32 bits). In RGB modes, the two buffers are treated as one 64-bit buffer, with each location containing four 16-bit pixels.

Hence, h0sp[13:12] defines a pixel or pixel component within the location, h0sp[11:9] defines the initial filter weighting and h0sp[8:0] provide additional accuracy for scale factor. The values of Y[0] or R[0], G[0] and B[0] are computed similarly to the computation of U and V in YUV 422 mode.

This register is cleared to zero after reset

10.8.3.2 Horiz_Scale RegOffset = 20h

This register contains the control for horizontal scaling and decimation.

This register is double buffered. The active register 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:-

```
hdf_tmp = (source_width +
           dest_width - 1)/dest_width
hpi = ((source_width/hdf_tmp) *
        4096)/dest_width
hdf = hdf_tmp - 1;
```

Note that the maximum value for hdf (programmed value) is 7 and for hpi is 4096.

This register is cleared to zero after reset

10.8.3.3 Vert_Start_Phase RegOffset = 24h

This register contains the vertical start phases for the vertical scaler.

This register is double buffered. The active register is only updated during vertical sync.

Bits 31-12 *Reserved.*

Bits 11-0 **Vertical Start Phase, vsp[11:0].** Defines the initial value of the vertical phase counter. The first output line of the video window will always be a horizontally scaled version of the line specified by Vid_Src_Base.

This register is cleared to zero after reset

10.8.3.4 Vert_Scale RegOffset = 28h

This register contains the control for vertical scaling and decimation.

This register is double buffered. The active register 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 replication 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

10.8.3.5 Color space converter specification RegOffset = 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

10.8.4 Video and Graphics mixing control registers

10.8.4.1 Mix Mode Register RegOffset = 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 displayed. Note that the video pixel can be compared either before or after the Color Space Converter. Note also that the chrom key can be programmed to ignore any or all component values.
1	1	*Reserved*

This register is cleared to zero after reset

10.8.4.2 Color Key Register RegOffset = 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

10.8.4.3 Chroma Key Low Register RegOffset = 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.**

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

10.8.4.4 Chroma Key High Register RegOffset = 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 operations.

Bits 15-8 **ch1high,** the high limit against which component 1 is compared during chroma key operations.

Bits 7-0 **ch0high,** the high limit against which component 0 is compared during chroma key operations.

This register is cleared to zero after reset

The operation of the chroma key can be summarized as follows:-

Let Cn represent component n, n = 0..2

Let Chnlow represent Chlow for component n, n = 0..2

Let Chnhigh represent Chigh for component n, n = 0..2

Kn be the result of the compare for component n, n = 0..2

```
for(n = 0; n < 3; n++)
if((Cn >= Chnlow) && (Cn <= chmhigh))
        Kn = 1
    else
        Kn = 0

If((I0||K0) && (I1||K1) && (I2||K2))
    display graphics pixel
else
    display video pixel
```

10.8.4.5 Status Register RegOffset = 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.

11 POWER MANAGEMENT

11.1 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 for the STPC.

The STPC provides the following hardware structures to assist the software in managing the power consumption by the system.

– System Activity detection

– Three power down timers:

– Doze timer for detecting lack system activity for short durations

– Standby timer for detecting lack of system activity for medium durations

– Suspend timer for detecting lack of system activity for long durations.

– house-keeping activity detection

– house-keeping timer to cope with short bursts of house-keeping activity while dozing or in standby state.

– Peripheral Activity detection

– Peripheral timer for detecting lack of peripheral activity

– STPCLK# modulation to adjust the system performance in various power down states of the system including full power on state

Lack of system activity for progressively longer period of times is detected by the three power down timers. These timers can generate SMI interrupts to CPU so that the SMM software can put the system in decreasing states of power consumption. System activity in a power down state can generate an SMI interrupt to allow the software to bring the system back up to the full power on state. The chip-set supports up to 3 power down states: Doze state, Standby state and Suspend state. These correspond to increasing levels of power savings.

The chip-set can detect presence/absence of the following System activities.

– DMA Request (DRQ) activity

– Interrupt Request (INTR) activity

– Parallel IO (PIO) activity

– Serial IO (SIO) activity

– Keyboard (KBD) activity

– Floppy Disk Controller (FDC) activity

– Hard Disk Controller (HDC) activity

– PCI master device activity

– A programmable address range

Each of these can be individually enabled. The presence of an enabled system activity resets the power down timers. The chip-set generates the SMI interrupt when no system activity is detected for the delay period programmed in the power-down timers. The software can then put appropriate sub-systems in power down mode, request STPCLK# assertion and stop CPU and other system clocks, program the current power-down state in the chip set and set up the next timer.

Presence of an enabled system activity, when the STPC is in a power down state will first enable any stopped clocks, wait for a a programmable delay to allow any internal PLLs to stablize and then de-assert STPCLK# to enable CPU execution. The device can optionally generate SMI interrupt to allow the SMM to bring the system back to power-on state.

The current revision of the STPC does not implement support for stopping CPU and other system clocks.

In Doze or Standby state, a house-keeping activity can bring the system back to full speed for a short period of time before returning back to Doze or Standby state. The chip-set can detect following house-keeping activities.

– DMA Request (DRQ) activity
– Interrupt Request (INTR) activity
– Keyboard (KBD) activity
– PCI master device activity

The house-keeping timer determines the length of time the system will be on before returning to the original power-down state. An activity can be either a system activity or a house-keeping activity but not both at the same time. Further, the Suspend state can not make use of this feature.

The absence of the following peripheral activities can be enabled to cause a SMI interrupt and thus allowing the software to put the unused peripherals in power down state while the remainder of the system is still in full power on state.

– Parallel IO (PIO) activity
– Serial IO (SIO) activity
– Keyboard (KBD) activity
– Floppy Disk Controller (FDC) activity
– Hard Disk Controller (HDC) activity
– A programmable address range

Each of these can be individually enabled for inactivity detection. The presence of a peripheral activity does not reset the peripheral timer. It always times out after the programmed delay period. An SMI interrupt is generated if any of the enabled peripheral was not active for this time period. The device provides IO access trapping to detect access to a powered-down peripheral so that the software can bring the peripheral to power on state before the access is completed.

The STPC can also do software transparent power management if so enabled. In this mode of operation, doze and standby time-outs will change the CPU clock without generating an SMI interrupt. The state transitions from fully-on to doze or standby and back to fully-on will take place automatically. Also note that the suspend state can never be entered automatically and always requires software assist.

The STPC decodes the following to detect activities of various kind:

Activity	Detected via
ISA DMA masters	Low to high transition of hold request of 206
PCI masters	High to low transition of any of PCIRQ2-0#
Parallel port	IO read/write at 378h-37Fh, 278h-27Fh and 3BCh-3BFh
Serial port	IO read/write at 3F8h-3FFh, 2F8h-2FFh, 3E8h-3EFh and 2E8h-2EFh
Keyboard	IO read/write at 60h, 62h, 64h and 66h
Floppy disk	IO read/write at 3F2h, 3F4h, 3F5h and 3F7h
Hard disk	IO read/write in 170h-177h, 376h,1F0h-1F7h and 3F6h address range as well as any bus master activity by the internal IDE controller.

11.2 POWER MANAGEMENT REGISTERS

11.2.1 Timer Register 0 Index 60h

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 up to the programmed valued. If any of the enabled system activities are detected before time out, the timer will reset and start again. These bits are encoded as follows:

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-power saving (software transparent power management) mode, change the power-down state to Standby state (refer to auto-power saving mode for details of the power saving features are enabled with standby state). Similar to the Suspend timer, presence of an enabled system activity will reset the timer to start counting again. These bits are encoded as follows:

4	3	2	Standby Timer reset
0	0	0	disabled
0	0	1	1 minutes
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.*

11.2.2 Timer Register 1 Index 61h

Bit 7 *Reserved.*

Bits 6-4 **House-keeping Timer.** This timer determines how long the PMU will be in Doze housekeeping state when an enabled house-keeping activity is detected while in doze or standby power-down states. It is encoded as follows:

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	1 milli-seconds
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:

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.

11.2.3 Timer Register 2 Index 8Dh

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-power saving (software transparent power management) mode, change the power-down state to Doze state (refer to auto-power saving mode for details of the power saving features that are enabled with Doze state). Similar to the suspend timer, presence of an enabled system activity will reset the timer to start counting again. This 3-bit field is encoded as follows:

7	6	5	Doze Timer reset
0	0	0	disabled
0	0	1	50 milli-seconds
0	1	0	100 milli-seconds
0	1	1	500 milli-seconds
1	0	0	1 second
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.

11.2.4 System Activity Enable Register 0 Index 62h

This is the first of the three registers that control which system activity to detect. 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.

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.

11.2.5 System Activity Enable Register 1 Index 63h

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 corresponds to ignoring all of the listed system activities.

11.2.6 System Activity Enable Register 2 Index 64h

This is the third of the three registers that control which system activity to detect.

Bit 7 **IRQ15-1** 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.

11.2.7 House-keeping Activity Enable Reg. 0 Index 65h

This register controls which house-keeping activity to detect. House-keeping activities are detected only in Doze and Standby states. If enabled, a house-keeping activity reverts the system back to power-on state for a short period of time programmed in the house-keeping timer. Set the following bits to a ‘1’ to enable activity detection and a ‘0’ to ignore the associated activity.

Bit 7 **DMA Request (DRQ)** activity

Bit 6 **PCI master device** activity

Bit 5 **Keyboards (KBD)** activity

Bit 4 **IRQ15-1** activity

Bit 3 **IRQ0** activity

Bit 2 **NMI** activity

Bits 1-0 *Reserved.*

This register defaults to 00h disabling detection of all house-keeping activity detection.

11.2.8 House-keeping Activity Enable Reg. 1 Index 66h

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.

11.2.9 Peripheral Inactivity detection Reg. 0 Index 67h

This register controls which peripheral inactivity is enabled for generating a SMI interrupt on a peripheral time-out. 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.

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.

11.2.10 Peripheral Inactivity detection Reg.1 Index 68

This is the second peripheral inactivity detection register.

Bits 7-0 *Reserved.*

This register is not implemented in this revision of the STPC.

11.2.11 Peripheral Activity detection Reg. 0 Index 69

This register controls which peripheral accesses will cause a SMI. Typically the power management software will detect non-usage of a peripheral device via Peripheral inactivity status registers, bring the peripheral into power down state and then enable trapping access to that peripheral via this register.

Thus when an application attempts to make use of a powered down peripheral, the access is trapped and a SMI interrupt is generated to allow software to re-power the peripheral device before allowing the access to complete. This is register is first of the two such registers.

A ‘1’ in a bit position enables SMI generation for the associate peripheral and a ‘0’ disables.

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.

11.2.12 Peripheral Activity detection Reg. 1 Index 6Ah

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.

11.2.13 Address range 0 register 0 Index 6Bh

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.

11.2.14 Address range 0 register 1 Index 6Ch

Bits 7-3 These bits are compared with PCI address bits 23-19 if range compare is enabled for memory cycle or compared against bits 7-3 if range compare is enabled for IO cycles.

Bit 2 This bit is compared with PCI address bit 18 if range compare is enabled for memory cycle or compared with address bit 2 if range compare is enabled for IO cycles and range is 4-bytes. Otherwise this bit when 1 specifies that the range of IO address to be compared is 16-bytes and when 0, the range is 8-bytes.

Bit 1 This bit is compared with PCI address bit 17 if range compare is enabled for memory cycle. Otherwise if range compare is enabled for IO cycles, this bit if 1 specifies that the range of IO address to be compared is 8/16-bytes and when 0 the range is 4-bytes.

Bit 0 This bit when ‘1’ specifies that range compare should be done for memory cycles and when ‘0’, for IO cycles.

This register defaults to 00h after reset.

11.2.15 Address range 1 register 0 Index 6Dh

11.2.16 Address range 1 register 1 Index 6Eh

11.2.17 Address range 2 register 0 Index 6Fh

11.2.18 Address range 2 register 1 Index 70h

These registers are not implemented in this revision.

11.2.19 SMI Control register 0 Index 71h

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 housekeeping 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 housekeeping 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.

11.2.20 SMI Control register 1 Index 72h

This register is not implemented in this revision.

11.2.21 SMI status register 0 Index 73h

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 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#.

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.

11.2.22 SMI Status register 1 Index 74h

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.

11.2.23 Peripheral Inactivity status register 0 Index 75h

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. A bit in this register is set to a ‘1’ only at peripheral timer time-out. It is set to a ‘0’, as soon as an activity from the associated peripheral is detected.

The status reflected in this register is not conditioned by whether the peripheral was enabled for inactivity detection through the Peripheral Inactivity Detection registers or not. The SMI interrupt however will be generated only if any of the enabled peripherals (via Peripheral Inactivity Enable register) were inactive for the entire duration of the peripheral time out.

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’.

11.2.24 Peripheral Inactivity status register 1 Index 76h

This is the second peripheral inactivity status register and is similar to peripheral inactivity status register 0.

This register is not implemented in this revision.

11.2.25 Activity status register 0 Index 77

This register records presence of activity. 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.

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.

11.2.26 Activity Status register 1 Index 78h

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.

11.2.27 Activity Status register 2 Index 79h

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.

11.2.28 PMU state register Index 7Ah

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:**

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

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 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.

11.2.29 General Purpose Register Index 7Bh

This is a read/write IO register that can be used by software. Writing to this register also updates the external ‘373 latch that can be used to control external devices for power-down purposes. Reads of this register return the value of this internal register.

The GPIOCS# signal will be asserted when writing to this register to latch the data on the ISA data bus.

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.

11.2.30 Clock Control register 0 Index 7Ch

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:

7	6	5		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 quarterperiod
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:

7	6	5		Doze 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	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 64µs.

Bit 0 *Reserved.*

This register defaults to 00h.

11.2.31 Clock Control register 1 Index 7Dh

11.2.32 Wake-up Control register Index 7Eh

11.2.33 Leakage Control register Index 7Fh

11.2.34 Power Control Register 0 Index 80h

11.2.35 Power Control Register 1 Index 81h

11.2.36 Power Control Register 2 Index 82h

11.2.37 Heat regulation control register 0 Index 83h

11.2.38 Heat regulation control register 1 Index 84h

11.2.39 LED control register Index 85h

11.2.40 External input definition register 0 Index 86h

11.2.41 External input definition register 1 Index 87h

These registers are not implemented in the current revision.

11.2.42 Doze timer read back register Index 88h

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. This register should not be used by the software.

Bits 7-0 Bits 8-1 of the current value of the doze timer.

Note that bit 0 of the current value of the doze timer is not readable.

11.2.43 Standby timer read back register Index 89h

This read only register is provided for test purposes to read back the current value of 5-bit standby timer. This register should not be used by software.

Bits 7-5 *Reserved.*

Bits 4-0 Bits 4-0 of the current value of the standby timer.

11.2.44 Suspend timer read back register Index 8Ah

This read only register is provided for test purposes to read back the current value of the 7-bit Suspend timer. This register should not be used by software.

Bit 7 *Reserved.*

Bits 6-0 Bits 6-0 of the current value of the suspend timer.

11.2.45 House-keeping timer read back Reg. Index 8Bh

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. This register should not be used by software.

Bits 7-0 Bits 8-1 of the house-keeping timer.

11.2.46 Peripheral timer read back register Index 8Ch

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. This register should not be used by software.

Bits 7-0 Bits 8-1 of the Peripheral timer.

12 ELECTRICAL SPECIFICATIONS

12.1 Introduction

The electrical specifications in this chapter are valid for the STPC Client.

12.2 Electrical Connections

12.2.1 Power/Ground Connections/Decoupling

Due to the high frequency of operation of the STPC Client, it is necessary to install and test this device using standard high frequency techniques. The high clock frequencies used in the STPC Client and its output buffer circuits can cause transient power surges when several output buffers switch output levels simultaneously. These effects can be minimized by filtering the DC power leads with low-inductance decoupling capacitors, using low impedance wiring, and by utilizing all of the VSS and VDD pins.

12.2.2 Unused Input Pins

All inputs not used by the designer and not listed in the table of pin connections in Chapter 3 should be connected either to VDD or to VSS. Connect active-high inputs to VDD through a 20 kΩW (±10%) pull-down resistor and active-low inputs to VSS and connect active-low inputs to VCC through a 20 kΩW (±10%) pull-up resistor to prevent spurious operation.

12.2.3 Reserved Designated Pins

Pins designated reserved should be left disconnected. Connecting a reserved pin to a pull-up resistor, pull-down resistor, or an active signal could cause unexpected results and possible circuit malfunctions.

12.3 Absolute Maximum Ratings

The following table lists the absolute maximum ratings for the STPC Client device. Stresses beyond those listed under Table 21 limits may cause permanent damage to the device. These are stress ratings only and do not imply that operation under any conditions other than those specified in section ”Operating Conditions”.

Exposure to conditions beyond Table 21 may (1) reduce device reliability and (2) result in premature failure even when there is no immediately apparent sign of failure. Prolonged exposure to conditions at or near the absolute maximum ratings (Table 21) may also result in reduced useful life and reliability.

Table 21. Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{DDx}	DC Supply Voltage	-0.3, 4.0	V
V_I, V_O	Digital Input and Output Voltage	-0.3, VDD + 0.3	V
T_{STG}	Storage Temperature	-40, +150	°C
T_{OPER}	Operating Temperature	0, +70	°C
P_{TOT}	Total Power Dissipation	4.8	W

12.4 DC Characteristics

Table 22. DC Characteristics

Recommended Operating conditions : VDD = 3.3V±0.3V, Tcase = 0 to 100°C unless otherwise specified

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
V_{DD}	Operating Voltage		3.0	3.3	3.6	V
P_{DD}	Supply Power	V_{DD} = 3.3V, H_{CLK} = 66Mhz		3.2	3.9	W
H_{CLK}	Internal Clock	(Note 1)			75	Mhz
V_{REF}	DAC Voltage Reference		1.215	1.235	1.255	V
V_{OL}	Output Low Voltage	I_{Load} =1.5 to 8mA depending of the pin			0.5	V
V_{OH}	Output High Voltage	I_{Load} =-0.5 to -8mA depending of the pin	2.4			V
V_{IL}	Input Low Voltage	Except XTALI	-0.3		0.8	V
		XTALI	-0.3		0.9	V
V_{IH}	Input High Voltage	Except XTALI	2.1		V_{DD}+0.3	V
		XTALI	2.35		V_{DD}+0.3	V
I_{LK}	Input Leakage Current	Input, I/O	-5		5	µA
C_{IN}	Input Capacitance	(Note 2)				pF
C_{OUT}	Output Capacitance	(Note 2)				pF
C_{CLK}	Clock Capacitance	(Note 2)				pF

Notes:

1. MHz ratings refer to CPU clock frequency.
2. Not 100% tested.

12.5 AC Characteristics

Table 24 through Table 29 list the AC characteristics including output delays, input setup requirements, input hold requirements and output float delays. These measurements are based on the measurement points identified in Figure 11 and Figure 12. The rising clock edge reference level VREF , and other reference levels are shown in Table 23 below for the STPC Client. Input or output signals must cross these levels during testing.

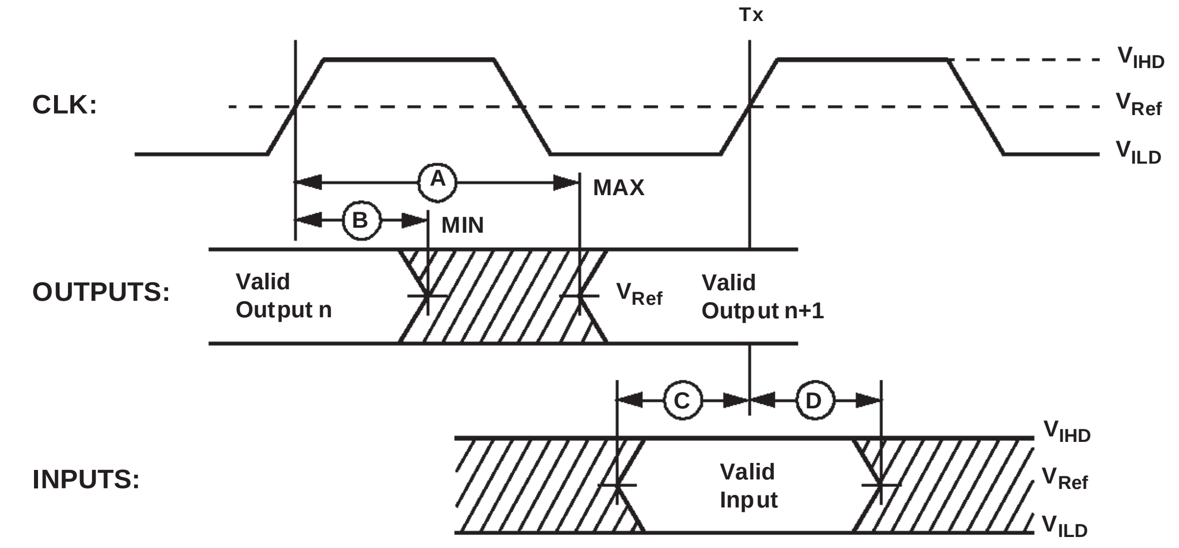
Figure 11 shows output delay (A and B) and input setup and hold times (C and D). Input setup and hold times (C and D) are specified minimums, defining the smallest acceptable sampling window a synchronous input signal must be stable for correct operation.

Table 23. Drive Level and Measurement Points for Switching Characteristics

Symbol	Value	Units
V_{REF}	1.5	V
V_{IHD}	3.0	V
V_{ILD}	0.0	V

Note: Refer to Figure 11.

Figure 11. Drive Level and Measurement Points for Switching Characteristics



LEGEND:
A - Maximum Output Delay Specification
B - Minimum Output Delay Specification
C - Minimum Input Setup Specification
D - Minimum Input Hold Specification

Figure 12. CLK Timing Measurement Points

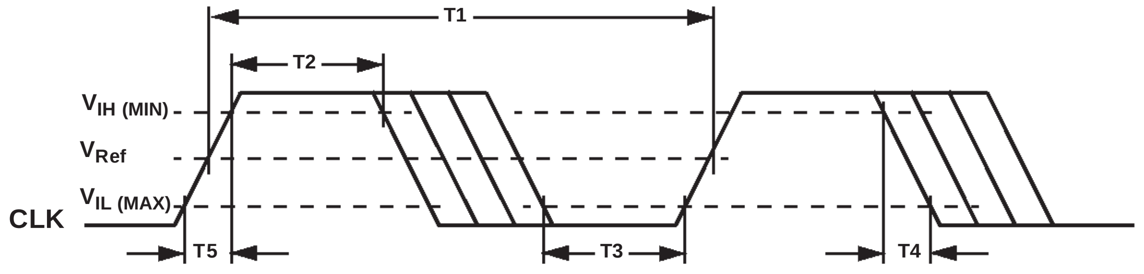


Table 24. PCI Bus AC Timing

Name	Parameter	Min	Max	Unit
t1	PCI_CLKI to AD[31:0] valid	2	11	ns
t2	PCI_CLKI to FRAME# valid	2	11	ns
t3	PCI_CLKI to CBE#[3:0] valid	2	11	ns
t4	PCI_CLKI to PAR valid	2	11	ns
t5	PCI_CLKI to TRDY# valid	2	11	ns
T6	PCI_CLKI to IRDY# valid	2	11	ns
T7	PCI_CLKI to STOP# valid	2	11	ns
T8	PCI_CLKI to DEVSEL# valid	2	11	ns
T9	PCI_CLKI to PCI_GNT# valid	2	12	ns
t10	AD[31:0] bus setup to PCI_CLKI	7		ns
t11	AD[31:0] bus hold from PCI_CLKI	0		ns
t12	PCI_REQ#[2:0] setup to PCI_CLKI	10		ns
t13	PCI_REQ#[2:0] hold from PCI_CLKI	0		ns
t14	CBE#[3:0] setup to PCI_CLKI	7		ns
t15	CBE#[3:0] hold to PCI_CLKI	0		ns
t16	IRDY# setup to PCI_CLKI	7		ns
t17	IRDY# hold to PCI_CLKI	0		ns
t18	FRAME# setup to PCI_CLKI	7		ns
t19	FRAME# hold from PCI_CLKI	0		ns

Table 25. IDE Bus AC Timing

Name	Parameter	Min	Max	Unit
t20	DD[15:0] setup to PIOR#/SIOR# falling	15		ns
t21	DD[15:0} hold to PIOR#/SIOR# falling	12		ns

Table 26. DRAM Bus AC Timing

Name	Parameter	Min	Max	Unit
t22	HCLK to RAS#[3:0] valid		15	ns
t23	HCLK to CAS#[7:0] bus valid		15	ns
t24	HCLK to MA[11:0] bus valid		15	ns
t25	HCLK to MWE# valid		15	ns
t26	HCLK to MD[63:0] bus valid		19	ns
t27	MD[63:0] Generic setup			ns
t28	GCLK2X to RAS#[3:0] valid		15	ns
t29	GCLK2X to CAS#[7:0] valid		15	ns
t30	GCLK2X to MA[11:0] bus valid		15	ns
t31	GCLK2X to MWE# valid		15	ns
t32	GCLK2X to MD[63:0] bus valid		18	ns
t33	MD[63:0] Generic hold			ns

Table 27. Video Input/TV Output AC Timing

Name	Parameter	Min	Max	Unit
t34	DCLK to TV_YUV[7:0] bus valid		18	ns
t35	VIDEO_D[7:0] setup to VCLK	5		ns
t36	VIDEO_D[7:0] hold from VCLK	2		ns
t37	VCLK to VTV_BT# valid		15	ns
t38	VCLK to VTV_HSYNC valid		15	ns
t39	VTV_BT# setup to VCLK	10		ns
t40	VTV_BT# hold from VCLK	5		ns
t41	VTV_HSYNC setup to VCLK	10		ns
t42	VTV_HSYNC hold from VCLK	5		ns

Table 28. Graphics Adapter (VGA) AC Timing

Name	Parameter	Min	Max	Unit
t43	DCLK to VSYNC valid		45	ns
t44	DCLK to HSYNC valid		45	ns

Table 29. ISA Bus AC Timing

Name	Parameter	Min	Max	Unit
t45	XTALO to LA[23:17] bus active		60	ns
t46	XTALO to SA[19:0] bus active		60	ns
t47	XTALO to BHE# valid		62	ns
t48	XTALO to SD[15:0] bus active		35	ns
t49	PCI_CLKI to ISAOE# valid		28	ns
t50	XTALO to GPIOCS# valid		60	ns
t51	XTALO to ALE valid		62	ns
t52	XTALO to MEMW# valid		50	ns
t53	XTALO to MEMR# valid		50	ns
t54	XTALO to SMEMW# valid		50	ns
t55	XTALO to SMEMR# valid		50	ns
t56	XTALO to IOR# valid		50	ns
t57	XTALO to IOW# valid		50	ns

13 MECHANICAL DATA

13.1 388-Pin Package

The pin numbering for the STPC 388-pin Plastic BGA package is shown in Figure 13.

Dimensions are shown in Figure 14, Table 30 and Figure 15, Table 31.

Figure 13. 388-Pin PBGA Package - Top View

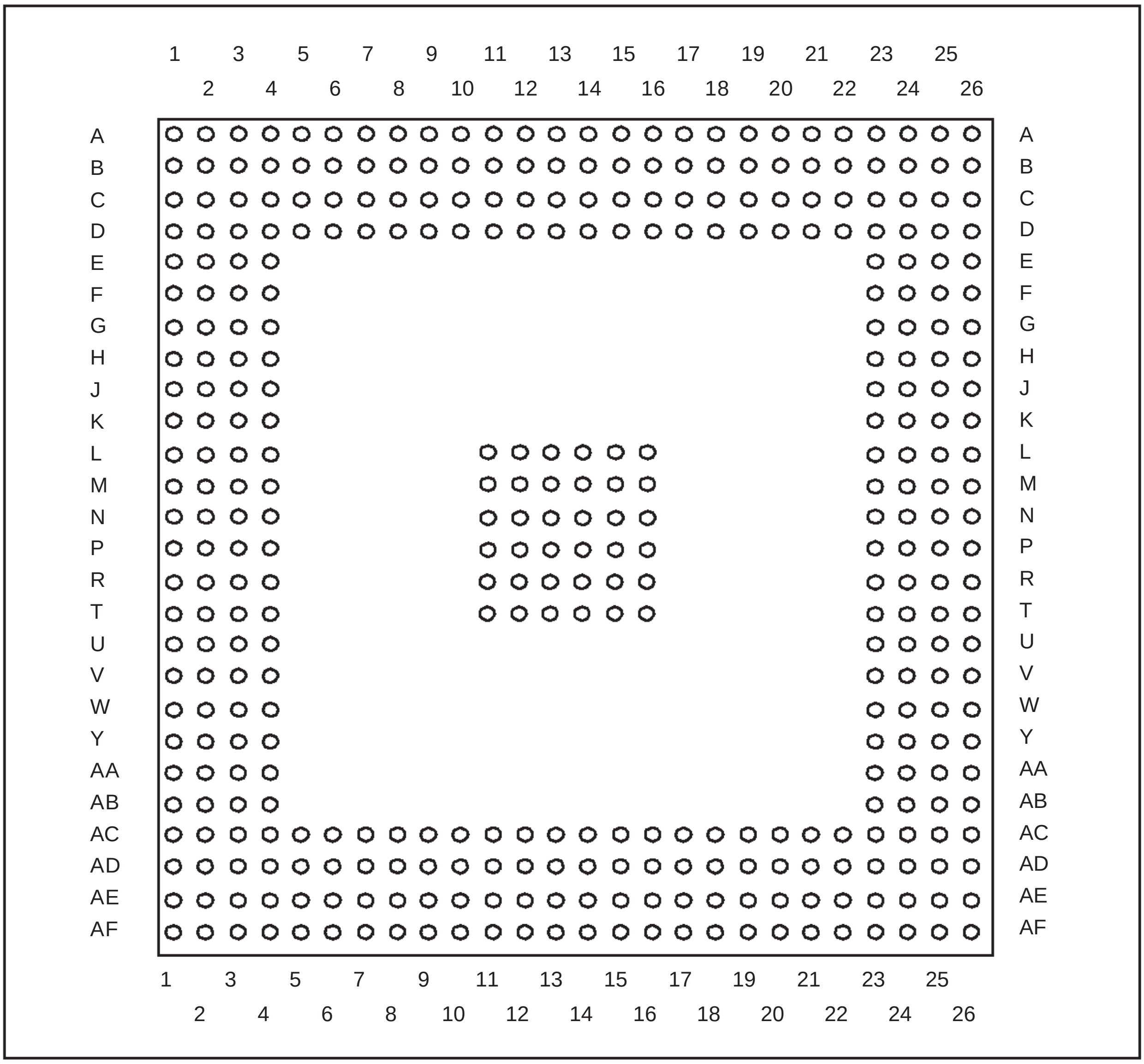


Figure 14. 388-Pin PBGA Package - Dimensions

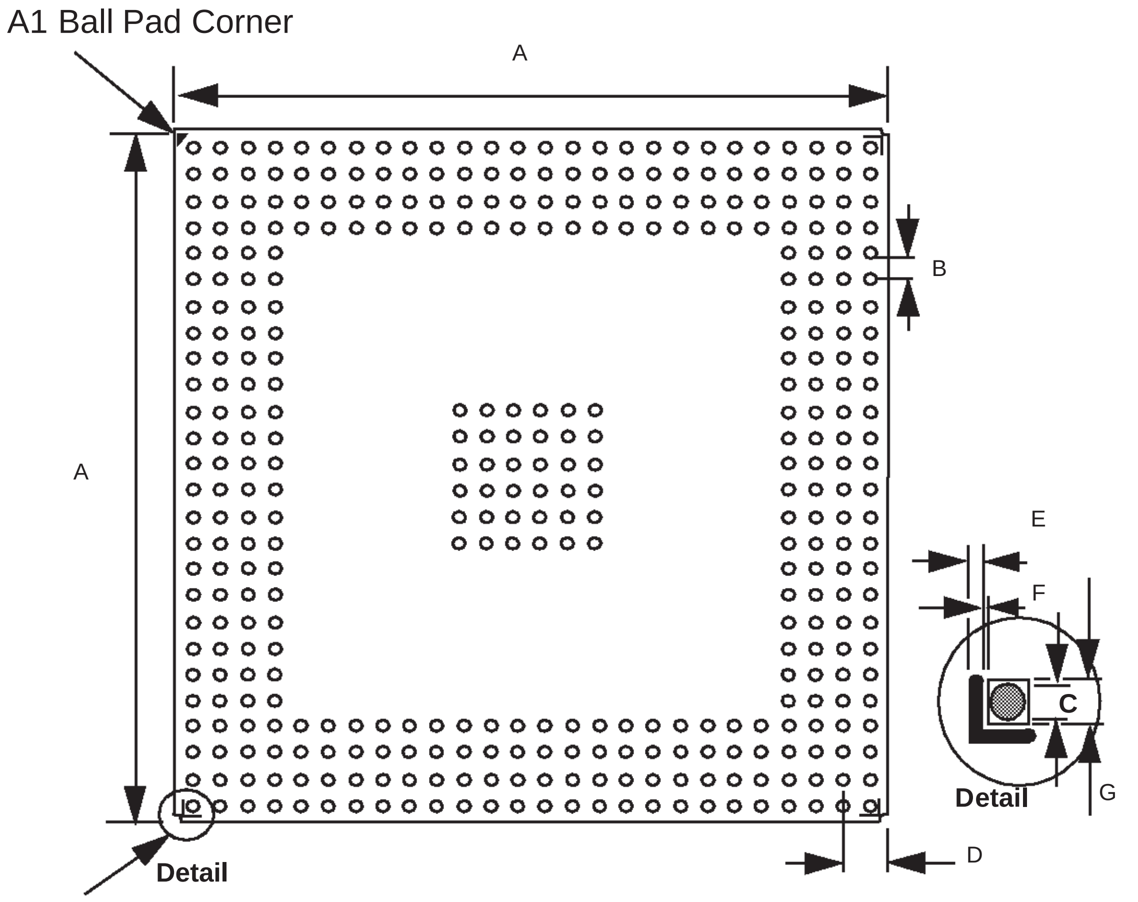


Table 30. PBGA388 - 388 Solder Ball Plastic 35mm x 35mm

Symbols	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	34.95	35.00	35.05	1.375	1.378	1.380
B	1.22	1.27	1.32	0.048	0.050	0.052
C	0.58	0.63	0.68	0.023	0.025	0.027
D	1.57	1.62	1.67	0.062	0.064	0.066
E	0.15	0.20	0.25	0.006	0.008	0.001
F	0.05	0.10	0.15	0.002	0.004	0.006
G	0.75	0.80	0.85	0.030	0.032	0.034

Figure 15. 388-Pin PBGA Package - Dimensions (Continued)

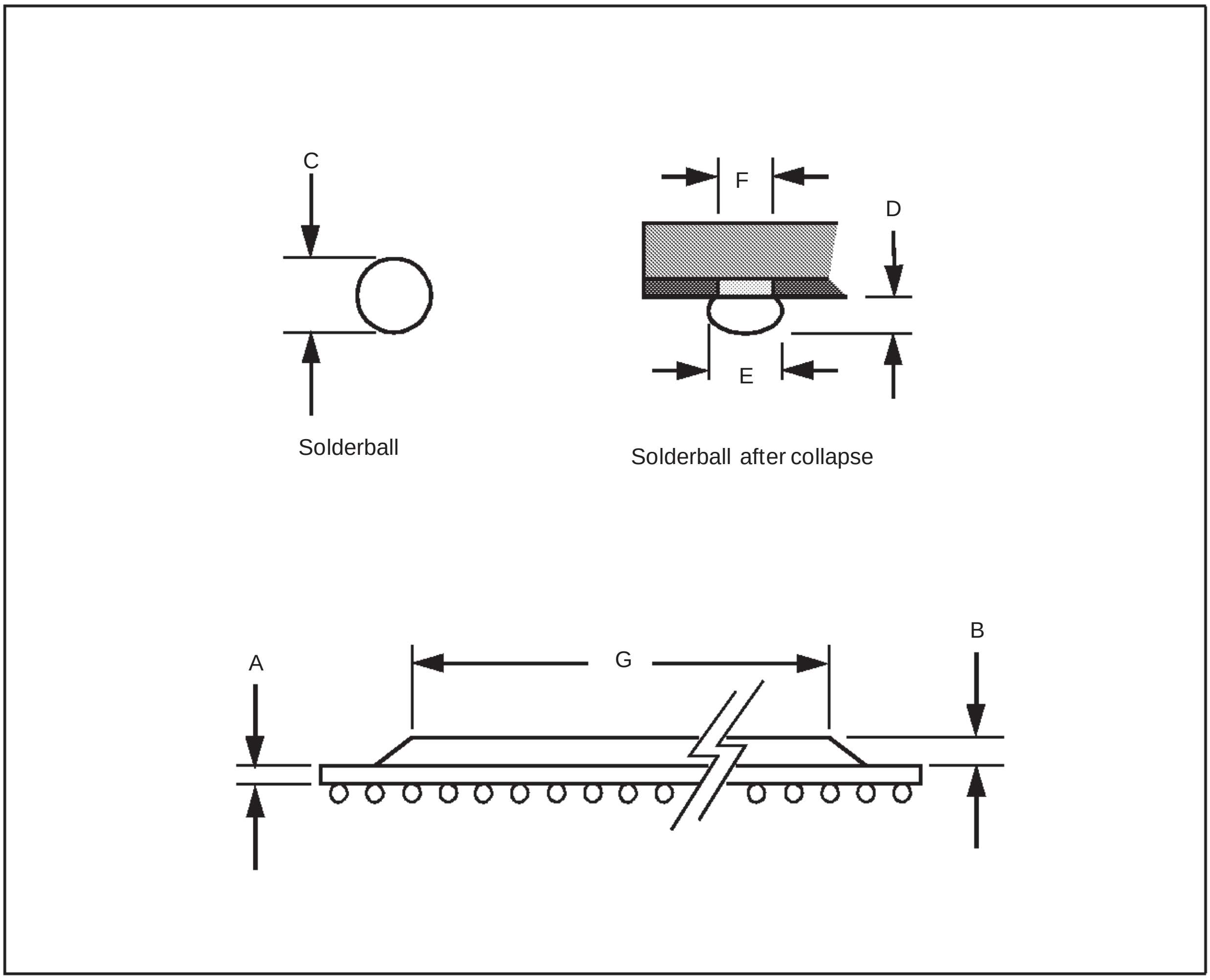
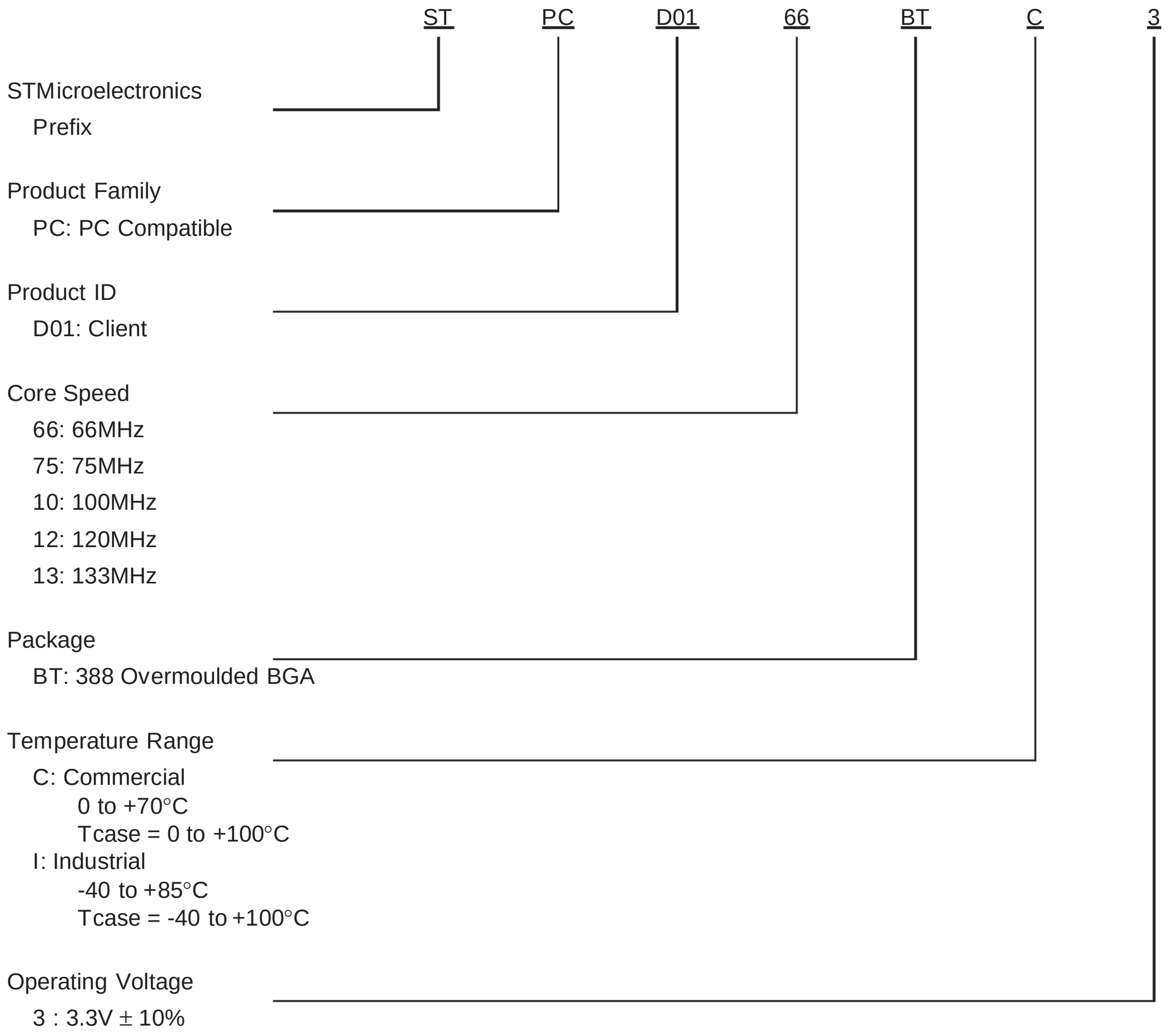


Table 31. PBGA388 - 388 Solder Ball Plastic 35mm x 35mm (Continued)

Symbols	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	0.50	0.56	0.62	0.020	0.022	0.024
B	1.12	1.17	1.22	0.044	0.046	0.048
C	0.60	0.76	0.92	0.024	0.030	0.036
D	0.52	0.53	0.54	0.020	0.021	0.022
E	0.63	0.78	0.93	0.025	0.031	0.037
F	0.60	0.63	0.66	0.024	0.025	0.026
G		30.0			11.8	

14 ORDERING DATA

14.1 Ordering Codes



ST PC D01 66 BT C 3

STMicroelectronics
Prefix

Product Family
PC: PC Compatible

Product ID
D01: Client

Core Speed
66: 66MHz
75: 75MHz
10: 100MHz
12: 120MHz
13: 133MHz

Package
BT: 388 Overmoulded BGA

Temperature Range
C: Commercial
0 to +70°C
Tcase = 0 to +100°C
I: Industrial
-40 to +85°C
Tcase = -40 to +100°C

Operating Voltage
3 : 3.3V ± 10%

14.2 Available Part Numbers

Part Number	Core Frequency (MHz)	Temperature Range (C)	Operating Voltage (V)
STPCD0166BTC3	66	0 to + 70°	3.3V ± 10%
STPCD0175BTC3	75		
STPCD0110BTC3	100		
STPCD0112BTC3	120		
STPCD0113BTC3	133		

14.3 Customer Service

More informations are available on STMicroelectronics internet site *http://www.st.com/stpc*.

For technical support, a mail-box is in place at *stpc.support@st.com*.

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - France - Germany - Italy - Japan - Korea - Malaysia - Malta - Mexico - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.