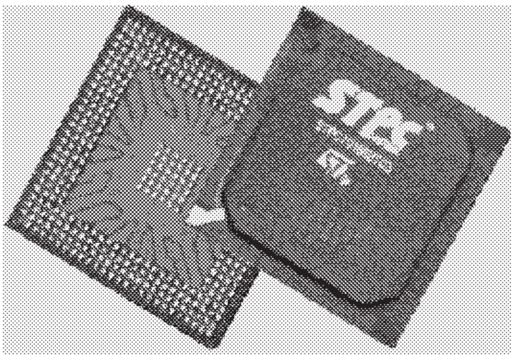
STPC INDUSTRIAL

PC Compatible Embedded Microprocessor

- POWERFUL X86 PROCESSOR
- 64-BIT BUS ARCHITECTURE
- 64-BIT 66MHz DRAM CONTROLLER
- SVGA GRAPHICS CONTROLLER
- 135MHz RAMDAC
- UMA ARCHITECTURE
- TFT DISPLAY CONTROLLER
- PCI MASTER / SLAVE / ARBITER
- LOCAL BUS INTERFACE
- ISA (MASTER/SLAVE) INTERFACE
  -INCLUDING THE IPC
- PC-CARD INTERFACE
  - PCMCIA
  - CARDBUS
- I/O FEATURES
  - PC/AT+ KEYBOARD CONTROLLER
  - PS/2 MOUSE CONTROLLER
  - 2 SERIAL PORTS
  - 1 PARALLEL PORT
- IPC
  - DMA CONTROLLER
  - INTERRUPT CONTROLLER
  - TIMER / COUNTERS
- POWER MANAGEMENT

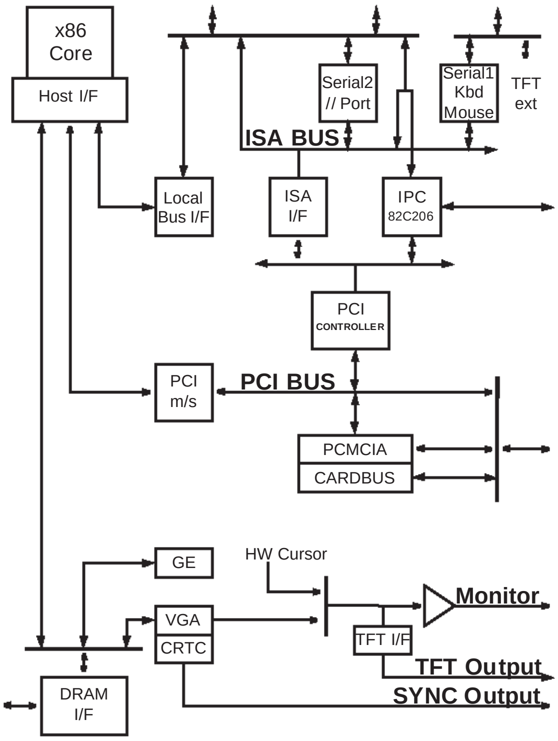
STPC INDUSTRIAL OVERVIEW

The STPC Industrial integrates a fully static x86 processor, fully compatible with standard fifth generation x86 processors, and combines it with powerful chipset, graphics, TFT, PC-Card, Local Bus, keyboard, mouse, serials and parallel interfaces to provide a single Industrial oriented PC compatible subsystem on a single device. The performance of the device is comparable with the performance of a typical P5 generation system.
The device is packaged in a 388 Plastic Ball Grid Array (PBGA).



PBGA388

Figure 1. Logic Diagram



Issue 1.1

- **X86 Processor core**
- Fully static 32-bit 5-stage pipeline, x86 processor fully PC compatible.
- Access up to 4GB of external memory.
- 8Kbyte unified instruction and data cache with write back capability.
- Parallel processing integral floating point unit, with automatic power down.
- Clock core speeds up to 100 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-MByte system memory in 4 banks and down to as little as 2Mbytes.
- Supports 4-MByte, 8-MByte, 16-MByte, and 32-MByte single-sided and double-sided DRAM SIMMs.
- Four quad-word write buffers for CPU to DRAM and PCI to DRAM cycles.
- Four quad-word read prefetch 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.
- Complete backward compatibility to VGA and SVGA standards.
- Hardware acceleration for text (generalized bit map expansion), bitblts, transparent blts and fills.
- Up to 64 x 64 bit graphics hardware cursor.
- Up to 4MB long linear frame buffer.
- 8, 16, 24 and 32 bit pixels.
- Drivers for Windows and other operating systems.

- **CRT Controller**
- Integrated 135MHz triple RAMDAC allowing for 1280 x 1024 x 75Hz display.
- Requires external frequency synthesizer and reference sources.
- 8, 16, 24 and 32-bit pixels.
- Interlaced or non-interlaced output.

- **TFT Interface**
- Programmable panel size up to 1024 by 1024 pixels.
- Support for 640 x 480, 800 x 600 & 1024 x 768 active matrix TFT flat panels with 9, 12, 18-bit interface.
- Support 1 & 2 Pixels per Clock.
- Programmable image positionning.
- Programmable blank space insertion in text mode.
- Programmable horizontal and vertical image expansion in graphic mode.
- A fully programmable PWM (Pulse Width Modulator) signals to adjust the flat panel brightness and contrast.
- Supports **PanelLink**^{TM} high speed serial transmitter externally for high resolution panel interface.

- **PCI Controller**
- Fully compliant with PCI Version 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.

- **Local Bus interface**
- 66MHz, low latency bus.
- Asynchronous / synchronous.
- 22-bit address and 16-bit data busses.
- 2 Programmable Flash EPROM Chip Select.
- 4 Programmable I/O Chip Select.
- Separate memory and I/O address spaces.
- Memory prefetch (improved performances).

- **ISA master/slave**
- Generation of the ISA clock from either 14.318MHz oscillator clock or system clock
- 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.

- **PC-Card interface**
- Support one PCMCIA 2.0 / JEIDA 4.1 68-pin standard PC Card Socket.
- Power Management support.
- Support PCMCIA/ATA specifications.
- Support I/O PC Card with pulse-mode interrupts.
- Provides an ExCA^{TM} implementation to PCMCIA 2.0 / JEIDA 4.1 standards.
- DMA support.

- **Keyboard interface**
- Fully PC/AT& compatible

- **Mouse interface**
- Fully PS/2 compatible

- **Serial interface**
- 15540 compatible
- Programmable word length, stop bits, parity.
- 16-bit programmable baud rate generator.
- Interrupt generator.
- Loop-back mode.
- 8-bit scratch register.
- Two 16-bit FIFOs.
- Two DMA handshake lines.

- **Parallel port**
- Standard Centronics mode supported.
- Nibble mode supported.

- **Integrated Peripheral Controller**
- Two 8237/AT compatible 7-channel DMA controllers.
- Two 8259/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.
- Supports IO trap & restart.
- Independent peripheral time-out timer to monitor hard disk, serial & parallel ports.
- Supports APM
- Supports RTC, interrupt and DMA wake ups

ExCA is a trademark of PCMCIA / JEIDA.

PanelLink is a trademark of SiliconImage, Inc

UPDATE HISTORY FOR STPC INDUSTRIAL OVERVIEW

The following changes have been made to the STPC Indusrial Overview on 27/10/99.

Section	Change	Text
	Replaced	“~~Support for VGA and SVGA active matrix TFT flat panels with 9, 12, 18-bit interface (1 pixel per clock).~~ ~~Support for XGA and SXGA active matrix TFT flat panels with 2 x 9-bit interface (2 pixels per clock)~~.” with “Support for 640 x 480, 800 x 600 & 1024 x 768 active matrix TFT flat panels with 9, 12, 18-bit interface. Support 1 & 2 Pixels per Clock.”

The following changes have been made to the STPC Indusrial Overview on 15/10/99.

Section	Change	Text
	Replaced	“~~Two fully programmable PWM (Pulse Width Modulator) signals to adjust the flat panel brightness and contrast.~~” with “A fully programmable PWM (Pulse Width Modulator) signals to adjust the flat panel brightness and contrast.”

Update History for STPC industrial overview

1 TABLE OF CONTENTS

2LIST OF TABLES

3 LIST OF FIGURES

4 GENERAL DESCRIPTION

At the heart of the STPC Industrial is an advanced 64-bit processor block, dubbed the 5ST86. The 5ST86 includes a powerful x86 processor core along with a 64-bit DRAM controller, advanced 64-bit 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).

The STPC Industrial has in addition to the 5ST86 a TFT output, a Local Bus interface, PC Card and super I/O features.

The STPC Industrial 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 means a reduction in total system memory for system performances that are equal to that of a comparable frame buffer and system memory based system, and generally much better, due to the higher memory bandwidth allowed 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 resolution screens and greater color depth. The processor bus runs at 66Mhz further increasing “standard” bandwidth by at least a factor of two.

The ‘standard’ PC chipset functions (DMA, interrupt controller, timers, power management logic) are integrated together with the x86 processor core; additional functions such as communication ports are accessed by the STPC Industrial via an internal ISA bus.

The PCI bus is the main data communication link to the STPC Industrial chip. The STPC Industrial translates appropriate host bus I/O and Memory cycles onto the PCI bus. It also supports the generation of Configuration cycles on the PCI bus. The STPC Industrial, 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.

Graphics functions are controlled through the on-chip SVGA controller and the monitor display is produced through the 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. The results of these operations change the contents of the on-screen or off-screen frame buffer areas of DRAM memory. The frame buffer can occupy a space up to 4 Mbytes anywhere in the physical main memory.

The maximum graphics resolution supported is 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.

To generate the TFT output, the STPC Industrial extracts the digital video stream before the RAMDAC and reformats it to the TFT format. The height and width of the flat panel are programmable through configuration registers up to a size of 1024 by 1024.

By default, lower resolution images cover only a part of the larger TFT panel. The STPC Industrial allows to expand the image vertically and horizontally in text mode by inserting programmable blank pixels. It allows expantion of the image vertically and horizontally in graphics mode by replicating pixels. The replication of J times every K pixel is independently programmable in the vertical and horizontal directions.

PanelLink^{TM} is a proprietary interconnect protocol defined by Silicon Image, Inc. It consists of a transmitter that takes parallel video/graphics data from the host LCD graphics controller and transmits it serially at high speed to the receiver which controls the TFT panel. The TFT interface is designed to support the connection of this control signal to the **PanelLink**^{TM} transmitter.

The STPC Industrial CARDBUS / PCMCIA controller has been specifically designed to provide the interface with PC-Cards which contain additional memory or I/O and provides an **ExCA**^{TM} implementation to PCMCIA 2.0 / JEIDA 4.1 standards.

The power management control facilities include socket power control, insertion/removal capability, power saving with Windows inactivity, NCS controlled Chip Power Down, together with further controls for 3.3v suspend with Modem Ring Resume Detection.

The need for system configuration jumpers is eliminated by providing address mapping support for PCMCIA 2.0 / JEIDA 4.1 PC-Card memory together with address windowing support for I/O space.

Selectable interrupt steering from PC-Card to internal system bus is also provided.

The STPC Industrial implements a multi-function parallel port. The standard PC/AT compatible logical address assignments for LPT1, LPT2 and LPT3 are supported.

The parallel port can be configured for any of the following 3 modes and supports the IEEE Standard 1284 parallel interface protocol standards as follow:

-Compatibility Mode (Forward channel, standard)

-Nibble Mode (Reverse channel, PC compatible)

-Byte Mode (Reverse channel, PS/2 compatible)

The STPC Industrial BGA package has 388 balls, but this is not sufficient for all the integrated functions, therefore some features are sharing the same balls and can not be used at the same time. The STPC Industrial configuration is done by ‘strap options’. It is a set of pull-up or pull-down resistors on the memory data bus, checked on reset, which auto-configure the STPC Industrial.

We can distinguish three main blocks ***independently configurables*** : The ISA / Local Bus block, the Serial 1 / TFT block, and the PCI / PC Card block.

From the first block, we can activate either the ISA bus and some IPC additionnal features, or the Local bus, the parallel port and the second serial interface.

From the second block, we can activate either the first serial port, or the TFT extension to get from 4 bit per colour to 6 bit per colour.

From the third block, we can activate either the PCI bus, or the PC Card interface (CardBus/ PCMCIA/ZoomVideo).

The STPC Industrial 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 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 periods of time 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 described above, these correspond to decreasing levels of power savings.

Power down puts the STPC Industrial 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. Because of the static nature of the core, no internal data is lost..

Figure 4.1. Functionnal description.

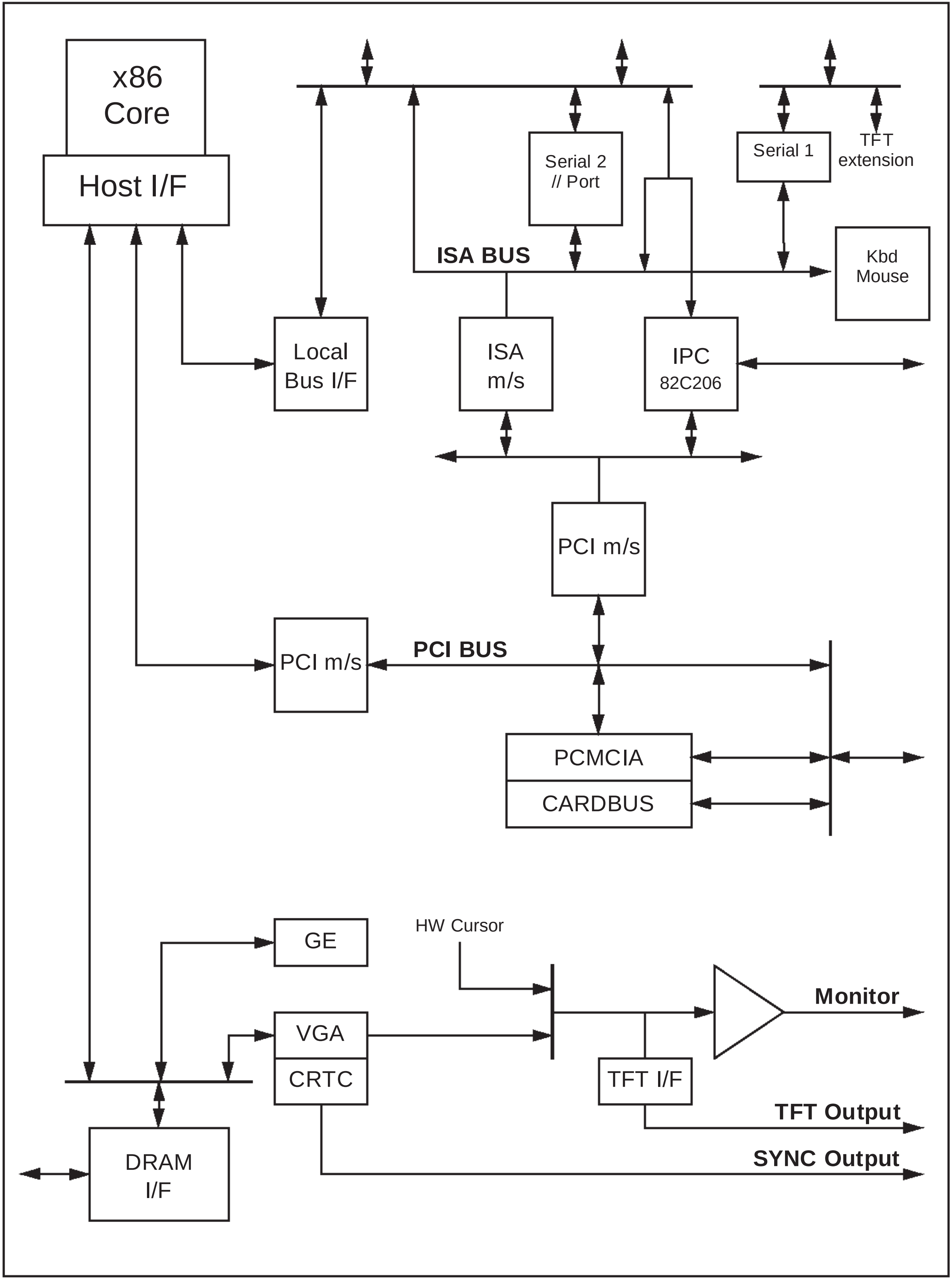


Figure 4.2. PCI, PCMCIA & CARDBUS modes:

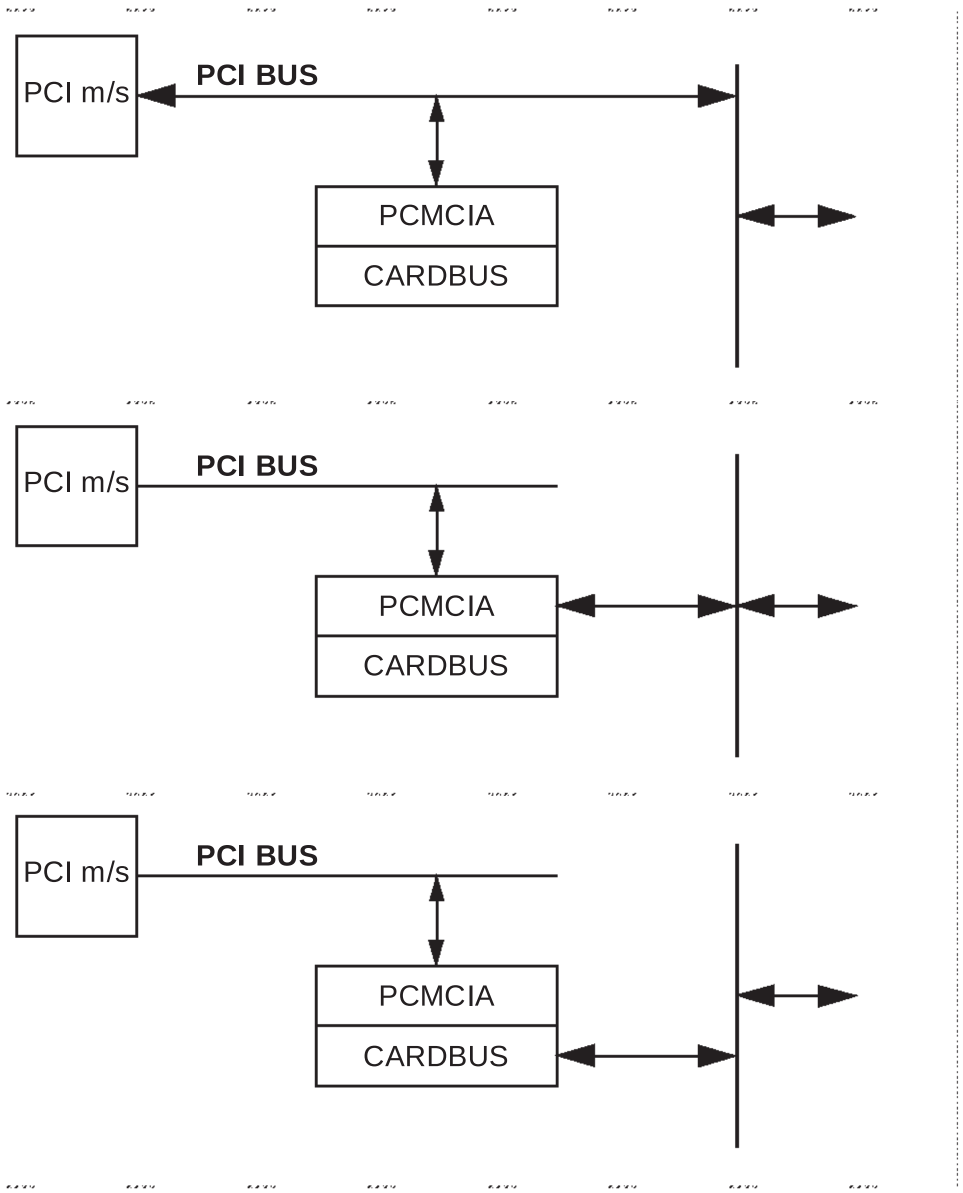


Figure 4.3. Local Bus and ISA bus modes:

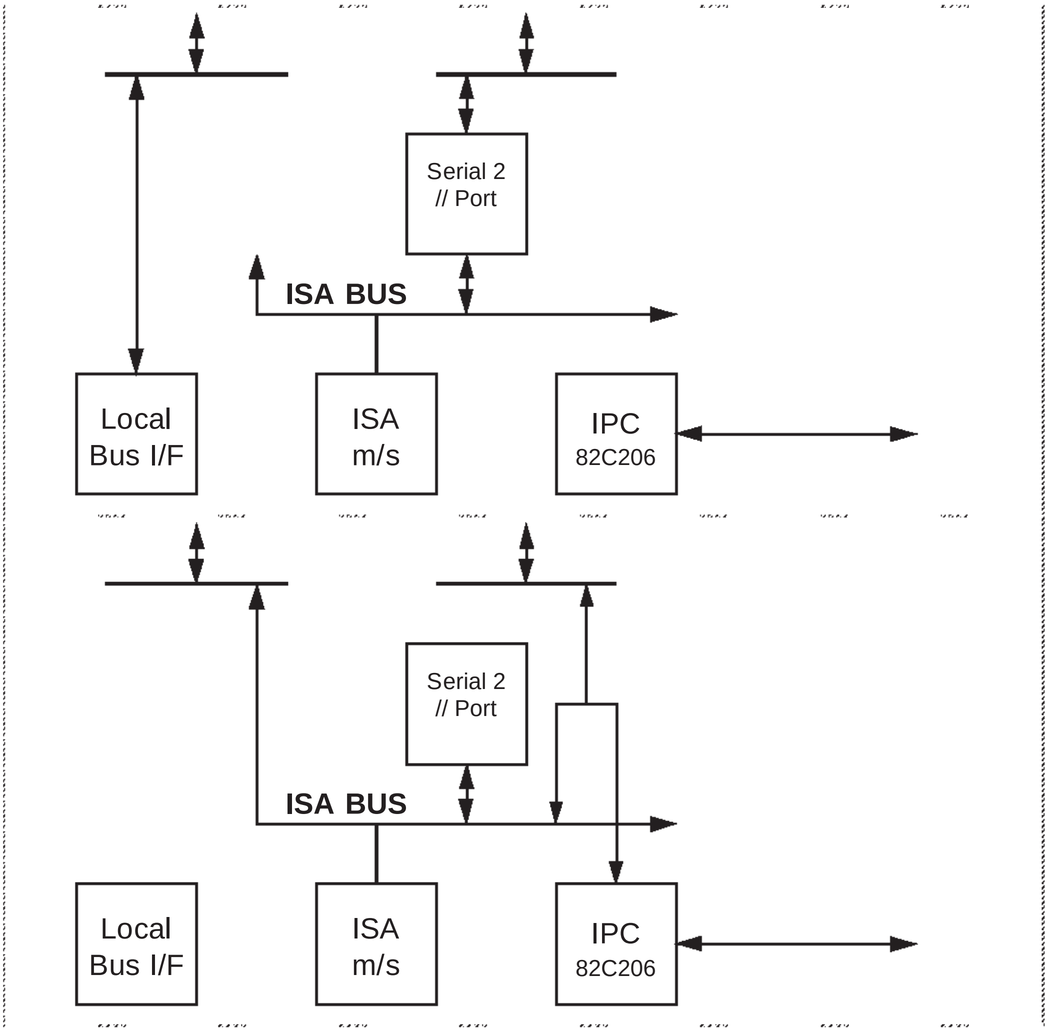


Figure 4.4. TFT in normal (serial 1 available) and extended modes (serial 1 unavailable).

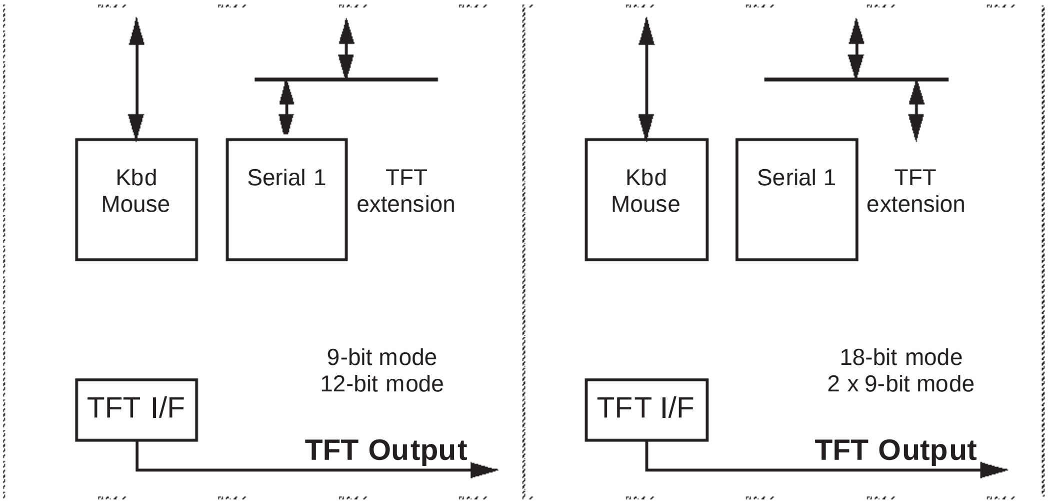


Figure 2. Typical PC oriented Application

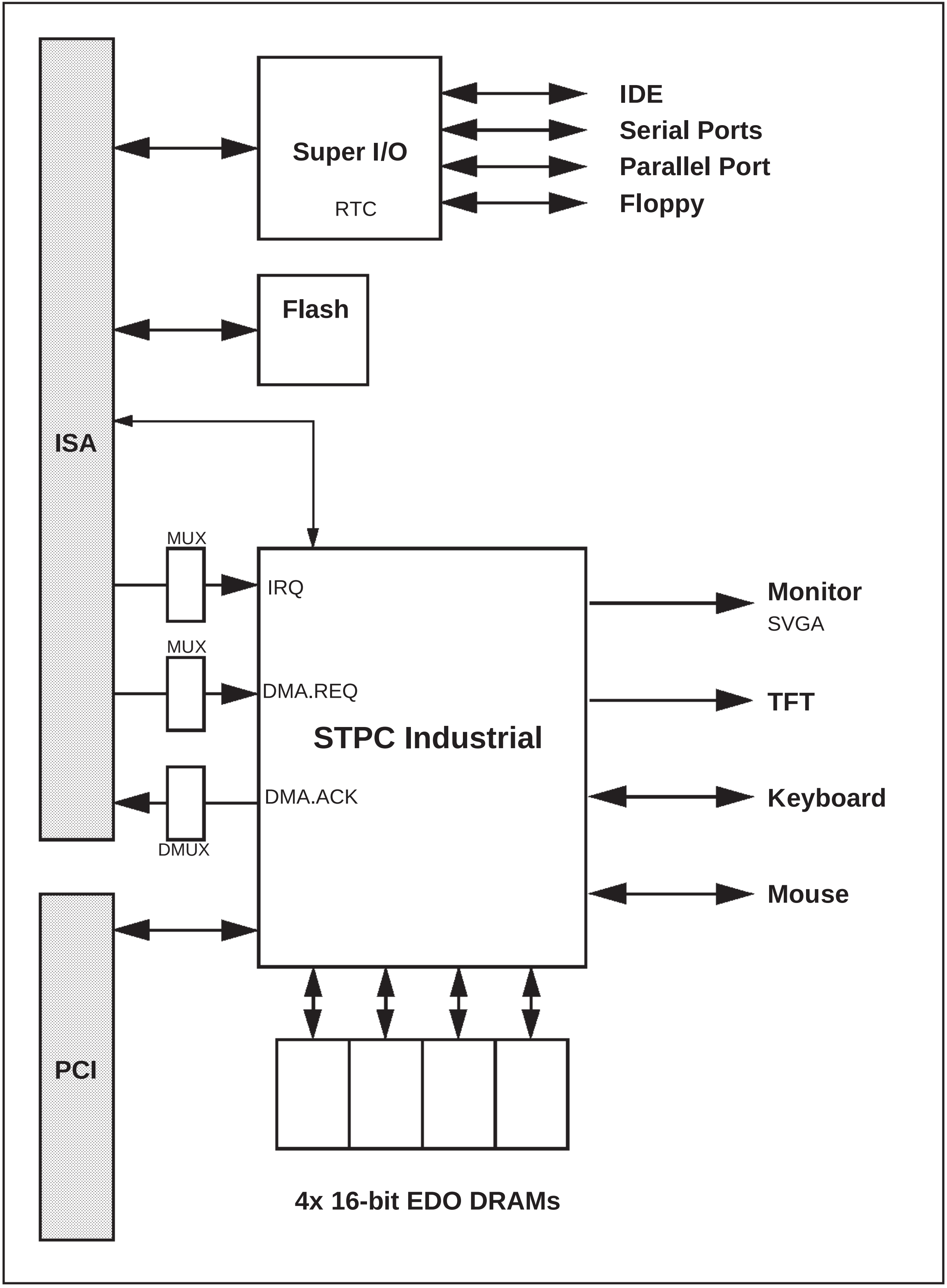
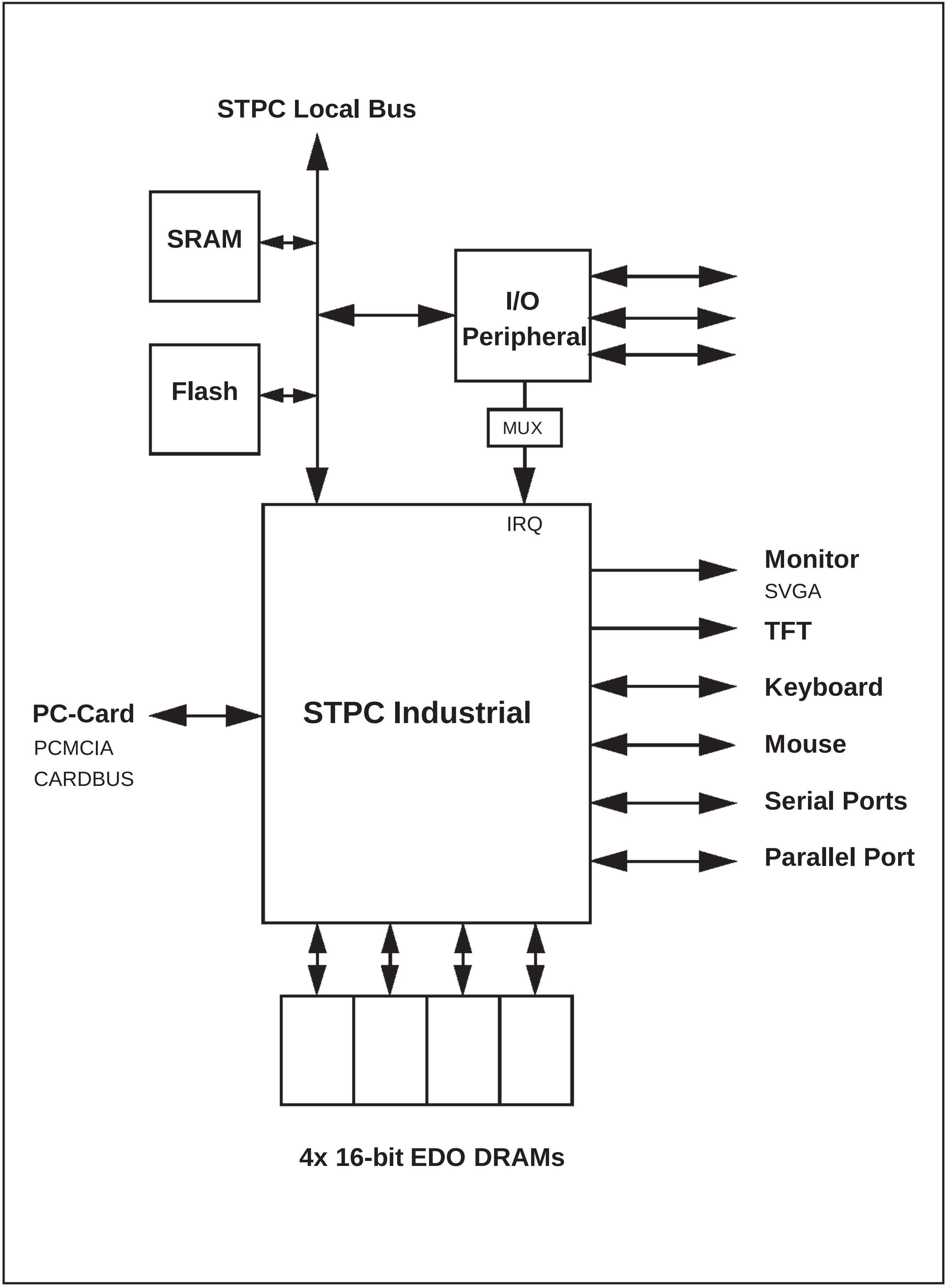


Figure 3. Typical Embedded Application



5. HOW TO USE THIS MANUAL

5.1 INTRODUCTION

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

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

5.2 SPECIFIC NOTES

5.2.1 RESERVED BITS

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

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

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

5.2.2 SIGNAL ACTIVE STATE

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

5.2.3 HEXADECIMAL NOTATION

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

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

5.2.4 ENDIAN

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

For an 8-bit value

N	7	6	5	4	3	2	1	0

For a 16-bit (word) value

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

For a 24-bit value

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

For a 32-bit (long word) value

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

For a 64-bit (QUAD word) value

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

5.3 ISSUING NOTES

There are three levels identified; Advanced data, Preliminary data and Full production release. Each level is identified in a specific way as follows.

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

6 PIN DESCRIPTION

6.1. INTRODUCTION

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

Figure 6-1 shows the STPC Industrial's external interfaces. It defines the main busses and their function. Table 6-1 describes the physical implementation listing signal types and their functionalities. Table 6-2 provides a full pin listing and description.

Table 6-4 provides a full listing of the STPC Industrial package pin location physical connection. Please refer to the pin allocation drawing for reference.

Due to the number of pins available for the package, and the number of functional I/Os, some pins have several functions, selectable by strap option on Reset. Table 6-3 provides a summary of these pins and their functions.

Table 6-1. Signal Description

Group name		Qty
Basic Clocks, Reset & Xtal (SYS)		13
DRAM Controller(DRAM)		89
PCI Controller	55	64
PC Card Interface	64	
Keyboard/Mouse Controller (SIO)		4
Local Bus I/F, Parallel I/F, Serial 2	75	75
ISA Interface/IPC extensions	73	
Serial 1 (SIO)	8	26
TFT output	24	
VGA Controller (VGA)		10
Grounds		74
V_{DD}		16
Analog specific V_{CC}/V_{DD}		16
Reserved		1
Total Pin Count		388

Figure 6-1. STPC Industrial External Interfaces

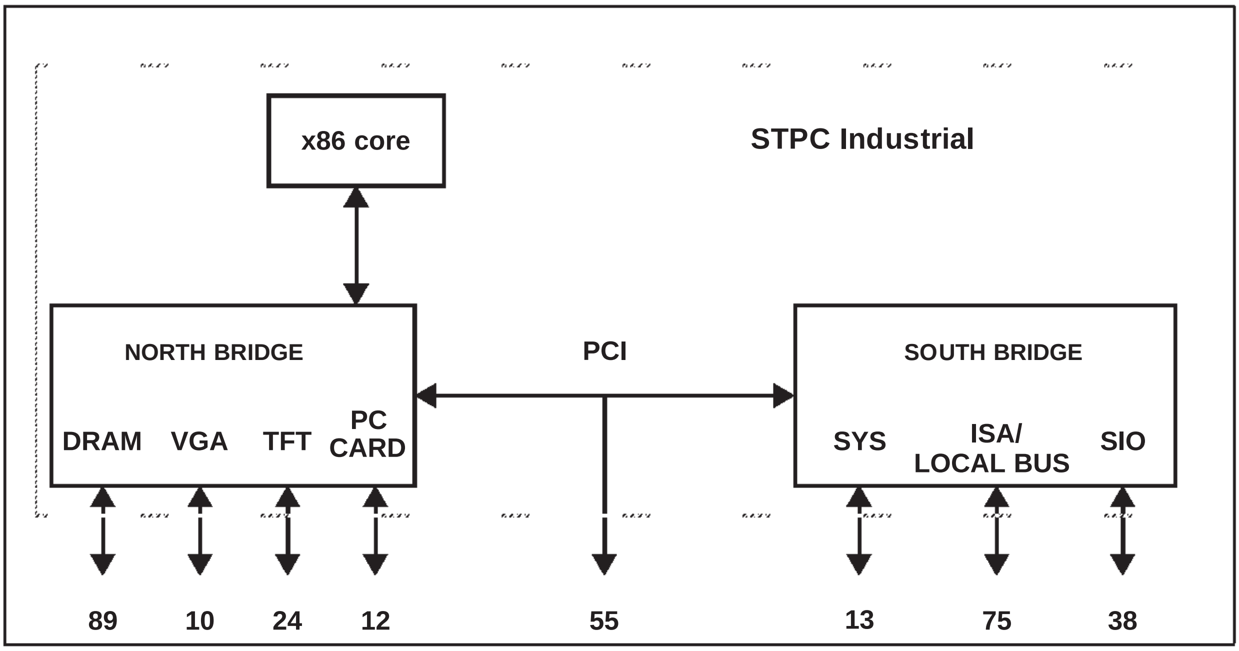


Table 6-2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
BASIC CLOCKS AND RESETS			
SYSRSTI#*	I	System Reset / Power good	1
SYSRSTO#*	O	Reset Output to System	1
XTALI	I	14.3MHz Crystal Input	1
XTALO	O	14.3MHz Crystal Output	1
PCI_CLKI*	I	33MHz PCI/CardBus Input Clock	1
PCI_CLKO	O	33MHz PCI/CardBus Output Clock	1
ISA_CLK, ISA_CLK2X	O	ISA Clock x1 and x2 (also Multiplexer Select Line For IPC)	2
CLK14M	O	ISA bus synchronisation clock	1
HCLK*	I/O	33 / 66MHz Host Clock (Test)	1
DEV_CLK*	O	24MHz Peripheral Clock	1
GCLK2X	I/O	80MHz Graphics Clock	1
DCLK	I/O	135MHz Dot Clock	1
V_{DD}_xxx_PLL		Power Supply for PLL Clocks	
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
LOCAL BUS INTERFACE (COMBINED WITH ISA BUS)			
PA[21:0]*	O	Address Bus [21:0]	22
PD[15:0]*	I/O	Data Bus [15:0]	16
PRDY#*	I	Ready	1
PWR#[1:0]*	O	Memory and I/O Write signals	2
PRD#[1:0]*	O	Memory and I/O Read signals	2
FCS#[1:0]*, IOCS#[3:0]*	O	Flash Memory and I/O Chip Select	6
ISA BUS INTERFACE (COMBINED WITH LOCAL BUS, PARALLEL PORT, SERIAL INTERFACE)			
LA[23:17]*	O	Unlatched Address	7
SA[19:0]*	O	Latched Address	20
SD[15:0]*	I/O	Data Bus	16
IOCHRDY*	I	I/O Channel Ready	1
ALE*	O	Address Latch Enable	1
BHE#*	O	System Bus High Enable	1
MEMR#*, MEMW#*	I/O	Memory Read & Write	2
SMEMR#*, SMEMW#*	O	System Memory Read and Write	2
IOR#*, IOW#*	I/O	I/O Read and Write	2
MASTER#*	I	Add On Card Owns Bus	1
MCS16#*, IOCS16#*	I	Memory Chip Select 16, I/O Chip Select 16	2
REF#*	I	Refresh Cycle	1
AEN*	O	Address Enable	1
IOCHCK#*	I	I/O Channel Check (ISA)	1
RTCRW#*	O	RTC Read / Write#	1
RTCDS#*	O	RTC Data Strobe	1
RTCAS#*	O	RTC Address Strobe	1

Table 6-2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
RMRTCCS#*	O	ROM / RTC Chip Select	1
GPIOCS#*	I/O	General Purpose Chip Select	1
IRQ_MUX[3:0]*	I	Multiplexed Interrupt Request	4
DACK_ENC[2:0]*	O	DMA Acknowledge	3
DREQ_MUX[1:0]*	I	Multiplexed DMA Request	2
TC*	O	ISA Terminal Count	1
KEYBOARD & MOUSE INTERFACE			
KBDATA*, MDATA*	I	Keyboard & Mouse Data Line	2
KBCLK*, MCLK*	O	Keyboard & Mouse Clock Line	2
SERIAL INTERFACE (SERIAL 1 COMBINED WITH TFT INTERFACE / SERIAL 2 COMBINED WITH IPC)			
SIN1*, SIN2*	I	Serial Data In (Serial 1, 2)	2
SOUT1*, SOUT2*	O	Serial Data Out (Serial 1, 2)	2
CTS1#*, CTS2#*	I	Clear To Send (Serial 1, 2)	2
RTS1#*, RTS2#*	O	Request To Send (Serial 1, 2)	2
DSR1#*, DSR2#*	I	Data Set Ready (Serial 1, 2)	2
DTR1#*, DTR2#*	O	Data Terminal Ready (Serial 1,2)	2
DCD1#*, DCD2#*	I	Data Carrier Detect (Serial 1, 2)	2
RI1#*, RI2#*	I	Ring Indicator (Serial 1, 2)	2
PARALLEL PORT (COMBINED WITH ISA BUS AND IPC)			
PE*	I	Paper End	1
SLCT*	I	SELECT	1
BUSY#*	I	BUSY	1
ERR#*	I	ERROR	1
ACK#*	I	Acknowledge	1
PDDIR#*	O	Parallel Device Direction	1
STROBE#*	O	PCS / STROBE#	1
INIT#*	O	INIT	1
AUTPFDX#*	O	Automatic Line Feed	1
SLCTIN#*	O	SELECT IN	1
PPD[7:0]*	I/O	Data Bus	8
PCMCIA INTERFACE (COMBINED WITH PCI / CARDBUS)			
RESET*	O	Reset	1
A[25:0]*	O	Address Bus	26
D[15:0]*	I/O	Data Bus	16
IORD#*, IOWR#*	O	I/O Read and Write	2
DREQ#* / WP* / IOIS16#*	I	DMA Request // Write Protect // I/O Size is 16 bit	1
BVD1*, BVD2*	I	Battery Voltage Detect	2
READY#*/BUSY#*/IREQ#*	I	Ready / Busy // Interrupt Request	1
WAIT#*	I	Wait	1
INPACK#*	I	Input Port Acknowledge	1
OE#* / TCw*	O	Output Enable // DMA Terminal Count	1
WE#* / TCr*	O	Write Enable // DMA Terminal Count	1
DACK* / REG#*	O	DMA Acknowledge // Register	1
CD1#*, CD2#*	I	Card Detect	2

Table 6-2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
CE1#*, CE2#*	O	Card Enable	2
VS1#*, VS2#*	I	Voltage Sense	2
VCC5_EN*	O	Power Switch control : 5v power	1
VCC3_EN*	O	Power Switch control : 3.3v power	1
VPP_PGM*	O	Power Switch control : Program power	1
VPP_VCC*	O	Power Switch control : VCC power	1
CARDBUS INTERFACE (COMBINED WITH PCI / PCMCIA)			
CCLKRUN*	I/O	Clock	1
CRST#*	O	Reset	1
CSTSCHG#*	I	System Change	1
CAD[31:0]*	I/O	Address / Data	32
CBE[3:0]*	I/O	Bus Commands / Byte Enables	4
CFRAME#*	I/O	Cycle Frame	1
CTRDY#*	I/O	Target Ready	1
CIRDY#*	I/O	Initiator Ready	1
CSTOP#*	I/O	Stop Transaction	1
CDEVSEL#*	I/O	Device Select	1
CPAR*	I/O	Parity Signal Transactions	1
CSERR#*	I	System Error	1
CPERR#*	I/O	Parity Error	1
CBLOCK#*	I/O	PCI Lock	1
CCD[2:1]*	I	Card Detect	2
CINT#*	I	Interrupt Request	1
CREQ#*	I	Request	1
CGNT#*	O	Grant	1
PCI INTERFACE (COMBINED WITH PCMCIA / CARDBUS)			
AD[31:0]*	I/O	Address / Data	32
BE[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
PCI_REQ#[2:0]*	I	PCI Request	3
PCI_GNT#[2:0]*	O	PCI Grant	3
PCI_INT[3:0]*	I	PCI Interrupt Request	4

Table 6-2. Definition of Signal Pins

Signal Name	Dir	Description	Qty
MONITOR INTERFACE			
RED, GREEN, BLUE	O	Red, Green, Blue	3
VSYNC*	I/O	Vertical Sync	1
HSYNC*	I/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
SCL / DDC[1]*	I/O	I C Interface - Clock / Can be used for VGA DDC[1] signal	1
SDA / DDC[0]*	I/O	I C Interface - Data / Can be used for VGA DDC[0] signal	1
TFT INTERFACE (COMBINED WITH SERIAL 1)			
R[5:0], G[5:0], B[5:0]	O	Red, Green, Blue	18
FPLINE	O	Horizontal Sync	1
FPFRAME	O	Vertical Sync	1
DE	O	Data Enable	1
ENAVDD	O	Enable Vdd of flat panel	1
ENVCC	O	Enable Vcc of flat panel	1
PWM	O	PWM back-light control	1
MISCELLANEOUS			
SPKRD*	O	Speaker Device Output	1
SCAN_ENABLE	I	Test Pin - Reserved	1

Note; *denotes theat the pin is V_{5T} (see Section 24)

6.2. SIGNAL DESCRIPTIONS

6.2.2 BASIC CLOCKS AND RESETS

SYSRSTI# *System Reset/Power good.* This input is low when 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 and CLK24M.
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 Industrial device, the TTL signal should be provided on XTALO.

PCI_CLKI *33MHz PCI Input Clock*

This signal must be connected to a clock generator and is usually connected to PCI_CLKO.

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

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 a divided down version of 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 ISA bus Clock signal. It is also used with ISA_CLK as the multiplexor control lines for the Interrupt Controller Interrupt input lines.

CLK14M *ISA bus synchronisation clock.* This is the buffered 14.318 Mhz clock to the ISA bus. This clock also provides the reference clock to the frequency synthesizer that generates GCLK2X and DCLK.

HCLK *Host Clock.* This is the host 1X clock. Its frequency can vary from 50 to 75 MHz. All host transactions and PCI transactions are synchronized to this clock. Host transactions executed by the DRAM controller are also driven by this clock.

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

GCLK2X *80MHz Graphics Clock.* This is the Graphics 2X clock, which drives the graphics engine and 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. For further details, refer to Section 7.1.3 bit 4.

6.2.3 MEMORY INTERFACE

MA[11:0] *Memory Address.* 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.* 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.
MD20-0 are also used as inputs at the rising edge of PWGD to latch in power-up configuration information into the ADPC strap registers.

RAS#[3:0] *Row Address Strobe.* 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.* 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.* 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.

6.2.4 LOCAL BUS INTERFACE (Combined with ISA Bus)

PA[21:0] *Memory Address.* This is the 22-bit Local Bus Address

PD[15:0] *Data Bus.* This is the 16-bit bidirectional Local Bus Data bus.

PRDY# *Ready.* This input signals the Local Bus Ready state.

PWR#1 *Memory and I/O Write signal* for MS Byte

PWR#0 *Memory and I/O Write signal* for LS Byte*.*

PRD#1 *Memory and I/O Read signals* for MS Byte*.*

PRD#0 *Memory and I/O Read signals* for LS Byte*.*

FCS#[1:0], IOCS#[3:0] *Flash Memory and I/O Chip select.*

6.2.5 ISA BUS INTERFACE

LA[23:17] *Unlatched Address.* These unlatched ISA Bus pins address bits 23-17 on 16-bit devices. When the 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.

SA[19:0] *Unlatched Address.* These are the 20 low bits of the system address bus of ISA. 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 are the external ISA databus pins.

IOCHRDY *IO Channel Ready.* 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 Industrial. The STPC Industrial 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 Industrial since the access to the system memory can be considerably delayed due to CRT refresh or a write back cycle.

ALE *Address Latch Enable.* This is the address latch enable output of the ISA bus and is asserted by the STPC Industrial 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 Industrial.
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 Industrial generates SMEMR# signal of the ISA bus only when the address is below one MByte or the cycle is a refresh cycle.

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

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 Industrial 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 Industrial does not drive IOCS16# (similar to PC-AT design). An ISA master access to an internal register of the STPC Industrial 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 Industrial 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 Industrial 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.

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.

RTCRW# *Real Time Clock RW#.* This pin is used as RTCRW#. This signal is asserted for any I/O write to port 71h.

RTCDS# *Real Time Clock DS*. This pin is used as RTCDS. This signal is asserted for any I/O read to port 71h.

RTCAS# *Real time clock address strobe.* This signal is asserted for any I/O write to port 70h.

RMRTCCS# *ROM/Real Time clock chip select.* This pin is a multi-function pin. 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 an 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.

IRQ_MUX[3:0] *Multiplexed Interrupt Request.* These are the ISA bus interrupt signals. They are to be encoded before connection to the STPC Industrial 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 IRQ# pin of the RTC.

6.2.6 IPC (Combined with Serial Interface)

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

6.2.7 KEYBOARD/MOUSE INTERFACE

KBCLK, *Keyboard Clock line.* Keyboard data is latched by the controller on each negative clock edge produced on this pin. The keyboard can be disabled by pulling this pin low by software control.

KBDATA, *Keyboard Data Line.* 11-bits of data are shifted serially through this line when data is being transferred. Data is synchronised to KBCLK.

MCLK, *Mouse Clock line.* Mouse data is latched by the controller on each negative clock edge produced on this pin. The mouse can be disabled by pulling this pin low by software control.

MDATA, *Mouse Data Line.* 11-bits of data are shifted serially through this line when data is being transferred. Data is synchronised to MCLK.

6.2.8 SERIAL INTERFACE (Serial 1 combined with TFT Interface) (Serial 2 combined with IPC)

SIN1, SIN2 *Input Serial input.* Data is clocked in using RCLK/16.

SOUT1, SOUT2 *Serial Output.* Data is clocked out using TCLK/16 (TCLK=BAUD#).

DCD1#, DCD2# *Input Data carrier detect.*

RI1#, RI2# *Input Ring indicator.*

DSR1#, DSR2# *Input Data set ready.*

CTS1#, CTS2# *Input Clear to send.*

RTS1#, RTS2# *Output Request to send.*

DTR1#, DTR2# *Output Data terminal read.*

6.2.9 PARALLEL PORT (Combined with ISA Bus an IPC)

PE *Paper End.* Input status signal from printer.

SLCT *Printer Select.* Printer selected input.

BUSY# *Printer Busy*.
Input status signal from printer.

ERR# *Error*. Input status signal from printer.

ACK# *Acknowledge.*
Input status signal from printer.

PDDIR# *Parallel Device Direction.*
Bidirectional control line output.

STROBE# *PCS/Strobe#.*
Data transfer strobe line to printer.

INIT# *Initialize Printer.* This output sends an initialize command to the connected printer.

AUTPFDX# *Automatic Line feed.* This output sends a command to the connected printer to automatically generate line feed on received carriage returns.

SLCTIN# *Select In.* Printer select output.

PPD[7-0] *Printer Data Lines* Data transfer lines to printer. Bidirectional depending on modes.

6.2.10 PCMCIA INTERFACE (Combined with PCI / Cardbus)

RESET *Card Reset.* This output forces a hard reset to a PC Card.

A[25:0] *Address Bus.* These are the 25 low bits of the system address bus of the PCMCIA bus. These pins are used as an input when an PCMCIA bus owns the bus and are outputs at all other times.

D[15:0] *I/O Data Bus (PCMCIA).* These are the external PCMCIA databus pins.

CA[25-0] *Card Address*. Used with the lower 11 bits of the ISA Address Bus to generate the Card Address.

IORD# *I/O Read.* This output is used with REG# to gate I/O read data from the PC Card, (only when REG# is asserted).

IOWR# *I/O Write*. This output is used with REG# to gate I/O write data from the PC Card, (only when REG# is asserted).

WP *Write Protect.* This input indicates the status of the Write Protect switch (if fitted) on memory PC Cards (asserted when the switch is set to write protect).

BVD1, BVD2 *Battery Voltage Detect.* These inputs will be generated by memory PC Cards that include batteries and are an indication of the condition of the batteries. BVD1 and BVD2 are kept asserted high when the battery is in good condition.

READY#/BUSY#/IREQ# *Ready/busy/Interupt request.* This input is driven low by memory PC Cards to signal that their circuits are busy processing a previous write command.

WAIT# *Bus Cycle Wait.* This input is driven by the PC Card to delay completion of the memory or I/O cycle in progress.

OE# *Output Enable.* OE# is an active low output which is driven to the PC Card to gate Memory Read data from memory PC Cards.

WE#/PRGM# *Write Enable.* This output is used by the host for gating Memory Write data. WE# is also used for memory PC Cards that have programmable memory.

REG# *Attribute Memory Select.* This output is inactive (high) for all normal accesses to the Main Memory of the PC Card. I/O PC Cards will only respond to IORD# or IOWR# when REG# is active (low). Also see Section 6.2.6

CD1#, CD2# *Card Detect.* These inputs provide for the detection of correct card insertion. CD#1 and CD#2 are positioned at opposite ends of the connector to assist in the detection process. These inputs are internally grounded on the PC Card therefore they will be forced low whenever a card is inserted in a socket.

CE1#, CE2# *Card Enable*. These are active low output signals provided from the PCIC. CE#1 enables even Bytes, CE#2 odd Bytes.

ENABLE# *Enable.* This output is used to activate/select a PC Card socket. ENABLE# controls the external address buffer logic.C card has been detected (CD#1 and CD#2 = ’0’).

ENIF# *ENIF*. This output is used to activate/select a PC Card socket.

EXT_DIR *EXternal Transreceiver Direction Control.* This output is high during a read and low during a write. The default power up condition is write (low). Used for both Low and High Bytes of the Data Bus.

VCC_EN#, VPP1_EN0, VPP1_EN1, VPP 2_EN0, VPP2_EN1 *Power Control.* Five output signals used to control voltages (VPP1, VPP2 and VCC) to a PC Card socket. Also see Section 16.7.5

GPI# *General Purpose Input. This signal is hardwired to 1.*

6.2.11 CARDBUS INTERFACE (Combined with PCI / PCMCIA)

For card bus pinouts, refer to the PCI pinout.

6.2.12 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.

BE[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 Industrial owns the bus and outputs when the STPC Industrial 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 Industrial 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 Industrial is the target of the current bus transaction. It is used as an input when STPC Industrial 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 Industrial initiates a bus cycle on the PCI bus. It is used as an input during the PCI cycles targeted to the STPC Industrial 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 Industrial and is used as an output when a PCI master cycle is targeted to the STPC Industrial.

DEVSEL# *I/O Device Select.* This signal is used as an input when the STPC Industrial 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 Industrial 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 Industrial initiated PCI transaction. Its assertion by either the STPC Industrial 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.

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

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

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

6.2.13 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.* This pin is an input driving the digital to analog converters. This allows an external voltage reference source to be used.

RSET *Resistor Current Set.* This is the reference current input to the RAMDAC. 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.

They can instead be used for accessing I C devices on board. DDC1 and DDC0 correspond to SCL and SDA respectively.

6.2.14 FLAT PANEL INTERFACE SIGNALS (Combined with Serial 1)

FPFRAME, *Vertical Sync. pulse Output.*

FPLINE, *Horizontal Sync. Pulse Output.*

DE, *Data Enable.*

R5-0, *Red Output.*

G5-0, *Green Output.*

B5-0, *Blue Output*.

ENAVDD *Enable VDD of Flat Panel.*

ENVCC *Enable VCC of Flat Panel.*

PWM *PWM Back-Light Control.*

6.2.15 MISCELLANEOUS

SPKRD *Speaker Drive.* This is the output to the speaker and is the AND of the counter 2 output with bit 1 of Port 61h and drives an external speaker driver. This output should be connected to a 7407 type high voltage driver.

SCAN_ENABLE *Reserved*. This pin is reserved for Test and Miscellaneous functions. It has to be set to ‘0’ or connected to ground in normal operation.

Table 6-3. Signals sharing the same pin

ISA BUS / IPC	LOCAL BUS	PARALLEL PORT	SERIAL INTERFACE
LA[23:22]	FCS#[0], PRD#[1]		
LA[21:20]	PA[21:20]		
LA[19:17]	PRD#[0], PWR#[1:0]		
SA[19:1]	PA[19:1]		
SA[0]	PRDY#		
SD[15:0]	PD[15:0]		
BHE#	FCS#[1]		
MEMR#, MEMW#	IOCS[3:2]		
SMEMR#, SMEMW#	IOCS[1:0]		
GPIOCS#		PE	
IOCHRDY		SLCT	
IOR#		BUSY#	
IOW#		ERR#	
MASTER#		ACK#	
MCS16#		PDDIR#	
IOCS16#		INIT#	
REF#		AUTPFDX#	
AEN		SLCTIN#	
IOCHCK#		PPD[7]	
RTCRW#		PPD[5]	
RTCDS#		PPD[4]	
RTCAS#		PPD[3]	
RMRTCCS#		PPD[2]	
ALE		PPD[1]	
DACK_ENC[0:2]			DCD2#, DSR2#, SIN2
DREQ_MUX[0:1]			CTS2#, RTS2#
TC			SOUT2

TFT INTERFACE	SERIAL 1
B[0,1]	DCD1#, CTS1#
G[0,1]	DSR1#, RTS1#
R[0,1]	SIN1, SOUT1

PCI	CARDBUS	PCMCIA
	CCLK	A[16]
	CRST#	RESET
AD[31:27]	CAD[31:27]	D[10,9,1,8,0]
AD[26:20]	CAD[26:20]	A[0:6]

PCI	CARDBUS	PCMCIA
AD[19]	CAD[19]	A[25]
AD[18]	CAD[18]	A[7]
AD[17]	CAD[17]	A[24]
AD[16]	CAD[16]	A[17]
AD[15]	CAD[15]	IOWR#
AD[14]	CAD[14]	A[9]
AD[13]	CAD[13]	IORD#
AD[12]	CAD[12]	A[11]
AD[11]	CAD[11]	OE# / TCw
AD[10]	CAD[10]	CE[2]
AD[9]	CAD[9]	A[10]
AD[8:0]	CAD[8:0]	D[15,7,13,6,12,5,11,4,3]
BE[3]	CBE[3]	DACK/REG#
BE[2]	CBE[2]	A[12]
BE[1]	CBE[1]	A[8]
BE[0]	CBE[0]	CE[1]
FRAME#	CFRAME#	A[23]
TRDY#	CTRDY#	A[22]
IRDY#	CIRDY#	A[15]
STOP#	CSTOP#	A[20]
DEVSEL#	CDEVSEL#	A[21]
PAR	CPAR	A[13]
	CPERR#	A[14]
SERR#	CSERR#	WAIT
LOCK#	CBLOCK#	A[19]
PCIREQ#[2]	CREQ#	INPACK#
PCIREQ#[1]	CCD1	CD1#
PCIREQ#[0]	CSTSCHG#	BVD1
PCIGNT#[2]	CGNT#	WE# / TCr
PCIGNT#[1]	CCD2	CD2#
PCIGNT#[0]		BVD2
PCI_INT[3]		VCC3_EN
PCI_INT[2]		VCC5_EN
PCI_INT[1]		VPP_PGM
PCI_INT[0]	CINT#	READY#
	CLKRUN	DREQ# / WP / IOIS16#
		A[18]

Table 6-4. Pinout.

Pin #	Pin name
C4	SYSRSTI#
A3	SYSRSTO#
AB25	XTALI
AB23	XTALO
G25	PCI_CLKI
H23	PCI_CLKO
B20	ISA_CLK
A20	ISA_CLK2X
AC26	CLK14M
H26	HCLK
J26	DEV_CLK
AC15	GCLK2X
AD16	DCLK
AE13	MA[0]
AC12	MA[1]
AF13	MA[2]
AD12	MA[3]
AE14	MA[4]
AC14	MA[5]
AF14	MA[6]
AD13	MA[7]
AE15	MA[8]
AD14	MA[9]
AF15	MA[10]
AE16	MA[11]
AD15	RAS#[0]
AF16	RAS#[1]
AC17	RAS#[2]
AE18	RAS#[3]
AD17	CAS#[0]
AF18	CAS#[1]
AE19	CAS#[2]
AF19	CAS#[3]
AD18	CAS#[4]
AE20	CAS#[5]
AC19	CAS#[6]
AF20	CAS#[7]
AD19	MWE#
AE21	MD[0]
AC20	MD[1]
AF21	MD[2]
AD20	MD[3]
AE22	MD[4]
AF22	MD[5]
AD21	MD[6]
AE23	MD[7]
AC22	MD[8]
AF23	MD[9]
AD22	MD[10]
AE24	MD[11]
AD23	MD[12]
AF24	MD[13]
AE26	MD[14]
AD25	MD[15]
AD26	MD[16]
AC25	MD[17]
AC24	MD[18]
AB24	MD[19]
AB26	MD[20]
AA25	MD[21]
Y23	MD[22]
AA24	MD[23]
AA26	MD[24]
Y25	MD[25]
Y26	MD[26]
Y24	MD[27]
W25	MD[28]
V23	MD[29]
W26	MD[30]
W24	MD[31]
V25	MD[32]
V26	MD[33]
U25	MD[34]
V24	MD[35]
U26	MD[36]
U23	MD[37]
T25	MD[38]
U24	MD[39]
T26	MD[40]
R25	MD[41]
R26	MD[42]
T24	MD[43]
P25	MD[44]
R23	MD[45]
P26	MD[46]
R24	MD[47]
N25	MD[48]
N23	MD[49]
N26	MD[50]
P24	MD[51]
M25	MD[52]
N24	MD[53]
M26	MD[54]
L25	MD[55]
M24	MD[56]
L26	MD[57]
M23	MD[58]
K25	MD[59]
L24	MD[60]
K26	MD[61]
K23	MD[62]
J25	MD[63]
B1	PA[0]
P1	LA[17] / PWR#[0]
N3	LA[18] / PWR#[1]
R2	LA[19] / PRD#[0]
C1	LA[20] / PA[20]
C2	LA[21] / PA[21]
P3	LA[22] / PRD#[1]
R1	LA[23] / FCS#[0]
P4	SA[0] / PRDY#
J2	SA[1] / PA[1]
H3	SA[2] / PA[2]
H1	SA[3] / PA[3]
J4	SA[4] / PA[4]
H2	SA[5] / PA[5]
G3	SA[6] / PA[6]
G1	SA[7] / PA[7]
G2	SA[8] / PA[8]
F1	SA[9] / PA[9]
F3	SA[10] / PA[10]
G4	SA[11] / PA[11]
F2	SA[12] / PA[12]
E1	SA[13] / PA[13]
E3	SA[14] / PA[14]
E4	SA[15] / PA[15]
E2	SA[16] / PA[16]
D1	SA[17] / PA[17]

PIN DESCRIPTION

Pin #	Pin name
D3	SA[18] / PA[18]
D2	SA[19] / PA[19]
P2	SD[0] / PD[0]
M3	SD[1] / PD[1]
N1	SD[2] / PD[2]
M4	SD[3] / PD[3]
N2	SD[4] / PD[4]
L3	SD[5] / PD[5]
M1	SD[6] / PD[6]
M2	SD[7] / PD[7]
L1	SD[8] / PD[8]
K3	SD[9] / PD[9]
L2	SD[10] / PD[10]
K4	SD[11] / PD[11]
K1	SD[12] / PD[12]
J3	SD[13] / PD[13]
K2	SD[14] / PD[14]
J1	SD[15] / PD[15]
T2	BHE# / FCS#[1]
R3	MEMR# / IOCS#[3]
T1	MEMW# / IOCS#[2]
R4	SMEMR# / IOCS#[1]
U2	SMEMW# / IOCS#[0]
AB2	IOCHRDY / SLCT
AB1	IOR# / BUSY#
Y3	GPIOCS# / PE
AA3	IOW# / ERR#
AC2	MASTER# / ACK#
AB4	MCS16# / PDDIR#
AB3	IOCS16# / INIT#
AD2	REF# / AUTPFDX#
AC3	AEN / SLCTIN#
E25	IOCHCK# / PPD[7]
E26	PPD[6]
F24	RTCRW# / PPD[5]
D25	RTCDS# / PPD[4]
E23	RTCAS# / PPD[3]
D26	RMRTCCS# / PPD[2]
E24	ALE / PPD[1]
C25	PPD[0]
AC1	STROBE#
D5	IRQ_MUX[0]
A4	IRQ_MUX[1]
C5	IRQ_MUX[2]
B3	IRQ_MUX[3]
AD1	SPKRD
V3	DACK_ENC[0]/DCD2#
Y2	DACK_ENC[1]/DSR2#
W4	DACK_ENC[2] / SIN2
Y1	DREQ_MUX[0]/CTS2#
W3	DREQ_MUX[1]/RTS2#
AA2	TC / SOUT2
Y4	DTR2#
AA1	RI2#
U4	SIN1 / R[0]
V1	SOUT1 / R[1]
V2	CTS1 / B[1]
U3	RTS1# / G[1]
U1	DSR1# / G[0]
W2	DTR1#
T3	DCD1# / B[0]
W1	RI1#
F25	KBCLK
F26	KBDATA
G24	MCLK
G23	MDATA
D18	RESET
C18	A[0]
A17	A[1]
D17	A[2]
B16	A[3]
C17	A[4]
A16	A[5]
B15	A[6]
A15	A[7]
C16	A[8]
B14	A[9]
D15	A[10]
A14	A[11]
C15	A[12]
B13	A[13]
D13	A[14]
A13	A[15]
C14	A[16]
B12	A[17]
C13	A[18]
A12	A[19]
B11	A[20]
A11	A[21]
D12	A[22]
B10	A[23]
C11	A[24]
A10	A[25]
D10	D[0]
B9	D[1]
C10	D[2]
A9	D[3]
B8	D[4]
C9	D[5]
B7	D[6]
D8	D[7]
A7	D[8]
B6	D[9]
D7	D[10]
A6	D[11]
C7	D[12]
A5	D[13]
C6	D[14]
B4	D[15]
B22	IORD#
D22	IOWR#
D24	WP
A18	BVD1
C26	BVD2
A21	READY#
C19	WAIT#
A25	INPACK#
C22	OE#
B18	WE#
B19	REG#
B24	CD1#
A24	CD2#
B23	CE1#
C23	CE2#
C20	VS1#
A19	VS2#
D20	VCC5_EN
C21	VCC3_EN

Pin #	Pin name
B21	VPP_PGM
A22	VPP_VCC
AD4	RED
AF4	GREEN
AE5	BLUE
AF3	VSYNC
AE4	HSYNC
AF5	VREF_DAC
AE6	RSET
AF6	COMP
AE3	SDA / DDC[1]
AF2	SCL / DDC[0]
AE7	B[2]
AF7	G[2]
AD7	R[2]
AE8	B[3]
AC9	G[3]
AF8	R[3]
AD8	B[4]
AE9	G[4]
AF9	R[4]
AE10	B[5]
AD9	G[5]
AF10	R[5]
AC10	RESERVED
AD10	FPLINE
AE11	FPFRAME
AF11	DE
AE12	ENAVDD
AF12	ENVCC
AD11	PWM
C8	SCAN_ENABLE
AD5	VDD_DAC1
AC5	VDD_DAC2
AE17	VDD_GCLK_PLL
AF17	VDD_DCLK_PLL
K24	VDD_ZCLK_PLL
H25	VDD_DEVCLK_PLL
J24	VDD_HCLK_PLL

Pin #	Pin name
A8	RESERVED
A23	RESERVED
B5	RESERVED
B17	RESERVED
C12	RESERVED
D6	VDD
D11	VDD
D16	VDD
D21	VDD
F4	VDD
F23	VDD
L4	VDD
L23	VDD
T4	VDD
T23	VDD
AA4	VDD
AA23	VDD
AC6	VDD
AC11	VDD
AC16	VDD
AC21	VDD
AC7	VSS_DAC1
AD6	VSS_DAC2
G26	VSS_DLL
H24	VSS_DLL
A1	VSS
A2	VSS
A26	VSS
B2	VSS
B25	VSS
B26	VSS
C3	VSS
C24	VSS
D4	VSS
D9	VSS
D14	VSS
D19	VSS
D23	VSS
H4	VSS
J23	VSS
L11:16	VSS
M11:16	VSS

Pin #	Pin name
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	VSS
AE2	VSS
AE25	VSS
AF1	VSS
AF25	VSS
AF26	VSS

6.5 Update History for Pin Description chapter

The following changes have been made to the Pin Description Chapter on the 13/10/99.

Section	Change	Text
Table 6-1.	Updated	Pin numbering updated.
Figure 6-1.	Updated	Pin numbering on each output
Table 6-2.	Removed	Reference to IPC.
Table 6-4.	Changed	AD16 changed from Reserved to DCLKOUT
Table 6-4.	Changed	AC10 changed from DCLKOUT to Reserved

The following changes have been made to the Pin Description Chapter on the 17/08/99.

Section	Change	Text
Table 6-2.	Added	“GPIOCS#” plus definition
6.2.5	Added	“**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.”
Table 6-3.	Removed	“ISAOE”
Table 6-4.	Removed	“ISAOE”
Table 6-1	Removed	“CLK” Signal removed
Table 6-2.	Removed	“CLK” Signal removed
Table 6-3	Removed	“CLK” Signal removed

The following changes have been made to the Pin Description Chapter on the 17/08/99.

Section	Change	Text
6.1.	Removed	VDD5, 5V power supply for PCI ESD protection
6.2.	Replaced	Pin A8, A23, B5, B17 & C12 from VDD% to RESERVED
6.2.	Replaced	“**PWR#1, PWR#0,** *Memory and I/O Write Signals*” with **“PWR#1** *Memory and I/O Write signal* for MS Byte **PWR#0** *Memory and I/O Write signal* for LS Byte*.”*
6.2.	Replaced	**“PRD#1, PRD0#,** *Memory and I/O Read Signals*” with **“PRD#1** *Memory and I/O Read signals* for MS Byte*.”* **“PRD#0** *Memory and I/O Read signals* for LS Byte*.”*

Update History for Pin Description chapter

7 STRAP OPTION

This chapter defines the STPC Industrial Strap Options and their location

Memory Data Lines	Refer to	Designation	Location	Actual Settings	Set to '0'	Set to '1'
MD0	-	Reserved	-	-	-	-
MD1	-	Reserved	-	-	-	-
MD2	DRAM Bank 1	Speed	Index 4A,bit 2	User defined	70 ns	60 ns
MD3		Speed	Index 4A,bit 3	Pull up		
MD4		Type	Index 4A,bit 4	User defined	EDO	FPM
MD5	DRAM Bank 0	Speed	Index 4A,bit 5	User defined	70 ns	60 ns
MD6		Speed	Index 4A,bit 6	Pull up		
MD7		Type	Index 4A,bit 7	User defined	EDO	FPM
MD8	-	Reserved	-	-	-	-
MD9	-	Reserved	-	-	-	-
MD10	DRAM Bank 3	Speed	Index 4B,bit 2	User defined	70 ns	60 ns
MD11		Speed	Index 4B,bit 3	Pull up		
MD12		Type	Index 4B,bit 4	User defined	EDO	FPM
MD13	DRAM Bank 2	Speed	Index 4B,bit 5	User defined	70 ns	60 ns
MD14		Speed	Index 4B,bit 6	Pull up		
MD15		Type	Index 4B,bit 7	User defined	EDO	FPM
MD16		Reserved	Index 4C,bit 0			
MD17	PCI Clock	PCI_CLKO Divisor	Index 4C,bit 1	User defined	HCLK / 3	HCLK / 2
MD18	Host Clock	HCLK Pad Direction	Index 4C,bit 2	User defined	External	Internal
MD19	Graphics Clock	GCLK2x Pad Direction	Index 4C,bit 3	User defined	External	Internal
MD20	DOT Clock	DCLK Pad Direction	Index 4C,bit 4	User defined	External	Internal
MD21		Reserved		Pull up		
MD22		External IPC Debug Option	Index 5F,bit 1	Pull up	External IPC	Internal IPC
MD23	-	Reserved	Index 5F,bit 2	Pull up	-	-
MD24	HCLK	HCLK PLL Speed	Index 5F,bit 3	User defined	000	25 MHz
MD25			Index 5F,bit 4	User defined	001	33 MHz
MD26			Index 5F,bit 5	User defined	010	40 MHz
					011	50 MHz
					100	60 MHz
					101	66 MHz
					110	75 MHz
					111	80 MHz
MD27		Reserved		Pull down		
MD28		Reserved		Pull down		
MD29		Reserved		Pull down		
MD30		Reserved		Pull down		
MD31		Reserved		Pull down		
MD32		Reserved		Pull down		
MD33		Reserved		Pull up		
MD34		Reserved		Pull down		
MD35		Reserved		Pull up		
MD 36		Reserved		Pull up		
MD 37		Reserved		Pull up		
MD 38		Reserved		Pull up		

Memory Data Lines	Refer to	Designation	Location	Actual Settings	Set to '0'	Set to '1'
MD 39		Reserved		Pull up		
MD 40		PCMCIA or PCI i/f	3C,bit 0	User defined	PCI	PCMCIA
MD 41		Local Bus or ISA i/f	3C,bit 1	User defined	ISA	Local Bus
MD 42		Key Board & Mouse	3C,bit 2	User defined	External	Internal
MD 43		Parallel Port	3C,bit 3	User defined	External	Internal
MD 44	Serial Port	UART1	3C,bit 4	User defined	External	Internal
MD 45		UART2	3C,bit 5	User defined	External	Internal
MD 46		Reserved	3C,bit 6	Pull down		
MD 47		Reserved	3C,bit 7	Pull down		
MD 48	TFT	Outputs on RFU pads	3D,bit 0	User defined	Disable	Enable
MD 49	Cardbus Socket	5V Availability	3D,bit 1	User defined	Not Available	Available
MD 50		3.3V Availability	3D,bit 2	User defined	Not Available	Available
MD 51		x.xV Available	3D,bit 3	User defined	Not Available	Available
MD 52		y.yV Available	3D,bit 4	User defined	Not Available	Available
MD 53		Reserved		Pull uo		
MD 56		Reserved		Pull up		
MD 57		Reserved		Pull down		
MD 58		Reserved		Pull up		
MD 59		Reserved		Pull down		

7.1 STRAP OPTION REGISTER DESCRIPTION

7.1.1 STRAP REGISTER 0 INDEX 4AH (STRAP0)

Bits 7-0, This register reflect the status of pins MD[7:0] respectively. They are expected to be connected on the system board to the SIMM configuration pins as follows:

Bit Sampled	Description
Bit 7	SIMM 0 DRAM type
Bits 6-5	SIMM 0 speed
Bit 4	SIMM 1 DRAM type
Bits 3-2	SIMM 1 speed
Bits 1-0	Reserved

Note that the SIMM speed and type information read here is meant only for the software and is not used by the hardware. The software must program the Host and graphics DRAM controller configuration registers appropriately based on these bits.

This register defaults to the values sampled on MD[7:0] pins after reset.

7.1.2 STRAP REGISTER 1 INDEX 4BH (STRAP1)

Bits 7-0, This register reflect the status of pins MD[15:8] respectively. They are expected to be connected on the system board to the SIMM configuration pins as follows:

Bit Sampled	Description
Bit 7	SIMM 2 DRAM type
Bits 6-5	SIMM 2 speed
Bit 4	SIMM 3 DRAM type:
Bits 3-2	SIMM 3 speed
Bit 1-0	Reserved

Note that the SIMM speed and type information read here is meant only for the software and is not used by the hardware. The software must program the Host and graphics dram controller configuration registers appropriately based on these bits.

This register defaults to the values sampled on MD[15:8] pins after reset.

7.1.3 STRAP REGISTER 2 INDEX 4CH (STRAP2)

Bits 4-0 of this register reflect the status of pins MD[20:16] respectively. Bit 5 of this register reflect the status of pin MD[23]. Bit 4 is writeable, writes to other bits in this register have no effect.

They are use by the chip as follows:

Bits 7-5, Reserved

Bit 4, This bit reflects the **value sampled on MD[20] pin** and controls the Dot clock (DCLK) source. Note this bit is writeable as well as readable.

Bit 3, This bit reflects the **value sampled on MD[19] pin** and controls the Graphics clock source.

Bit 2, This bit reflects the **value sampled on MD[18] pin** and controls the Host/CPU clock source as follows: setting to ’0’: External. HCLK pin is an input, setting to ’1’: Internal. HCLK pin is an output and is connected to the internal frequency synthesizer output.

Bit 1, This bit reflects the **value sampled on MD[17] pin** and controls the PCI clock output as follows:

Setting to ’0’, the PCI clock output = HCLK / 3
Setting to ’1’, the PCI clock output = HCLK / 2

Bit 0, Reserved.

This register defaults to the values sampled on MD[23] & MD[20:16] pins after reset.

7.1.4 STRAP REGISTER 3 INDEX 3CH (STRAP3)

Bits 7-0 of this register reflect the status of pins MD[47:40] respectively.

They are use by the chip as follows:

Bit 7-6, Reserved.

Bit 5, UART2 internal or external. This bit reflects the **value sampled on MD[45] pin** and controls the UART2 I/F as follows:

Setting to ’0’, UART2 is external.

Setting to ’1’, UART2 is internal.

Bit 4, UART1 internal or external and additional TFT outputs. This bit reflects the **value sampled on MD[44] pin** and controls the UART1 I/F and the additional TFT I/F as follows:

Setting to ’0’, UART1 is external and an additional 6 TFT outputs (lowest bits - 2 red, 2 green and 2 blue) are enabled.

Setting to ’1’, UART1 is internal.

Note that when strap option testbus enabled (see Section 7.1.3 Strap Register 2 bit 0) is driven to 1 it takes priority over this strap which becomes meaningless.

Bit 3, Parallel Port internal or external. This bit reflects the **value sampled on MD[43] pin** and controls the Parallel Port i/f as follows:

Setting to ’0’, the Parallel Port is external

Setting to ’1’, the Parallel Port is internal

Bit 2, KB/Mouse internal or external. This bit reflects the **value sampled on MD[42] pin** and controls the KB/Mouse controller i/f as follows:

Setting to ’0’, the KB/Mouse controller is external

Setting to ’1’, the KB/Mouse controller is internal

Bit 1, Local Bus i/f or ISA I/F. This bit reflects the **value sampled on MD[41] pin** and sets whether the Local Bus i/f or the ISA i/f is available at the device i/f as follows:

Setting to ’0’, selectes the ISA I/F

Setting to ’1’, selectec the Local Bus I/F

Bit 0 PCMCIA I/F or PCI I/F. This bit reflects the **value sampled on MD[40] pin** and sets whether the PCMCIA i/f or the PCI i/f is available at the device i/f as follows:

Setting to ’0’, selects the PCI I/F

Setting to ’1’, selects the PCMCIA I/F

This register defaults to the values sampled on MD[47:40] pins after reset.

7.1.5 STRAP REGISTER 4 INDEX 3Dh (STRAP4)

Bits 5-0 of this register reflect the status of pins MD[53:48] respectively.

They are use by the chip as follows:

Bits 7-5 Reserved.

Bit 4, y.y V present on board. This bit reflects the **value sampled on MD[52] pin** and is used to notify the Cardbus socket management unit if the y.y V vcc voltage (where y.y is less than x.x) is present on board as follows

Setting to ’0’, y.y V Vcc voltage is not available

Setting to ’1’: y.y V Vcc voltage is available.

Bit 3, x.x V present on board. This bit reflects the **value sampled on MD[51] pin** and is used to notify the Cardbus socket management unit if the x.x V vcc voltage (where x.x is less than 3.3) is present on board as follows:

Setting to ’0’, x.x V Vcc voltage is not available.

Setting to ’1’: x.x V Vcc voltage is available.

Bit 2, 3.3 V present on board. This bit reflects the **value sampled on MD[50] pin** and is used to notify the Cardbus socket management unit if the 3.3 V vcc voltage is present on board as follows:

Setting to ’0’, 3.3 V vcc voltage is not available.

Setting to ’1’, 3.3 V vcc voltage is available.

Bit 1, 5 V present on board. This bit reflects the **value sampled on MD[49] pin** and is used to notify the Cardbus socket management unit if the 5 V vcc voltage is present on board as follows:

Setting to ’0’, 5 V vcc voltage is not available.

Setting to ’1’, 5 V vcc voltage is available.

Bit 0, This bit reflects the **value sampled on MD[48] pin** and is used to enable the TFT controller outputs on pads RFU0-RFU11 as follows:

RFU0	: R[2]
RFU1	: R[3]
RFU2	: R[4]
RFU3	: R[5]
RFU4	: G[2]
RFU5	: G[3]
RFU6	: G[4]
RFU7	: G[5]
RFU8	: B[2]
RFU9	: B[3]
RFU10	: B[4]
RFU11	: B[5]

This register defaults to the values sampled on MD[53:48] pins after reset.

7.1.6 HCLK PLL STRAP REGISTER 0 INDEX 5FH (HCLK_STRAP0)

Bits 5-0 of this register reflect the status of pins MD[26:21] respectively.

They are use by the chip as follows:

Bits 7-6, Reserved.

Bits 5-3, These pins reflect the **value sampled on MD[26:24] pins** respectively and control the Host clock frequency synthesizer.

Bit 2-1, Reserved.

Bit 0, Reserved.

This register defaults to the values sampled on above pins after reset.

7.2 UPDATE HISTORY FOR STRAP OPTIONS CHAPTER

The following Changes have been made to the Strap Options Chapter on the 17/08/99.

MD[30:27] from 1111 to 0000.
MD[35:31] from 10110 to 10100.
MD[53] from blank to Pull up
MD[58:56] from 010 to 101.

The following changes have been made to the Strap Options Chapter on the 13/08/99.

Strap Option MD[22:21] changed to Reserved.

Added Paragraph 6.1, Strap Register Description.

Update History for Strap Options chapter

8 LIST OF REGISTERS

This chapter lists all the registers accessable by software.

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
7	**Power on strap Registers**			0022h	
7.1.1	Strap Register 0	Strap0	Configuration	0023h	Index 04Ah
7.1.2	Strap Register 1	Strap1	Configuration		Index 04Bh
7.1.3	Strap Register 2	Strap2	Configuration		Index 04Ch
7.1.4	Strap Register 3	Strap3	Configuration		Index 3Ch
7.1.5	Strap Register 4	Strap4	Configuration		Index 3Dh
7.1.6	HCLK PLL Strap Register 0	HCLK_Strap	Configuration		Index 05Fh
9.5	**Cache related registers**			0022h	
9.5.1	Cache Architecture Register 0	Cash_Arc0	Configuration	0023h	Index 020h
9.5.2	Cache Architecture Register 1	Cash_Arc1	Configuration		Index 021h
9.5.3	Cache Architecture Register 2	Cash_Arc2	Configuration		Index 022h
9.6	**Address decode related registers**			0022h	
9.6.1	Memory Hole Control Register	Mem_Hole	Configuration	0023h	Index 024h
9.6.2	Shadow Control Register 0	Shadow_0	Configuration		Index 025h
9.6.3	Shadow Control Register 1	Shadow_1	Configuration		Index 026h
9.6.4	Shadow Control Register 2	Shadow_2	Configuration		Index 027h
9.6.5	Shadow Control Register 3	Shadow_3	Configuration		Index 028h
9.6.6	VGA Decode Register	VGA_Dec	Configuration		Index 029h
9.7	**Host DRAM controller registers**			0022h	
9.7.1	DRAM Bank 0 Register	DRAM_B0	Configuration	0023h	Index 030h
9.7.2	DRAM Bank 1 Register	DRAM_B1	Configuration		Index 031h
9.7.3	DRAM Bank 2 Register	DRAM_B2	Configuration		Index 032h
9.7.4	DRAM Bank 3 Register	DRAM_B3	Configuration		Index 033h
9.7.5	Memory Bank Width Register	Mem_Width	Configuration		Index 034h
9.7.6	DRAM Bank 0 Timing Parameter Register	DRAM_T0	Configuration		Index 035h
9.7.10	Graphics Memory Size Register	Graph_Mem	Configuration		Index 036h
9.7.11	Memory Type Register	Mem_Type	Configuration		Index 037h
9.7.7	DRAM Bank 1 Timing Parameter Register	DRAM_T1	Configuration		Index 038h
9.7.12	DRAM Refresh Register	DRAM_Ref	Configuration		Index 039h
9.7.8	DRAM Bank 2 Timing Parameter Register	DRAM_T2	Configuration		Index 03Ah
9.7.9	DRAM Bank 3 Timing Parameter Register	DRAM_T3	Configuration		Index 03Bh
10.5	**North Bridge PCI related registers**				IDSEL = ad[11]

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
10.3	Configuration Address Register	Config_ Address	IO	0xCF8h	
10.4	Configuration Data Register	Config_Data	IO	0xCFCh	
10.5.1	North Bridge Vendor Identification Register	NB_V_ID	PCI Config		Index 0x0h
10.5.2	North Bridge Device Identification Register	NB_D_ID	PCI Config		Index 0x2h
10.5.3	North Bridge PCI Command Register	NB_Com	PCI Config		Index 0x4
10.5.4	North Bridge PCI Status Register	NB_Stat	PCI Config		Index 0x6h
10.5.5	North Bridge PCI Revision Id Register	NB_R_ID	PCI Config		Index 0x8h
10.5.6	North Bridge Device Class Code Register	NB_C_Code	PCI Config		Index 0x9h
10.5.7	North Bridge Header Type Register	NB_Head	PCI Config		Index 0xEh
10.5.8	North Bridge Control Register	NB_Cont	PCI Config		Index 0x50h
10.5.9	North Bridge PCI Error Status Register	NB_E_Stat	PCI Config		Index 0x54h
10.7	**South Bridge PCI Function 0 Configuration Registers**			0xCF8h	IDSEL = ad[12]
10.7.1	South Bridge Vendor Identification Register	SB_V_ID0	PCI config	0xCFCh	Index 0x0h
10.7.2	South Bridge Device Identification Register	SB_D_ID0	PCI Config		Index 0x2h
10.7.3	South Bridge PCI Command Register	SB_Com_0	PCI Config		Index 0x4h
10.7.4	South Bridge PCI Status Register	SB_Stat0	PCI Config		Index 0x6h
10.7.5	South Bridge PCI Revision Id Register	SB_R_ID0	PCI Config		Index 0x8h
10.7.6	South Bridge Device Class Code Register	SB_C_Code0	PCI Config		Index 0x9h
10.7.7	South Bridge Header Type Register	SB_Head0	PCI Config		Index 0xEh
10.7.8	South Bridge Miscellaneous Register	SB_Misc0			Index 040h
11.4	**ISA standard Registers**				
11.4.1	DMA 1 Channel 0 Base and Current Address	DMA1_CBA0	IO	0000h	
11.4.1	DMA 1 Channel 0 Base and Current Count	DMA1_CBC0	IO	0001h	
11.4.1	DMA 1 Channel 1 Base and Current Address	DMA1_CBA1	IO	0002h	
11.4.1	DMA 1 Channel 1 Base and Current Count	DMA1_CBC1	IO	0003h	
11.4.1	DMA 1 Channel 2 Base and Current Address	DMA1_CBA2	IO	0004h	
11.4.1	DMA 1 Channel 2 Base and Current Count	DMA1_CBC2	IO	0005	
11.4.1	DMA 1 Channel 3 Base and Current Address	DMA1_CBA3	IO	0006h	
11.4.1	DMA 1 Channel 3 Base and Current Count	DMA1_CBC3	IO	0007h	
11.4.1	DMA 1 Read Status / Write Command Register	DMA1_RSWC	IO	0008h	
11.4.1	DMA 1 Request Register	DMA1_RR	IO	0009h	

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.4.1	DMA 1 Read Command / Write Single Mask Register	DMA1_RCWSM	IO	000Ah	
11.4.1	DMA 1 Mode Register	DMA1_Mode	IO	000Bh	
11.4.1	DMA 1 Set / Clear Byte Pointer Flip - Flop	DMA1_SCBPFF	IO	000Ch	
11.4.1	DMA 1 Read Temp Register / Master Clear	DMA1_RTMC	IO	000Dh	
11.4.1	DMA 1 Clear Mask / Clear All Request	DMA1_CMCAR	IO	000Eh	
11.4.1	DMA 1 Read / Write all Mask Register Bits	DMA1_RWMB	IO	000Fh	
11.4.2	Interrupt Controller 1 Registers	IC_1	IO	0020h	
11.4.2	Interrupt Controller 1 Mask Register	IC_1MR	IO	0021h	
11.4.3	Interval Timer Register Counter 0 Count	IT_0	IO	0040h	
11.4.3	Interval Timer Register Counter 1 Count	IT_1	IO	0041h	
11.4.3	Interval Timer Register Counter 2 Count	IT_2	IO	0042h	
11.4.3	Command Mode Register	IT_3	IO	0043h	
11.4.4	Port Bh Register	Port _B	IO	0061h	
11.4.5	Port 60h Register	Port_60		0060h	
11.4.5	Port 64h Register	Port_64		0064h	
11.4.6	Port 70h Register	Port_70	IO	0070h	
11.4.7	Interrupt Controller 2 Registers	IC_2R	IO	00A0h	
11.4.7	Interrupt Controller 2 Mask	IC_2M	IO	00A1h	
11.4.8	DMA Controller 2 Registers	DMA_Cont2	IO		
11.4.8	DMA 2 Channel 0 Base and Current Address	DMA2_CBA0	IO	00C0h	
11.4.8	DMA2 Channel 0 Base and Current Count	DMA2_CBC0	IO	00C2h	
11.4.8	DMA 2 Channel 1 Base and Current Address	DMA2_CBA1	IO	00C4h	
11.4.8	DMA 2 Channel 1 Base and Current Count	DMA2_CBC1	IO	00C6h	
11.4.8	DMA 2 Channel 2 Base and Current Address	DMA2_CBA2	IO	00C8h	
11.4.8	DMA 2 Channel 2 Base and Current Count	DMA2_CBC2	IO	00CAh	
11.4.8	DMA 2 Channel 3 Base and Current Address	DMA2_CBA3	IO	00CCh	
11.4.8	DMA 2 Channel 3 Base and Current Count	DMA2_CBC3	IO	00CEh	
11.4.8	DMA 2 Read Status / Write Command Register	DMA2_RSWC	IO	00D0h	
11.4.8	DMA 2 Request Register	DMA2_RR	IO	00D2h	
11.4.8	DMA 2 Read Command / Write Single Mask Register	DMA2_RCWSM	IO	00D4h	
11.4.8	DMA 2 Mode Register	DMA2_Mode	IO	00D6h	
11.4.8	DMA 2 Set / Clear Byte Pointer Flip - Flop	DMA2_SCBPFF	IO	00D8h	
11.4.8	DMA 2 Read Temporary / Master Clear	DMA2_RTMC	IO	00DAh	
11.4.8	DMA 2 Clear Mask / Clear All Requests Register	DMA2_CMCAR	IO	00DCh	

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
11.4.8	DMA 2 Read / Write all Mask Register Bits	DMA2_RWMRB	IO	00DEh	
11.4.9	DMA Page Registers	DMA_Page	IO		
11.4.9	DMA Page Registers Port 80h (reserved)	Port_80	IO	0080h	
11.4.9	DMA Page Register Channel 2	DMA_PRC2	IO	0081h	
11.4.9	DMA Page Register Channel 3	DMA_PRC3	IO	0082h	
11.4.9	DMA Page Register Channel 1	DMA_PRC1	IO	0082h	
11.4.9	DMA Page Register Port 84h (Reserved)	Port_84	IO	0084h	
11.4.9	DMA Page Register Port 85h (Reserved)	Port_85	IO	0085h	
11.4.9	DMA Page Register Port 86h (Reserved)	Port_86	IO	0086h	
11.4.9	DMA Page Register Channel 0	DMA_PRC0	IO	0087h	
11.4.9	DMA Page Register Port 87h	Port_87	IO	0088h	
11.4.9	DMA Page Register Channel 6	DMA_PRC6	IO	0089h	
11.4.9	DMA Page Register Channel 7	DMA_PRC7	IO	008Ah	
11.4.9	DMA Page Register Channel 5	DMA_PRC5	IO	008Bh	
11.4.9	DMA Page Register Port 8Bh (Reserved)	Port_8B	IO	008Ch	
11.4.9	DMA Page Register Port 8Ch (Reserved)	Port_8C	IO	008Dh	
11.4.9	DMA Page Register Port 8Dh (Reserved)	Port_8D	IO	008Eh	
11.4.9	DMA Page Register Port 8Eh (Reserved)	Port_8E	IO	008Fh	
11.5	**ISA Configuration Registers**			0022h	
11.5.1	Miscellaneous Control Register 0	Misc_Cont0	Configuration	0023h	Index 050h
11.5.2	Miscellaneous Control Register 1	Misc_Cont1	Configuration		Index 051h
11.5.3	PIRQA Routing control Register 0	PAR_Cont0	Configuration		Index 052h
11.5.4	PIRQB Routing control Register 0	PBR_Cont0	Configuration		Index 053h
11.5.5	PIRQC Routing control Register 0	PCR_Cont0	Configuration		Index 054h
11.5.6	PIRQD Routing control Register 0	PDR_Cont0	Configuration		Index 055h
11.5.7	Interrupt Level Control Register 0	IRQ_Lev_C_0	Configuration		Index 056h
11.5.8	Interrupt Level Control Register 1	IRQ_Lev_C_1	Configuration		Index 057h
11.5.9	IPC Configuration Register	IPC_Conf	Configuration		Index 001h
11.5.10	ISA I/O port select and sync. Register	ISA_Sync	Configuration		Index 059h
11.5.11	UART and Parallel Port IRQ Routing	IRQ_Rout	Configuration		Index 91h
12.3	**VGA registers**				
12.4	General Registers				
12.4.1	Motherboard Enable Register	MBEN		0x094h	
12.4.2	Add-in VGA Enable Register	ADDEN		0x46E8h	
12.4.3	Video Subsystem Enable 1 Register	VSE1		0x102h	
12.4.4	Video Subsystem Enable 2 Register	VSE2		0x3C3h	
12.4.5	Miscellaneous Output Register	MISC		0x3CC/ 0x3C2h	
12.4.6	Input Status Register #0	INP0		0x3C2h	
12.4.7	Input Status Register #1	INP1		0x3XAh	
12.5	Sequencer Registers				

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
12.5.1	Sequencer Index Register	SRX		0x03C4h	
12.5.2	Sequencer Reset Register	SR0		0x03C5h	Index 000h
12.5.3	Sequencer Clocking Mode Register	SR1			Index 001h
12.5.4	Sequencer Plane Mask Register	SR2			Index 002h
12.5.5	Sequencer Character Map Register	SR3			Index 003h
12.5.6	Sequencer Memory Mode Register	SR4			Index 004h
12.5.7	Extended Register Lock/Unlock Register	SR6			Index 006h
12.6	Graphics Controller Registers				
12.6.1	Graphics Controller Index Register	GRX		0x03CEh	
12.6.2	Graphics Set/Reset Register	GR0		0x03CFh	Index 000h
12.6.3	Graphics Enable Set/Reset Register	GR1			Index 001h
12.6.4	Graphics Color Compare Register	GR2			Index 002h
12.6.5	Raster Op/Rotate Count Register	GR3			Index 003h
12.6.6	Graphics Read Map Select Register	GR4			Index 004h
12.6.7	Graphics Mode Register	GR5			Index 005h
12.6.8	Graphics Miscellaneous Register	GR6			Index 006h
12.6.9	Graphics Color Don't Care Register	GR7			Index 007h
12.6.10	Graphics Bit Mask Register	GR8			Index 008h
12.7	Attribute Controller Registers				
12.7.1	Attribute Controller Index Register	ARX		0x3C0h	
12.7.2	Attribute Palette Registers	AR0 - ARF		0x3C1/ 0x3C0h	
12.7.3	Attribute Ctrl Mode Register	AR10		0x3C1/ 0x3C0h	
12.7.4	Attribute Ctrl Overscan Color Register	AR11		0x3C1/ 0x3C0h	
12.7.5	Attribute Color Plane Enable Register	AR12		0x3C1/ 0x3C0h	
12.7.6	Attribute Horz Pixel Panning Register	AR13		0x3C1/ 0x3C0h	
12.7.7	Attribute Color Select Register	AR14		0x3C1/ 0x3C0h	
12.8	**CRT Controller Registers**				
12.8.1	Index Register	CRX	*see Note 1*	0x3X4h	
12.8.2	Horizontal Total Register	CR0	*see Note 1*	0x3X5h	Index 000h
12.8.3	Horiz display End Register	CR1			Index 001h
12.8.4	Horiz Blanking Start Register	CR2			Index 002h
12.8.5	Horiz Blanking End Register	CR3			Index 003h
12.8.6	Horiz Retrace Start Register	CR4			Index 004h
12.8.7	Horizontal Retrace End Register	CR5			Index 005h
12.8.8	Vertical Total Register	CR6			Index 006h
12.8.9	Overflow Register	CR7			Index 007h
12.8.10	Screen A Preset Row Scan Register	CR8			Index 008h

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
12.8.11	Character Cell Height Register	CR9			Index 009h
12.8.12	Cursor Start Register	CRA			Index 00Ah
12.8.13	Cursor End Register	CRB			Index 00Bh
12.8.14	Start Address High Register	CRC			Index 00Ch
12.8.15	Start Address Low Register	CRD			Index 00Dh
12.8.16	Text Cursor Offset High Register	CRE			Index 00Eh
12.8.17	Text Cursor Offset Low Register	CRF			Index 00Fh
12.8.18	Vertical Retrace Start Register	CR10			Index 010h
12.8.19	Vertical Retrace End Register	CR11			Index 011h
12.8.20	Vertical Display End Register	CR12			Index 012h
12.8.21	Offset Register	CR13			Index 013h
12.8.22	Underline Location Register	CR14			Index 014h
12.8.23	Vertical Blanking Start reg	CR15			Index 015h
12.8.24	Vertical Blanking End Register	CR16			Index 016h
12.8.25	Mode Register	CR17			Index 017h
12.8.26	Line Compare Register	CR18			Index 018h
12.8.27	Graphics Control Data	CR22			Index 019h
12.8.28	Attribute Address Flipflop	CR24			Index 020h
12.8.29	Attribute Index Readback	CR26			Index 021h
12.9	**VGA Extended Registers**		*see Note 1*	0x3X4h	
12.9.1	Repaint Control Register 0	CR19	*see Note 1*	0x3X5h	Index 019h
12.9.2	Repaint Control Register 1	CR1A			Index 01Ah
12.9.3	Repaint Control Register 2	CR1B			Index 01Bh
12.9.4	Repaint Control Register 3	CR1C			Index 01Ch
12.9.5	Page Register 0	CR1D			Index 01Dh
12.9.6	Page Register 1	CR1E			Index 01Eh
12.9.7	Graphics Extended Enable Register	CR1F			Index 01Fh
12.9.8	Graphics Extended GBASE Register	CR20			Index 020h
12.9.9	Graphics Extended Aperture Register	CR21			Index 021h
12.9.10	Repaint Control Register 4	CR25			Index 025h
12.9.11	Repaint Control Register 5	CR27			Index 027h
12.9.12	Palette Control Register	CR28			Index 028h
12.9.13	Cursor Height Register	CR29			Index 029h
12.9.14	Cursor Color 0 Register A	CR2A			Index 02Ah
12.9.15	Cursor Color 0 Register B	CR2B			Index 02Bh
12.9.16	Cursor Color 0 Register C	CR2C			Index 02Ch
12.9.17	Cursor Color 1 Register A	CR2D			Index 02Dh
12.9.18	Cursor Color 1 Register B	CR2E			Index 02Eh
12.9.19	Cursor Color 1 Register C	CR2F			Index 02Fh
12.9.20	Graphics Cursor Address Register 0	CR30			Index 030h
12.9.21	Graphics Cursor Address Register 1	CR31			Index 031h
12.9.22	Graphics Cursor Address Register 2	CR32			Index 032h

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
12.9.23	Urgent Start Register	CR33			Index 033h
12.9.24	Displayed Frame Y Offset 0 Register	CR34			Index 034h
12.9.25	Displayed Frame Y Offset 1 Register	CR35			Index 035h
12.9.26	Interlace Half Field Start Register	CR39			Index 039h
12.9.27	Implementation Number Register	CR3A			Index 03Ah
12.9.28	Graphics Version Register	CR3B			Index 03Bh
12.9.29	DRAM Timing Parameter Register	CR3C			Index 03Ch
12.9.30	DRAM Arbitration control Register 0	CR3D			Index 03Dh
12.9.31	DRAM Arbitration control Register 1	CR3E			Index 03Eh
12.9.32	DDC Control Register	CR3F			Index 03Fh
12.12	**RAMDAC registers**				
12.12.1	Palette Pixel Mask Register	Pixel_Mask		0x3C6h	
12.12.2	Palette Read index Register	Read_Index		0x3C7h	
12.12.3	Palette State Register	Palette_State		0x3C7h	
12.12.4	Palette Write Index Register	Write_Index		0x3C8h	
12.12.5	Palette Data Register	Palette_Data		0x3C9h	
12.13	**DCLK Control registers**			022h	
12.13.1	DCLK Control Register 00	DCLK00	PCI Config	023h	Index 0x42h
12.13.2	DCLK Control Register 01	DCLK01	PCI Config		Index 0x43h
12.13.3	DCLK Control Register 10	DCLK10	PCI Config		Index 0x44h
12.13.4	DCLK Control Register 11	DCLK11	PCI Config		Index 0x45h
12.13.5	DCLK Control Register 20	DCLK20	PCI Config		Index 0x46h
12.13.6	DCLK Control Register 21	DCLK21	PCI Config		Index 0x47h
12.13.7	DCLK Control Register 30	DCLK30	PCI Config		Index 0x48h
12.13.8	DCLK Control Register 31	DCLK31	PCI Config		Index 0x49h
13.	**Gaphics Engine**				
13.5	**VGA Operand Sources**				
13.9.1	Back Ground Color Register	Background		8400000h	Index 0x004h
13.9.2	Cursor Coordinate Register	Cursor_XY			Index 0x11Ch
13.9.3	Top of Data FIFO Register	Data_Port			Index 0x804h
13.9.4	Destination Operand Base Address Register	Dst_Base			Index 0x018h
13.9.5	Destination Pitch Register	Dst_Pitch			Index 0x028h
13.9.6	Destination Operand Coordinate Register	Dst_XY		8410000h	
13.9.7	Foreground Color Register	Foreground		8400000h	Index 0x034h
13.9.8	Height Register	Height			Index 0x048h
13.9.9	Pattern Base Address Operand Register	Pattern			Index 0x058h
13.9.10	Pixel Depth Operand Register	Pixel_Depth			Index 0x07Ch
13.9.11	Raster Operation Register	ROP			Index 0x08Ch

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
13.9.12	Source Base Address Operand Register	Src_Base			Index 0x098h
13.9.13	Source Pitch Operand Register	Src_Pitch			Index 0x0ACh
13.9.14	Source Coordinate Register	Src_XY			Index 0x0BDh
13.9.15	Status Register	Status			Index 0x908
13.9.16	Width Register	Width			Index 0x0C8h
13.9.17	Extra Use Register	Xtra			Index 0x0D4h
13.9.18	SRC Transparency Compare Register	SRC_Transparency			Index 0xECh
13.9.19	DST Transparency Compare Register	DST_Transparency			Index 0xFCh
13.12	**Graphics clock registers**			022h	
13.12.1	GCLK Control Register 0	GCLK00	PCI Config	023h	Index 0x40h
13.12.2	GCLK Control Register 1	GCLK01	PCI Config		Index 0x41h
14.	**Video Controller**				
14.2	Video Pipeline Registers		*see Note 2*	X480000h	
14.2.1.1	Video Source Base Register	Video_Src_Base			Index 0x0h
14.2.1.2	Video Source Pitch Register	Video_Src_Pitch			Index 0x4h
14.2.1.3	Video Source Dimension Register	Video_Src_Dim			Index 0x8h
14.2.1.4	CRTC Burst Length Register	CRTC_Burst_Length			Index 0xCh
14.2.1.5	Video Burst Length Register	Video_Burst_Length			Index 0x10h
14.2.2	Destination Specification Registers				
14.2.2.1	Video Destination Register	Video_Dst_XY			Index 0x14h
14.2.2.2	Video Destination Dimension Register	Vid_Dst_Dim			Index 0x18h
14.2.3	Filter Control Registers				
14.2.3.1	Horizontal Scaling and Decimation Register	Horiz_Scl			Index 0x20h
14.2.3.2	Vertical Control and Decimation Register	Vert_Scl			Index 0x28h
14.2.3.3	Color Space Converter Specification Register	Clr_Con_Spec			Index 0x2Ch
14.2.4	**Video and Graphics mixing control Registers**				
14.2.4.1	Mix Mode Register	Mix_Mode			Index 0x30h
14.2.4.2	Color Key Register	CLR_Key			Index 0x34h
14.2.4.3	Chroma Key Low Register	CKL			Index 0x38h
14.2.4.4	Chroma Key High Register	CKH			Index 0x3Ch
14.2.4.5	Status Register	Filter_Stat			Index 0x40h

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
15.3	**TFT Configuration Register Description**	see 15.2	Configuration	22h	
15.2.4.	TFT BASE address	TFT_Base	Configuration	23h	0x1Dh
15.3.5.	Input Scan Line Active Pixel Count Register	Inp_Scan	Configuration		000h
15.3.6.	Input Horizontal Back Porsch Register	Inp_HBP	Configuration		001h
15.3.7.	FP Horizontal Synchronisation Width Register	FP_Sync	Configuration		002h
15.3.8.	FP Horizontal Back Porsch Register	FP_HBP	Configuration		004h
15.3.10.	FP Horizontal Front Porsch Register	FP_HFP	Configuration		005h
15.3.11.	FP Vertical Synchronisation Width Register	FP_VSW	Configuration		006h
15.3.12.	FP Vertical Back Porsch Register	FP_VBP	Configuration		007h
15.3.13.	FP Vertical Active Line Count Register	FP_VALC	Configuration		008h
15.3.14.	Interfce Control Register	TFT_IC	Configuration		009h
15.3.15.	PWM Control Register	PWM_cont	Configuration		00Ah
15.3.16.	Power Control Register	PWR_Cont	Configuration		00Bh
15.3.17.	Blank Red Register	B_Red	Configuration		00Ch
15.3.18.	Blank Green Register	B_Green	Configuration		00Dh
15.3.19.	Blank Blue Register	B_Blue	Configuration		00Eh
15.3.20.	Polarity Control Register	Pol_Cont	Configuration		00Fh
16.2	**PCCard Configuration Registers**			CF8h	IDSEL = ad[13]
16.2.1	Vendor ID Register	PC_V_ID	Configuration	CFCh	Index 0x00h
16.2.2	Device IDRegister	PC_D_ID	Configuration		Index 0x02h
16.2.3	Command Register	PC_Com	Configuration		Index 0x04h
16.2.4	Status Register	PC_Stat	Configuration		Index 0x06h
16.2.5	Revision ID Register	PC_R_ID	Configuration		Index 0x08h
16.2.6	Class Code Register	PC_C-Code	Configuraion		Index 0x09h
16.2.7	Cache Limit Size Register	PC_Cache	Configuration		Index 0x0Ch
16.2.8	Latency Timer Register	PC_Lat	Configuration		Index 0x0Dh
16.2.9	Header Type Register	PC_Head	Configuration		Index 0x0Eh
16.2.10	BIST Register	PC_BIST	Configuration		Index 0x0Fh
16.2.11	CardBus Socket Register	PC_CB_Sock	Configuration		Index 0x10h
16.2.12	Secondary Status Register	PC_Stat2	Configuration		Index 0x16h
16.2.13	PCI Bus Number Register	PC_PCI_Bus	Configuration		Index 0x18h
16.2.14	CardBus Number Register	PC_CB_Num	Configuration		Index 0x19h
16.2.15	Subordinate Bus Number Register	PC_Sub_Bus	Configuration		Index 0x1Ah
16.2.16	CardBus Latency Timer Register	PC_CB_Lat	Configuration		Index 0x1Bh
16.2.17	Memory Base Address 0 Register	PC_M_Base0	Configuration		Index 0x1Ch
16.2.18	Memory Limit Address 0 Register	PC_M_Limit0	Configuration		Index 0x20h
16.2.19	Memory Base Address 1 Register	PC_M_Base1	Configuration		Index 0x24h
16.2.20	Memory Limit Address 1 Register	PC_M_Limit1	Configuration		Index 0x28h
16.2.21	I/O Base Register 0 Lower Half	PC_IO_LH0	Configuration		Index 0x2Ch
16.2.22	I/O Limit Register 0 Lower Half	PC_Lim_LH0	Configuration		Index 0x30h
16.2.23	I/O Base Register 1 Lower Half	PC_IO_LH1	Configuration		Index 0x34h

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
16.2.24	I/O Limit Register 1 Lower Half	PC_Lim_LH1	Configuration		Index 0x38h
16.2.25	I/O Interrupt Line Register	PC_Int_Line	Configuration		Index 0x3Ch
16.2.26	Interrupt Pin Register	PC_Int_Pin	Configuration		Index 0x3Dh
16.2.27	Bridge Control Register	PC_B_Cont	Configuration		Index 0x3Eh
16.2.28	PCCard 16 bit IF Legacy Mode Base Address	PC_C16_Leg	Configuration		Index 0x44h
16.2.29	ERR_Enable	PC_ERR_EN	Configuration		Index 0x50h
16.2.30	P2H_Enable	PC_P2H_EN	Configuration		Index 0x52h
16.2.31	ERR_Status	PC_ERR_Stat	Configuration		Index 0x54h
16.2.32	ST_RSVD	PC_ST_RSVD	Configuration		Index0x58h
16.6.1	**PCMCI A On-Chip Register Set**	Set using Index 0x44h of PCCard Register			
16.6.2	**General Setup Registers**				
16.6.2.1	Identification and Revision	Id_Rev	Configuration		Index 000h
16.6.2.2	Interface Status	Int_Stat	Configuration		Index 001h
16.6.2.3	Power and RESETDRV Control	PRD_Cont	Configuration		Index 002h
16.6.2.4	Card Status Change	CSC	Configuration		Index 004h
16.6.2.5	Address Window Enable	AW_En	Configuration		Index 006h
16.6.2.6	Card Detect and General Control	C_Cont	Configuration		Index 016h
16.6.2.7	Global Control Register	Global	Configuration		Index 1Eh
16.6.3	**Interrupt Register**				
16.6.3.1	Interupt and General Control	Int_Cont			Index 003h
16.6.3.2	Card Status Change Interupt Configuration	CSCIC			Index 005h
16.6.4	**IO Registers**				
16.6.4.1	I/O Control	I/O_Cont			Index 007h
16.6.4.2	I/O Address 0 Start Low Byte	Add0_SLB			Index 008h
16.6.4.3	I/O Address 0 Start High Byte	Add0_SHB			Index 009h
16.6.4.4	I/O Address 0 Stop Low Byte	Add0_SLB			Index 00Ah
16.6.4.5	I/O Address 0 Stop High Byte	Add0_SHB			Index 00Bh
16.6.4.6	I/O Address 1 Start Low Byte	Add2_SLB			Index 00Ch
16.6.4.7	I/O Address 1 Start High Byte	Add2_SHB			Index 00Dh
16.6.4.8	I/O Address 1 Stop Low Byte	Add2_StLB			Index 00Eh
16.6.4.9	I/O Address 1 Stop High Byte	Add2_StHB			Index 00Fh
16.6.5	**Memory Registers**				
16.6.5.1	System Memory Address 0 Mapping Start Low Byte	Mem_Add0_SLB			Index 010h
16.6.5.2	System Memory Address 0 Mapping Start High Byte	Mem_Add0_SHB			Index 011h
16.6.5.3	System Memory Address 0 Mapping Stop Low Byte	Mem_Add0_StLB			Index 012h

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
16.6.5.4	System Memory Address0 Mapping Stop High Byte	Mem_Add0_StHB			Index 013h
16.6.5.5	Card Memory Offset Address 0 Low Byte	CM_Add0_LB			Index 014h
16.6.5.6	Card Memory Offset Address 0 High Byte	CM_Add0_HB			Index 015h
16.6.5.7	System Memory Address 1 Mapping Start Low Byte	SM_Add1_MSLB			Index 018h
16.6.5.7	System Memory Address 1 Mapping Start High Byte	SM_Add1_MSHB			Index 019h
16.6.5.7	System Memory Address 1 Mapping Stop Low Byte	SM_Add1_MStLB			Index 01Ah
16.6.5.7	System Memory Address 1 Mapping Stop High Byte	SM_Add1_MStHB			Index 01Bh
16.6.5.7	Card Memory Offset Address 1 Low Byte	CM_Add1_LB			Index 01Ch
16.6.5.7	Card Memory Offset Address 1 High Byte	CM_Add1_HB			Index 01Dh
16.6.5.7	System Memory Address 2 Mapping Start Low Byte	SM_Add2_MSLB			Index 020h
16.6.5.7	System Memory Address 2 Mapping Start High Byte	SM_Add2_MSHB			Index 021h
16.6.5.7	System Memory Address 2 Mapping Stop Low Byte	SM_Add2_MStLB			Index 022h
16.6.5.7	System Memory Address 2 Mapping Stop High Byte	SM_Add2_MStHB			Index 023h
16.6.5.7	Card Memory Offset Address 2 Low Byte	CM_Add2_LB			Index 024h
16.6.5.7	Card Memory Offset Address 2 High Byte	CM_Add2_HB			Index 025h
16.6.5.7	System Memory Address 3 Mapping Start Low Byte	SM_Add3_MSLB			Index 028h
16.6.5.7	System Memory Address 3 Mapping Start High Byte	SM_Add3_MPHB			Index 029h
16.6.5.7	System Memory Address 3 Mapping Stop Low Byte	SM_Add3_MPStLB			Index 02Ah
16.6.5.7	System Memory Address 3 Mapping Stop High Byte	SM_Add3_MStHB			Index 02Bh
16.6.5.7	Card Memory Offset Address 3 Low Byte	CM_Add3_LB			Index 02Ch
16.6.5.7	Card Memory Offset Address 3 High Byte	CM_Add3_HB			Index 02Dh
16.6.5.7	System Memory Address 4 Mapping Start Low Byte	SM_Add4_MSLB			Index 030h
16.6.5.7	System Memory Address 4 Mapping Start High Byte	SM_Add4_MSHB			Index 031h
16.6.5.7	System Memory Address 4 Mapping Stop Low Byte	SM_Add4_MStLB			Index 032h
16.6.5.7	System Memory Address 4 Mapping Stop High Byte	SM_Add4_MStHB			Index 033h
16.6.5.7	Card Memory Offset Address 4 Low Byte	CM_Add4_LB			Index 034h
16.6.5.7	Card Memory Offset Address 4 High Byte	CM_Add4_HB			Index 035h
▨	▨	▨	▨	▨	▨
16.7	**CardBus Register Description**	Set using Index 0x10h of PCCard Register			
16.7.1	Socket Event Register	PC_SEV	Configuration		Index 0x00h
16.7.2	Socket Event Mask Register	PC_SEV_M	Configuration		Index 0x04h
16.7.3	Socket Force Event Register	PC_SFE	Configuration		Index 0xCh
16.7.4	Socket Present State Register	PC_SPS	Configuration		Index 0x08h

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
16.7.5	Socket Control Register	PC_S_Cont	Configuration		Index 0x10h
16.7.6	DMA DREQ Routing Configuration Register	PC_DRC	Configuration		Index 0x92h
17.2	**Local Bus Registers**		16 bit access	22h	
17.3	**Local Bus Address Decode Registers**			23h	
17.3.1	I/O Slot Base Register 0	IOAREG0			10h
17.3.1	I/O Slot Base Register 1	IOAREG1			11h
17.3.1	I/O Slot Base Register 2	IOAREG2			12h
17.3.1	I/O Slot Base Register 3	IOAREG3			13h
17.3.2	I/O Slot Mask Register 0	IOMREG0			14h
17.3.2	I/O Slot Mask Register 1	IOMREG1			15h
17.4	**Local Bus Timing Registers**				
17.4.1	Memory Timing Template 0	TIMEBANK0			16h
17.4.1	Memory Timing Template 1	TIMEBANK1			17h
17.4.2	I/O Timing Template 0	TIMEIO0			18h
17.4.2	I/O Timing Template 1	TIMEIO1			19h
17.4.2	I/O Timing Template 2	TIMEIO2			1Ah
17.4.2	I/O Timing Template 3	TIMEIO3			1Bh
17.5	**Local Bus Control Register**				
17.5.1	Control Register	CONTROL			1Ch
17.5.2	I/O Width Register	IOWIDTH			1Eh
18.4	**Keyboard / Mouse Controller Register**				
18.4.1	Output Buffer	Out_Buf	I/O	60h	
18.4.2	Input Buffer	In_Buf	I/O	60h	
18.4.3	Command Register	Com_Reg	I/O	64h	
18.4.4	Status Register	Stat_Reg	I/O	64h	
19.4	**Serial Port Register**			3F8h	
19.4.2	Reciever Buffer Register	RBR	I/O	2F8h	Index 000h
19.4.3	Transmiter Holding Register	THR	I/O		Index 000h
19.4.4	Interupt Enable Register	IER	I/O		Index 001h
19.4.5	Interupt Identification Register	IIR	I/O		Index 002h
19.4.9	FIFO Control Register	FCR	I/O		Index 002h
19.4.10	Line Control Register	LCR	I/O		Index 003h
19.4.11	Modem Control Register	MCR	I/O		Index 004h
19.4.12	Line Status Register	LSR	I/O		Index 005h
19.4.13	Modem Status Register	MSR	I/O		Index 006h
19.4.14	Scratch Register	SCR	I/O		Index 007h
19.4.15	Divisor Latch (LS)	DLL	I/O		Index 000h
19.4.15	Divisor Latch (MS)	DLM	I/O		Index 001h
20	**Parallel Port**				
20.2.5	**Configuration Register**				
20.2.5.2	Configuration Select Register	CSR	Configuration	3F0h	

Table 8-1. Registers described in this manual.

Section	Register Name	Mnemonic	Purpose	Address	Access type
20.2.5.3	Configuration Register 1	CR1	Configuration	3F1h	Index 001h
20.2.5.4	Configuration Register 4	CR4	Configuration	3F1h	Index 004h
20.2.6	**Parallel Port Registers**			see 20.2.6	
20.2.6.2	Status Register	P_Stat	I/O		Index 001h
20.2.6.3	Control Register	P_CTRL	I/O		Index 002h
21.2	**Power Management Controller Registers:**			0022h	
21.2.1	Timer Register 0	Timer0	Configuration	0023h	Index 060h
21.2.2	Timer Register 1	Timer1	Configuration		Index 061h
21.2.3	Timer Register 2	Timer2	Configuration		Index 08Dh
21.2.4	System Activity Enable Register 0	Sys_Activ_en0	Configuration		Index 062h
21.2.5	System Activity Enable Register 1	Sys_Activ_en1	Configuration		Index 063h
21.2.6	System Activity Enable Register 2	Sys_Activ_en2	Configuration		Index 064h
21.2.7	House-Keeping Activity Enable Register 0	HK_Activ_en0	Configuration		Index 065h
21.2.8	House-Keeping Activity Enable Register 1	HK_Activ_en1	Configuration		Index 066h
21.2.9	Peripheral Inactivity Detection Register 0	Perif_Inact0	Configuration		Index 067h
21.2.10	Peripheral Activity Detection Register 0	Perif_Act0	Configuration		Index 069h
21.2.11	Peripheral Activity Detection Register 1	Perif_Act1	Configuration		Index 06Ah
21.2.12	Address Range 0 Register 0	Add_ Rang0-0	Configuration		Index 06Bh
21.2.13	Address Range 0 Register 1	Add_ Rang0-1	Configuration		Index 06Ch
21.2.14	SMI Control Register 0	SMI_Cont0	Configuration		Index 071h
21.2.15	SMI Status Register 0	SMI_Stat0	Configuration		Index 073h
21.2.16	SMI Status Register 1	SMI_Stat1	Configuration		Index 074h
21.2.17	Peripheral Inactivity Status Register 0	Perif_Stat0	Configuration		Index 075h
21.2.18	Activity Status Register 0	Activ_Stat0	Configuration		Index 077h
21.2.19	Activity Status Register 1	Activ_Stat1	Configuration		Index 078h
21.2.20	Activity Status Register 2	Activ_Stat2	Configuration		Index 079h
21.2.21	PMU State Register	PMU	Configuration		Index 07Ah
21.2.22	General Purpose Register	GP	Configuration		Index 07Bh
21.2.23	Clock Control Register 0	Clk_Cont0	Configuration		Index 07Ch
21.2.24	Doze Timer Read Back Register	Doze	Configuration		Index 088h
21.2.25	Standby Timer Read Back Register	Standby	Configuration		Index 089h
21.2.26	Suspend Timer Read Back Register	Suspend	Configuration		Index 08Ah
21.2.27	House-Keeping Timer Read Back Register	HK_Timer	Configuration		Index 08Bh
21.2.28	Peripheral Timer Read Back Register	Perif_Timer	Configuration		Index 08Ch

Note 1: X can stand for B (Monochrome Display) or D (Color Display)

Note 2: X is the value of the G_Base and can range from 8h to Fh

Table 8-2. CPU Registers located in the ST 486 Manual

These registers are described in the ST 486 Datasheet

Section	Register Name	Mnemonic	Purpose	Address	Access type
4.4.4.4	**Configuration Registers**			22h	
4.4.4.4	Configuration Control 1	CCR1	IO	23h	C1h
4.4.4.4	Configuration Control 2	CCR2	IO		C2h
4.4.4.4	Configuration Contol 3	CCR3	IO		C3h
4.4.4.4	SMM Address Region	SMAR	IO		CDH
4.4.4.4	Device Indentification 0	DIR0	IO		FEh
4.4.4.4	Device Identification 1	DIR1	IO		FFh

8.1 UPDATE HISTORY FOR LIST OF REGISTERS CHAPTER

The following changes have been made to the List of Registers Chapter on 03/11/99.

Section	Change	Text
8	Added	Register 91h in the ISA section

The following changes have been made to the List of Registers Chapter on 11/10/99.

Section	Change	Text
8	Added	Video controller Registers
8	Added	“Set using Index 0x44h of PCCard Register” in line **PCMCI A On-Chip Register Set**

The following changes have been made to the List of Registers Chapter.

Section	Change	Text
8	Added	“TFT BASE address” with associated text.

9. DRAM CONTROLLER

9.1 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.

9.2 MEMORY CONTROLLER

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

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

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

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

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

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

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

9.3 Memory Address Map

9.3.1 00000000h-0009FFFFh (640K)

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

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

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

9.3.2 000A0000h-000BFFFFh (128K)

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

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

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

This segment is never cacheable.

9.3.3 000C0000h-000C3FFFh (16K)

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

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

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

9.3.4 000C4000h-000C7FFFh (16K)

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

9.3.5 000C8000h-000CBFFFh (16K)

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

9.3.6 000CC000h-000CFFFFh (16K)

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

9.3.7 000D0000h-000DFFFFh (64K)

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

9.3.8 000E0000h-000EFFFFh (64K)

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

9.3.9 000F0000h - 000FFFFFh (64K)

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

9.3.10 00100000H (1M) - TOP OF ADDRESSABLE DRAM MEMORY

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

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

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

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

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

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

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

9.3.12 FFFF0000 - FFFFFFFFh (4G-64K)

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

This is also true for address E0000h, D0000h and C0000h provided I/O register Index 51h (see Section 11.5.2) is set correctly.

9.3.13 EXTENDED GRAPHICS SEGMENT

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

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

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

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

This address segment is always considered non-cacheable.

9.3.14 MEMORY HOLE

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

9.3.15 SMM MEMORY

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

9.3.16 ADDRESSABLE DRAM MEMORY

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

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

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

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

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

The addressable DRAM memory =

TOPM - TOGM + MHOLE_SIZE + REMAP_SIZE

9.3.17 CPU ADDRESS TO DRAM ADDRESS MAPPING

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

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

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

STPC also defines a memory hole to allow the existence of memory devices on the PCI or ISA busses. The size of the CPU address space is increased by these memory holes, if they exist. CPU address space D0000h to EFFFFh is mapped to the add-in card BIOS area. If this ROM space is not shadowed, then the CPU address space is increased by another 128 KBytes (also see Section 9.6.1).

For example:

Total populated DRAM = 4 MBytes

Frame buffer size = 256 KBytes

Memory hole size = 1 MByte

Memory hole starting address = 200000h

Shadow feature for D0000h to EFFFFh = disabled

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

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

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

9.4 IO ADDRESS MAP .

Table 9-1. : IO map space

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

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

These register occupy the map in the IO space in the Table 9-1 below

Notes:

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

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

9.4.1 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 (seeTable 9-2) :

Table 9-2. PCI configuration address space

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

9.5 CACHE RELATED REGISTERS

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

This register controls various attributes of the L2 cache.

.Bit 7 **CPU pipeline access support**

Bit 7	CPU pipelined access
0	not supported
1	supported

.Bit 6 **Burst addressing order**

Bit 6	Burst order
0	Intel
1	linear

.Bit 5 **L1 write back indication**

Bit 5	L1 write back
0	Not supported
1	Supported

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

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

Bit 2 **Number of L2 banks.**

Bit 2	L2 Banks
0	One bank
1	Two banks

When programmed to 2 banks, L2 interleaving is enabled.

Bit 1 **L2 write back control.**

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

Bit 0 **L2 cache enable.**

Bit 1	L2 cache
0	disabled
1	enabled

This register defaults to 00h at reset.

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

This register controls various attributes of L2 cache.

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

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

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

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

Bits 3-2 **Source FIFO low water mark.** These bits control the degree of concurrency between a L1 cache line fill and start of the next memory access. A cache line wide read buffer is implemented.

Due to pipelining, it is possible that the buffer may be filled up ahead of drain. Then if the next access is also a read from memory, these bits determine when the next read will be kicked off relative to the drain of the current line from the read buffer. The optimal value is a function of the drain rate of the buffer which depends on the cache RAM type and the programmed burst parameters. A value of ‘0’ for this field is the least optimal value but will always work.

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

Bit 1 **Read around write enable.**

Bit 4	Read around write enable
0	reads can not proceed around any posted writes
1	reads can go around a posted write if it is to a different address to the posted writes

Bit 0 *Reserved*.

This register defaults to 00h at reset.

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

Bit 7 *Reserved.*

Bit 6 **Slow host data driver.**

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

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

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

Applicable to asynchronous SRAMs only. Must be ‘0’ for synchronous SRAMs.

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

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

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

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

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

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

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

This register defaults to FFh at reset.

9.6 ADDRESS DECODE RELATED REGISTERS

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

9.6.1 MEMORY HOLE CONTROL REGISTER - C.I. 24H (MEM_HOLE)

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

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

0 = disabled

1 = enabled

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

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

Bits 3-0 **Memory Hole Start Address.** These bits control the bits 23-20 of the memory hole starting address. The memory hole starting address must be aligned to the hole size.

This register defaults to 00h at reset.

Programming notes;

This memory hole is also non-cacheable.

9.6.2 SHADOW CONTROL REGISTER 0 - C.I. 25H (SHADOW_0)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

This register defaults to 00h at reset.

Programming Notes;

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

9.6.3 SHADOW CONTROL REGISTER 1 - C.I. 26H (SHADOW_1)

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

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

This register defaults to 00h at reset.

Programming Notes

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

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

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

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

This register defaults to 00h at reset

Programming Notes

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

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

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

Bit 7 **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 128KBytes of SMRAM. Physically this memory is located in the system memory behind the higher address range. This area of the system memory is normally unused since this address range is normally mapped to frame buffer which has its own memory.

When the CPU is operating in SMM, accesses in the range of A0000-BFFFFh goes to SMRAM instead of VGA frame buffer. The rest of the address map remains unchanged.

The address range A0000h-BFFFFh is always non-cacheable.

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

0 = cacheability disabled

1 = cacheability enabled

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

0 = cacheability disabled

1 = cacheability enabled

Bits 4-2 *Reserved*

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

0 = shadow disabled for read cycle

1 = shadow enabled for read cycle

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

0 = shadow disabled for write cycle

1 = shadow enabled for write cycle

This register defaults to 00h at reset.

Programming notes

T*he rest of the shadow RAM segments have the cacheability bits hardwired to '0' (disabled). This register also provides control over the address range for which ROM chip-select (ROMCS#) will be asserted allowing various BIOSes (system, video, disk etc.) to be implemented in a single part. Bit 7 of this register also provides accessibility to the SMM mode RAM (SMRAM).*

9.6.6 VGA DECODE REGISTER - C.I. 29H (VGA_DEC)

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

Bits 7-6 *Reserved*

Bit 5 stop g-clock (Graphics clock).

Bit 4 stop p-clock (PCI Clock).

Bit 3 stop d-clock (Dot clock).

Bit 2 **Palette Snoop Enable:**

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

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

Bit 1 **Internal VGA Disable.** This bit if set to a '0' will disable internal VGA. Otherwise if set to a '1', it will enable the internal VGA.

Bit 0 **Add-in Decode Enable.** This bit if set to a '0' will map the internal VGA to add-in card address space. Otherwise if set to a '1' it will map the VGA to mother-board address space.

This register defaults to 03h after reset.

9.7 DRAM CONTROLLER REGISTERS

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

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

9.7.1 DRAM BANK 0 REGISTER - C.I. 30H (DRAM__B0)

This 8-bit register controls the top address of the DRAM bank 0. Register bit 7-0 corresponding 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

9.7.2 DRAM BANK 1 REGISTER - C.I. 31H (DRAM_B1)

This register controls the top address of DRAM bank 1.

9.7.3 DRAM BANK 2 REGISTER - C.I. 32H (DRAM_B2)

This register controls the top address of DRAM bank 2.

9.7.4 DRAM BANK 3 REGISTER - C.I. 33H (DRAM_B3)

This register controls the top address of DRAM bank 3.

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

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.

9.7.6 BANK 0 TIMING PARAMETER - C.I. 35H (DRAM_T0)

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.

9.7.7 BANK 1 TIMING PARAMETER - C.I. 38H (DRAM_T1)

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.

9.7.8 BANK 2 TIMING PARAMETER - C.I. 3AH (DRAM_T2)

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.

9.7.9 BANK 3 TIMING PARAMETER - C.I. 3BH (DRAM_T3)

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.

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

This register defines the size of DRAM used by the 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.

9.7.11 MEMORY TYPE REGISTER - C.I. 37H (MEM_TYPE)

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.

9.7.12 DRAM REFRESH - C.I. 39H (DRAM_REF)

The register refresh disable bit also contains a 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.

Programming notes

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

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

9.8 DRAM INTERFACE

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

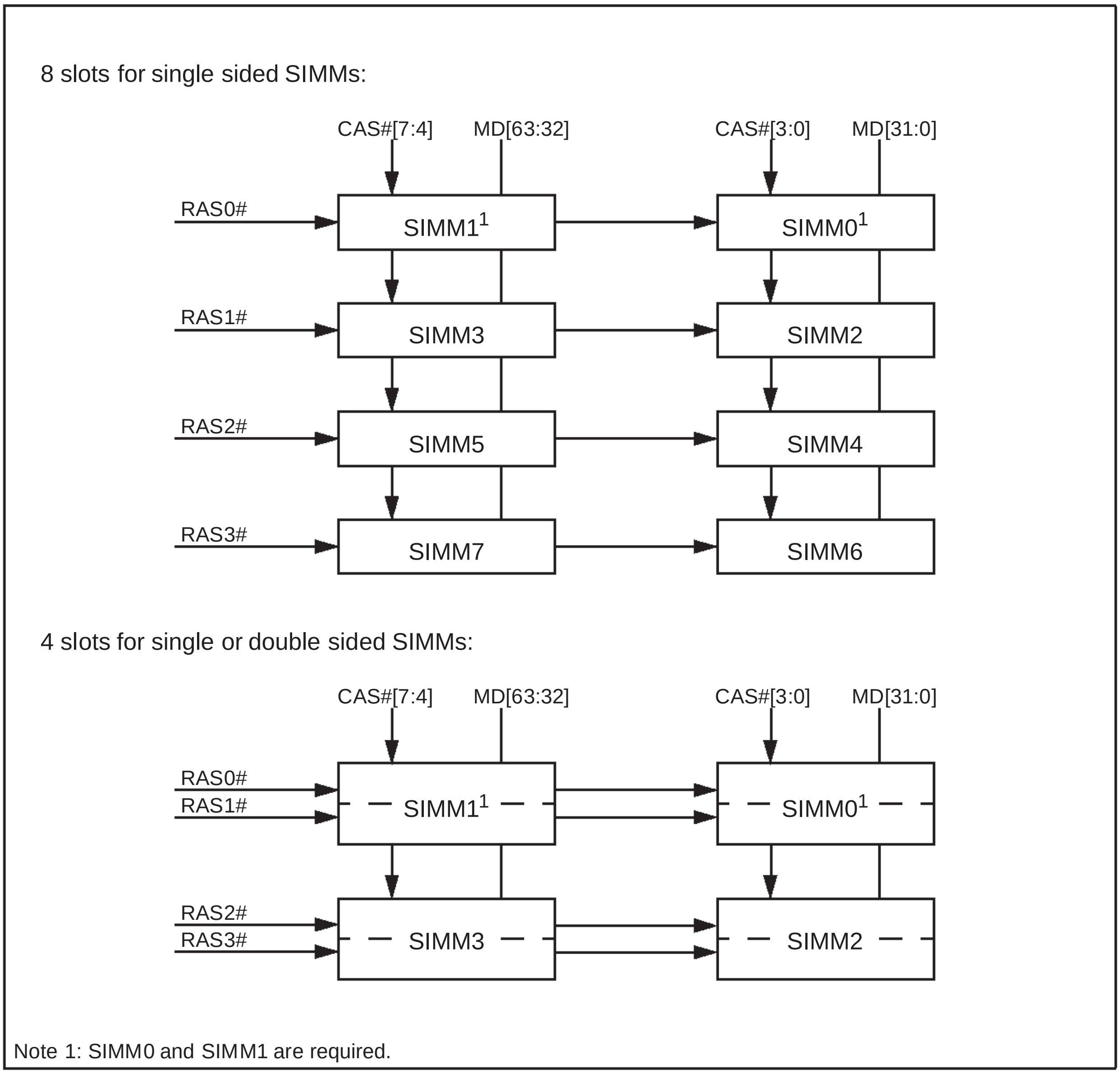
The following SIMMs are supported:

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

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

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

Figure 9-1. 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 9.9 , DRAM arbitration. The host clock domain DRAM controller processes host initiated read and write cycles as well as DRAM refresh cycles. Graphics, CRTC and Video Scaler read and write cycles are processed by the graphics domain DRAM controller (see section 5.7).

9.8.1 FAST PAGE MODE (FPM) DRAM

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

9.8.2 EDO DRAM

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

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

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

9.8.3 Host Address to MA bus Mapping

Graphics memory resides at the beginning of Bank 0. Host memory begins at the top of graphics memory and extends to the top of populated DRAM. The first step in the DRAM address generation is to add the Top of Graphics Memory (TOGM) to the Host Address (HA) to form the Remapped Address (RA);

$$RA = HA + TOGM$$

The second step is figure out which DRAM bank the remapped address falls on. The remapped address is compared to the top address for each bank (bk0ta ... bk3ta).

```
if (RA <= bk0ta) bk0sel = 1;
else if (RA <= bk1ta) bk1sel = 1;
else if (RA <= bk2ta) bk2sel = 1;
else if (RA <= bk3ta) bk3sel = 1;
```

Now the bank attributes can be retrieved from a lookup table to select the final DRAM row and column address mappings.

The bank attributes can be retrieved from a lookup table to select the final DRAM row and column address mappings. (Table 9-3)

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.

The 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 9-3. 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	X	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
16M	32-bit	A25	A24	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	X	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
16M	64-bit	A25	A24	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3

A = Corresponding remapped host address bit DRAM row address

MA = memory address

X = don’t care

9.8.4 DRAM MODULE PRESENCE AND TYPE DETECT

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

Table 9-4. DRAM Page Mode Detect

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

Table 9-5. 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

9.9 DRAM ARBITRATION:

Following agents compete for the system DRAM memory:

CPU

PCI masters

ISA masters

Graphics engine

CRT controller

Video Output

Video Input Port

Refresh controller

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

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

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

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

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

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

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

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

9.10 UPDATE HISTORY FOR DRAM CONTROLLER CHAPTER

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

Section	Change	Text
9.6.6	Added	p-clock (PCI-Clock)

The following changes have been made to the Memory Chapter.

Section	Change	Text
9.2	Replaced	“ ~~Bank 0 SIMMs must be used in pairs, 64 bits wide, since the Graphics Controller does not support 32 bit banks. Banks 1,2 and 3 can be 32 or 64 bits wide~~.” With “The internal Graphics Controller does not support 32 bit banks, therefore bank 0 SIMMs must be used in pairs, 64 bits wide. Banks 1,2 and 3 can be 32 or 64 bits wide.”
9.2	Added	“ is programmed by software and”
9.4	Replaced	“~~following map in the IO space”~~ With “space in the Table 9-1 below”
9.4	Added	Table title; ”IO Map Space”
9.4.1	Added	Table title “PCI configuration address space”
9.5.3	Replaced	“~~Local Bus Disable”~~ With “Reserved”
9.6.5	Replaced	“~~VGA. It is presented here in the sequence of the Configuation Index. The function is~~” With “VGA”
9.6.6	Added	“clock (Graphics”
9.6.6	Added	“clock (Dot“
9.7	Replaced	“~~Index~~” With “Configuration”
9.7.12	Replaced	~~“Refresh disable bit. This register also contains the number of host clocks for the DRAM refresh interval~~.” With “The register refresh disable bit also contains a number of host clocks for the DRAM refresh interval.
Table 9-3.	Replaced	“~~A24~~” With “X”
9.7.8	Added	JEDEC standard second generation 72 pin DRAM SIMM Modules have the following presence detect pins
9.8.4	Removed	“~~Table +~~ ~~S = short to Ground~~ ~~O = Open circuit~~ ~~The configuration codes are unique within the to 32MB density range. Software probing of memory locations is necessary to support a wider range of SIMM configurations.~~”

UPDATE HISTORY FOR DRAM CONTROLLER CHAPTER

10 PCI CONTROLLER

10.1 INTRODUCTION

The PCI bus is the main data communication link inside the STPC chip. The STPC can be configurated for having the PCI bus output for connecting external PCI devices.

Three PCI devices are present internal to the PC Industrial, the North Bridge, the South Bridge and the PCCard bridge.

The STPC contains also a PCI arbiter which arbitrates between the 3 internal devices and for up to two external PCI master devices. Figure 10.1: below shows the layout of the PCI controllers within the STPC

Please refer to “PCI specification 2.1”, from PCI-SIG, to have more details on PCI bus standard.

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

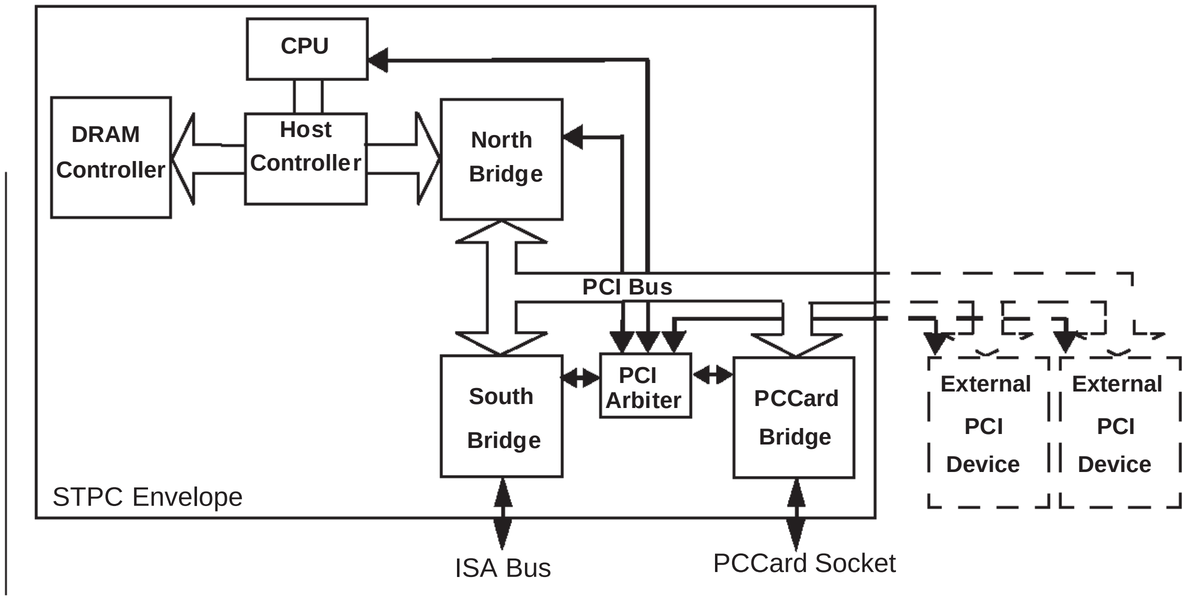
The North Bridge, as a PCI bus agent (North Bridge class) fully complies with PCI specification 2.1. The North Bridge also implements the PCI mandatory header registers in Type 0 PCI configuration space for easy porting of PCI-aware system BIOS. The North Bridge is assigned the Device Number 0xBh, which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

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

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

The Cardbus PCI bridge is detailled in Section 16 .

Figure 10.1- PCI Layout



10.1.1 PCI ERROR HANDLING:

Under control of South Bridge configuration registers, an ISA initiated transaction ending in target abort can generate a 1 PCICLK long pulse on SERR#, which in turn can be made to generate an NMI to the CPU.

10.1.2 PCI ARBITER:

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

10.1.3 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 South Bridge.

10.2 METHOD FOR ACCESSING THE PCI CONFIGURATION REGISTERS

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

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

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

- Reading or Writing 32 bit data at CFCh

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

An illustration of these registers is shown in Table 10.1& Table 10.2 below.

Table 10.1- Register CF8h

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

Table 10.2- Register CFCh

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

10.3 CONFIGURATION ADDRESS REGISTER IO CF8H (CONFIG_ADDRESS)

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 **PCI configuration register access enable.** When set to a ‘1’, host CPU IO cycles in address range CFCh-CFFh are converted to configuration cycles on the PCI bus. Otherwise if set to a ‘0’, IO cycles in this address range pass through as normal IO cycles on the PCI bus.

Bits 30-24 *Reserved.* Must be written to ‘0’. Read back as ‘0’.

Bits 23-16 **Bus Number.** This field selects a specific bus number in the system.
Bus Number 0 is assigned to the PCI bus directly behind the North Bridge. This field is driven on bits 23-16 of the AD bus during the address phase.

Bits 15-11 **Device Number.** This field selects a specific device on the bus. During a Type-0 configuration cycle, this field is decoded to assert the appropriate IDSEL line. The North Bridge is assigned the Device Number 0xBh, which corresponds to IDSEL on AD11 signal. The South Bridge is assigned the Device Number 0xCh, which corresponds to IDSEL on AD12 signal.

Bits 10-8 **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.

10.4 CONFIGURATION DATA REGISTER IO CFCH (CONFIG_DATA)

This is a 32-bit register accessible via 8-, 16- and

10.5 THE NORTH BRIDGE CONFIGURATION REGISTERS

Table 10.3- North Bridge Reset Values

31----------------------------- 24	23-----------------------------16	15 -----------------------------8	7----------------------------0	
Device ID : 02h	Device ID: 09h	Vendor ID : 10h	Vendor ID: 4Ah	00h
Status : 0280h		Command : 0007h		04h
Base class code: 00h	Sub class code: 00h	Program. Inter. Reg. : 00h	Revision : 00h	08h
	Header Type: 00h			0Ch
				...
Control Register : 00000000h				50h
Error Status Register : 00000000h				54h

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

Bus = 0h

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

Function = 0h (North Bridge / PCI)

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

10.5.1 NORTH BRIDGE VENDOR IDENTIFICATION REGISTER PCI CONFIG. OFFSET 0H-1H (NB_V_ID)

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

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.

10.5.2 NORTH BRIDGE DEVICE IDENTIFICATION REGISTER PCI CONFIG. OFFSET 2H-3H (NB_D_ID)

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

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.

10.5.3 NORTH BRIDGE PCI COMMAND REGISTER PCI CONFIG. OFFSET 4H-5H (NB_COM)

This is the 16-bit PCI command register.

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

Bit 8 **SERR# enable.** If this bit is set to a ‘1’, the North Bridge may assert SERR# upon detecting a target abort in response to an North Bridge initiated PCI transaction, upon being forced to end a non-configuration space transaction with a master abort, or if a parity error on the PCI bus is detected. If this bit is set to ‘0’, the North Bridge will not assert SERR#.

Bit 7 **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 bit is hardwired to a ‘0’. Writes to it have no effect.

Bit 4 **Master Write and Invalidate Enable.** This bit is hardwired to a ‘0’. Writes to it have have no effect.

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

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

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

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

This register defaults to 0007h after reset.

10.5.4 NORTH BRIDGE PCI STATUS REGISTER PCI CONFIG. OFFSET 6H-7H (NB_STAT)

This is the 16-bit PCI Status register.

Bit 15 **Detected parity error.** This bit is set when a PCI parity error is detected. It may be cleared by software by writing a ‘1’ back to this bit.

Bit 14 **Signaled SERR#.** This bit is set to a ‘1’ when SERR# is asserted by the North Bridge. Writing a ‘1’ to this bit will clear it.

Bit 13 **Signaled Master Abort.** This bit is set to a 1 when the North Bridge terminates a PCI transaction with a master abort. Writing a ‘1’ to this bit will clear it.

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

Bit 11 **Signaled Target Abort.** This bit is hardwired to ‘0’.

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

Bit 8 **Data Parity Error Detected.** This bit is set to ‘1’ when a PCI data parity error is detected. Writing a ‘1’ will clear it.

Bit 7 **Fast Back-to-Back Capable.** Hardwired to ‘1’. Indicates that the North Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return ‘1’, writes have no effect.

Bits 6-0 *Reserved.* These bits are hardwired to ‘0’s.

This register defaults to 0280h after reset.

10.5.5 NORTH BRIDGE PCI REVISION ID REGISTER PCI CONFIG. OFFSET 8H (NB_R_ID)

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

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

10.5.6 NORTH BRIDGE DEVICE CLASS CODE REGISTER PCI CONFIG. OFFSET 9H-BH (NB_C_CODE)

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 (09h) **Programming Interface Register.** These bits are hardwired to 00h.

10.5.7 NORTH BRIDGE HEADER TYPE REGISTER PCI CONFIG. OFFSET EH (NB_HEAD)

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

10.5.8 NORTH BRIDGE PCI CONTROL REGISTER PCI CONFIG. OFFSET 50H (NB_CONT)

Bit 31-23 *Reserved.* Hardwired to ‘0’.

Bit 22 **PCI 2.0 Enable.** If this bit is set to ‘1’, North Bridge will be compatible with PCI 2.0 standard. If this bit is set to ‘0’, North Bridge is compatible with PCI 2.1 standard.

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

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

Bit 19-5 *Reserved.* Hardwired to ‘0’.

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

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

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

Bit 1 **SERR_ on PERR_ enable.**

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

10.5.9 NORTH BRIDGE PCI ERROR STATUS REGISTER PCI CONFIG. OFFSET 54H (NB_E_STAT)

Bit 31-5 *Reserved.* Hardwired to ‘0’.

Bit 4 **Read Data Parity Error Status.** This bit is set when a PCI read data parity error is detected. Writing a ‘1’ will clear it.

Bit 3 **Write Data Parity Error Status.** This bit is set when a PCI write data parity error is detected. Writing a ‘1’ will clear it.

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

Bit 1 **Parity Error Status.** System errors as a result of a parity error status. This bit is set to ‘1’ when SERR# was asserted as a result of parity error. Writing a ‘1’ will clear it.

Bit 0 **Received Target Abort Error.** System errors as a result of a received target abort. This bit is set to ‘1’ when SERR# was asserted as a result of recieving a target abort. Writing a ‘1’ will clear it.

10.6 THE SOUTH BRIDGE

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

The functions associated are listed below:

Bus = 0

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

Function = 0(South Bridge)

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

10.6.1 SOUTH BRIDGE PCI INTERFACE:

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

10.6.2 SOUTH BRIDGE PCI MASTER INTERFACE:

The ISA bus communicates with the PCI bus via the PCI master interface. This 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 grants is asserted, meaning that the PCI arbiter in the South Bridge is currently asserting a grant to ISA, then the master interface will begin the PCI cycle immediately. The ISA bus cannot generate PCI burst cycles.

10.6.3 PCI CYCLE TERMINATION:

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.

10.6.4 PCI TARGET INTERFACE:

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

10.6.5 PCI ADDRESS DECODE:

ISA resources are accessed only via subtractive decode.

10.6.6 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 South Bridge decodes and claims a PCI cycle and begins the process of forwarding it to the ISA controllers a counter is preloaded with the count of 8. This counter is decremented every PCI clock in which the ISA controller is not yet finished with the transaction. When the count reaches 0 a retry is signaled on the PCI bus, but the ISA controller continues to process the transaction. Eventually the 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.

10.7 SOUTH BRIDGE PCI FUNCTION 0 CONFIGURATION REGISTERS

Table 10.4- South Bridge Configuration Space Register Reset Values

31------------------------------24	23---------------------------16	15-----------------------------8	7------------------------------0	
Device ID: 02h	Device ID: 09h	Vendor ID: 10h	Vendor ID: 4Ah	00h
Status: 0280h		Command: 000Fh		04h
Base class code: 06h	Sub class code: 01h	Program. Inter. Reg. : 00h	Revision ID: 00h	08h
	Header: 80h			0Ch
				...
				...
			Miscellaneous reg : 00h	40h

This section describes the PCI to South Bridge control. The registers and reset values are illistrated in Table 10.4.

10.7.1 SOUTH BRIDGE VENDOR IDENTIFICATION REGISTER PCI CONFIG. OFFSET 0H-1H (SB_V_ID0)

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

Bits 15-0 These bits are hardwired to 104Ah.

Writes to this register have no effect.

10.7.2 SOUTH BRIDGE DEVICE IDENTIFICATION REGISTER PCI CONFIG. OFFSET 02H-03H (SB_D_ID0)

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

Bits 15-0 These bits are hardwired to 0209h

Writes to this register have no effect.

10.7.3 SOUTH BRIDGE PCI COMMAND REGISTER PCI CONFIG. OFFSET 4H-5H (SB_COM0)

This is the 16-bit PCI command register.

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

Bit 8 **SERR# enable.** If this bit is set to a ‘1’, the South Bridge may assert SERR# upon detecting a target abort in response to a South Bridge initiated PCI transaction on behalf of an ISA master. If this bit is set to ‘0’, the South Bridge will not assert SERR#.

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

Bit 6 **PERR# response.** Setting this bit to ‘1’ enables parity error detection.

Bit 5 **VGA Palette Snoop enable.** This bits is hardwired to a ‘0’. Writes to it have no effect.

Bit 4 **Master Write and Invalidate Enable.** This bit is hardwired to a ‘0’. Writes to it have no effect.

Bit 3 **Enable Special Cycles.** This bit is hardwired to a ‘1’. The South Bridge writes to it have no effect. The South Bridge responds to halt and shutdown cycles.

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

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

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

This register defaults to 000Fh after reset.

10.7.4 SOUTH BRIDGE PCI STATUS REGISTER PCI CONFIG. OFFSET 06H-07H (SB_STAT0)

This is the 16-bit PCI Status register.

Bit 15 *Reserved.* This bit is hardwired to ‘0’.

Bit 14 **Signaled SERR#.** This bit is set to a ‘1’ when SERR# is asserted by the South Bridge on behalf of an ISA master cycle. Writing a ‘1’ to this bit will clear it. SERR# is asserted in response to a target abort during an ISA master cycle on PCI bus and if bit-8 of the 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 the PCI transaction is initiated by the South Bridge on behalf of an ISA master is terminated with a target abort. Writing a ‘1’ to this bit will clear it.

Bit 11 **Signaled Target Abort.** This bit is set to a ‘1’ when the South Bridge terminates a PCI transaction with a target abort. Writing a ‘1’ to this bit will clear it. The South Bridge will generate target abort if a A1-0 of a PCI IO cycle does not match the Byte enables.

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

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

Bit 7 **Fast Back-to-Back Capable.** Hardwired to ‘1’. Indicates that the South Bridge, while acting as target, is capable of accepting fast back-to-back transactions. Reads will always return ‘1’, writes have no effect.

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

This register defaults to 0280h after reset.

10.7.5 SOUTH BRIDGE PCI REVISION ID REGISTER PCI CONFIG. OFFSET 08H (SB_R_ID0)

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

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

10.7.6 SOUTH BRIDGE DEVICE CLASS CODE REGISTER PCI CONFIG. OFFSET 09H-0BH (SB_C_CODE0)

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 are hardwired to 06h (Bridge Device).

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

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

10.7.7 SOUTH BRIDGE HEADER TYPE PCI CONFIG. OFFSET 0EH (SB_HEAD0)

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

10.7.8 SOUTH BRIDGE MISCELLANEOUS REGISTER PCI CONFIG. OFFSET 40H (SB_MISC0)

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

Bit 0 **PCI 2.0 Enable.** If this bit is set to ‘1’, South Bridge will be compatible with PCI 2.0 standard. If this bit is set to ‘0’, South Bridge is compatible with PCI 2.1 standard.

This register defaults to 00h after reset.

11 ISA INTERFACE

11.1 INTRODUCTION

The ISA/Local Bus Interface provides control of the high speed Local Bus, and gives access to the peripherals available in the STPC device and to Memory and external devices on the ISA bus.

Control of the ISA bus is by the East Bridge which acts as a bridge between the host CPU bus and the PCI bus. Reads and writes which are initiated by the CPU are subtractively decoded. Reads and writes that target 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 cache, 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 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 will create one or more ISA bus cycles.

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 10.1.3.

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.

11.2 PCI / ISA CYCLES

11.2.1 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.

11.2.2 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 Section 11.2.1).

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 register 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 21h. In this case, the West Bridge copy of the register is read, using a PCI read cycle.

11.2.3 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 50h.

11.2.4 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.

11.2.5 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.

11.2.6 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.

11.2.7 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 01h. 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.

11.3 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.

11.3.1 Real Time Clock Read and Write

The Real Time Clock (RTC) is connected to the XBUS. However the RTC is not connected to the command lines of the XBUS. Instead, four input pins of the RTC (CS#, AS, RW#, DS) are controlled directly by the 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 AS is directly connected to the RTCAS output. AS is the address strobe for the RTC, and it is asserted (high) during any IO write to port 70h.

The RTC input RW# is connected to the logical OR of the RTCRW# and ISAOE# outputs. RW# is write pulse for the RTC, and it will be asserted (low) during any IO write to port 71h.

The RTC input DS is connected to the logical 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 IRQ8 input. There is an internal inverter between the pin IRQ8 and the interrupt controller to maintain compatibility with the PC-AT without requiring additional external glue logic.

11.3.2 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 controller 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 during the I/O operation. This function is transparent to the software and no BIOS modification is required.

11.4 ISA STANDARD REGISTERS

Table 11-1. DMA 1 Registers

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

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.

11.4.1 DMA 1 registers

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

There are 16 DMA 1 registers. They are as shown in Table 11-1

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

Interrupt Controller 1 registers

Table 11-2. Interrupt Controller 1 registers

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

11.4.2 Interrupt controller 1

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

There are two Interrupt controller 1 registers. They are as shown in Table 11-2

Note that not all bits of the address are used.

11.4.3 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 11.4.4).

There are 4 Interval Timer registers. They are as shown in Table 11-3

Note that not all bits of the address are decoded.

Table 11-3. Interval Timer registers

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

11.4.4 Port B

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

Bit 7 **Parity Error.** This bit is set to a ‘1’ whenever a parity error is detected during system memory read operation. Once set, this bit can be cleared by setting bit 2 of this register to a ‘1’. Bit 2 should be reset to a ‘0’ to enable recording the next parity error. The parity error generates NMI to the host CPU if NMI is enabled.
This bit is read-only.

On the IBM PC-AT, parity error Flip/Flop latch (F/F) was clocked via MEMRD# and cleared asynchronously via bit 2 of this register. The output of the F/F is fed to the Port B without any synchronization.

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 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 **Parity Check Enable.** This bit is connected to the asynchronous clear input of the F/F which records the parity error. It must be set to a ‘1’ clear the F/F and then set to a ‘0’ to enable further parity errors. This bit is read/write and cleared to a ‘0’ by 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.

11.4.5 Port 60h and 64h

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

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

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

Fast host CPU reset only is generated by two methods:

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

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

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

11.4.6 Port 70h

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

11.4.7 Interrupt Controller 2 registers

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

Interrupt controller 2 occupies two register locations. They are as shown in Table 11-4.

Note that not all address bits are decoded.

:

Table 11-4. Interrupt Controller 2 registers

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

11.4.8 DMA Controller 2 registers

There are 16 DMA 2 registers. They are as shown in Table 11-5

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

.

Table 11-5. DMA Controller 2 registers

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

11.4.9 DMA Page registers

Table 11-6. DMA Page registers

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

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 11-6.

11.5 ISA CONFIGURATION REGISTERS

These registers are addressed through the Address Configuration Index (C.I.) and Data registers at IO addresses 22h and 23h respectively.

11.5.1 Miscellaneous Control Register 0 - C.I.50h

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.** If this bit is 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.

11.5.2 Miscellaneous Control register 1 - C.I.51h

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 cannot disable the write to the shadowed BIOS in DRAM since after shadow is enabled, all writes to the 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.

11.5.3 PIRQA Routing control register 0 - C. I.52h

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.

11.5.4 PIRQB Routing control register 0 - C.I. 53h

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.

11.5.5 PIRQC Routing control register 0 - C. I. 54h

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.

11.5.6 PIRQD Routing control register 0 - C.I. 55h

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.

11.5.7 Interrupt Level Control Register 0 - C. I. 56h

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 with ISA).

Bits 2-0 *Reserved.* Writes have no affect and the reads return undefined value.

This register defaults to 00h.

11.5.8 Interrupt Level Control Register 1C. I. 57h

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

Bits 7-6 **IRQ Control IRQ[15-14].** If set to a '1', the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible with ISA).

Bit 5 *Reserved.* Writes have no affect and the reads return undefined value.

Bits 4-1 **IRQ Control IRQ[12-9].** If set to a '1', the corresponding interrupt request is treated as a level input, otherwise it is treated as the edge sensitive input (compatible with 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.

11.5.9 IPC Configuration register - C.I. 01h

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 **EMR, 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 **CLK, 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.

11.5.10 ISA I/O port select and sync. - C.I. 59 (ISA_Sync)

Reg.

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 the PCI clock (i.e. Index 0x50h, bit 4 is set to 1). Setting this bit will result in small improvement in the ISA performance. This register also controls UART and Parallel Port base address selection.

Bits 7-6 **UART Base Address Selection**

Bit 7	Bit 6	UART selection
0	0	COM1 (IRQ4- Default)
0	1	COM2 (IRQ3- Default)
1	0	COM3 (IRQ4- Default)
1	1	COM4 (IRQ3- Default)

Bits 5-4 **UART1 Base Address Selection**

Bit 5	Bit 4	UART selection
0	0	COM1 (IRQ4- Default)
0	1	COM2 (IRQ3- Default)
1	0	COM3 (IRQ4- Default)
1	1	COM4 (IRQ3- Default)

Note: Both the UARTs cannot be configured to the same BASE Address (There is a hardware protection). If this is attempted the UART will automatically to COM1 and UART1 will default to COM2.

These bits are used to generate chip select UARTs.

If the internal UARTs are disabled then these chip selects also go out of the chip for external use.

Bits 3- 2 **Parallel Port Base Address Selection**

To redefine IRQ routing see Section 11.5.11

Bit 7	Bit 6	Parallel Port selection
0	0	LPT1 (IRQ7- Default)
0	1	LPT2 (IRQ5- Default)
1	0	LPT3 (IRQ7- Default)
1	1	LPT1 (IRQ7- Default)

Note: these bits are used to generate chip selects for the internal Parallel Port only. As the internal

Parallel Port has a built in address decoder.

Bit 1, *Reserved,* returns a value of ‘0’ when read.

Bit 0 **Synchronisation Enable.**
0 = Enabled
1 = Disabled

This register defaults to 00h at reset.

Programming notes

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

11.5.11 UART and Parallel Port IRQ Routing C.I. 91h (IRQ_Rout)

This register controls the selection of the default IRQ for the UART and Parallel Port. The register is reset after power on. So the UART’s are configured to use IRQ4 (COM1) and IRQ3 (COM2) also the Parallel Port is configured for the IRQ7 (LPT1).

Note: The routing is only effective if the internal UARTs and the Parallel Port are enabled.

Bit 7-3, *Reserved.*

Bit 2-1, **UART Bit Swap**, by setting bit 1 or bit 2 or both to 1 will swap the IRQ 3 and 4 for UART’s to COM1 and COM2 respectively. If bit 2 & 1 are set to 0, the IRQ’s will not be swapped.

Bit 0, **Parallel Port Bit Swap**, by setting this bit to 0 will force the Parallel Port IRQ on LPT1 to IRQ5, LPT2 to IRQ7 and LPT3 to IRQ5.

11.6 UPDATE HISTORY FOR ISA INTERFACE CHAPTER

The following changes have been made to the ISA Interface Chapter on 13/10/99.

Section	Change	Text
11.5.10	Replaced	UART Base Address Selection and UART1 Base Address Selection Bits 7-4
11.5.10	Replaced	Parallel Port Base Address Selection Bit 3-2
11.5.10	Added	Bit 1 as reserved
11.5.11	Added	Added Chapter 11.3.2

The following changes have been made to the ISA Interface Chapter on 13/10/99.

Section	Change	Text
11.5.10	Added	ISA I/O port select and sync. - C.I. 59 (ISA_Sync)

Update History for ISA interface chapter

12 VGA CONTROLLER

12.1 INTRODUCTION

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

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

12.2 VGA CONTROLLER

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

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

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

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

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

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

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

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

12.3 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.

12.4 GENERAL VGA REGISTERS

12.4.1 MOTHERBOARD ENABLE REGISTER MBEN 094H (RW)

Bits 7-6 *Reserved,* read as ‘0’s.

Bit 5 **Motherboard Enable.** If the VGA is configured to operate on the motherboard, then when this bit is set to ‘0’, it allows read and write access to port 102h. All other IO and memory accesses are ignored. When set to a ‘1’, this bit allows access to all IO and memory, but access to port 102h is ignored.

Bit 4 *Reserved,* reads as ‘0’.

Bit 3 **MBEN Video System Enable.** When ‘0’ this bit disables all IO and memory accesses to the VGA as well as the DAC registers. Accesses to 94h remain enabled. When ‘1’, Video system enable bits of port 0102h and 03C3h determine the accessibility of the VGA. The VGA continues to display video data while disabled.

Bits 2-0 *Reserved,* read as ‘0’s.

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

The contents of this register are 28h after reset.

Programming notes

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

12.4.2 ADD-IN VGA ENABLE REGISTER ADDEN 46E8H (RW)

Bits 7-5 *Reserved,* read as 0’s.

Bit 4 **Addin Enable.** If the VGA is configured to operate on an add-in card, then when this bit is set to ‘1’, it allows read and write access to port 102h. All other IO and memory accesses are ignored. When set to a ‘0’, this bit allows access to IO and memory, but access to port 102h is ignored.

Bit 3 **ADDEN Video System Enable.** When ‘0’ this bit disables all IO and memory accesses to the VGA as well as the DAC registers. Accesses to 46E8h remain enabled. When ‘1’, Video system enable bits of port 0102h determine the accessibility of the VGA. The VGA continues to display video data while disabled.

Bits 2-0 *Reserved,* read as ‘0’s.

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

This register is defined to be 00h after reset.

Programming notes

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

12.4.3 VIDEO SUBSYSTEM ENABLE 1 REGISTER VSE1 102H (RW)

Bits 7-1 *Reserved,* read as ‘0’s.

Bit 0 **VSE1 Video System Enable.** When ‘0’, this bit disables all IO and memory accesses to the VGA as well as DAC registers except port 102h.

Port 102h remains accessible to allow enabling of the VGA. Ports 46E8h and 94h are also not affected by this bit. The VGA continues to display video data while disabled.

This register is defined to be 00h after reset.

Programming notes

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

12.4.4 VIDEO SUBSYSTEM ENABLE 2 REG IO ADRESS 0X3C3H (RW)

Bits 7-1 *Reserved,* read as ‘0’s.

Bit 0 **VSE2 Video System Enable.** When ‘0’, this bit disables all IO and memory accesses to the VGA as well as DAC registers except ports 102h and 3C3h. Ports 102h and 03C3h remain 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.

This register is defined to be 00h after reset.

Programming notes

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

12.4.5 MISCELLANEOUS OUTPUT REGISTER MISC 3CCH/3C2H (R/W)

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

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

For older IBM compatible color monitors, the polarity of the vertical and horizontal retrace pulses was used to define the vertical scan rate, as follows in Table 12-1 :

Table 12-1Definition of IBM vertical & horizontal retrace pulses

Bit7	Bit6	Active Lines	Vertical Total
0	0	Reserved	Reserved
0	1	400 lines	414 lines
1	0	350 lines	362 lines
1	1	480 lines	496 lines

Bit 5 **Odd/Even Page Select.** This bit selects between two 64K pages of memory (of a 128K plane) when the VGA is in odd/even mode (replaces the least significant bit of the memory address). ‘0’ = low 64K page. ‘1’ = high 64K page. This bit is only effective in Mode 0, 1, 2, 3, or 7.

Bit 4 *Reserved*, reads as ‘0’.

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

Bit 1 **Enable RAM.** When ‘0’, this bit disables host accesses to the display RAM. The access to the ROM, however, remains enabled. Setting this bit to ‘1’ enables accesses to the display buffer.

Bit 0 **IO Address.** This bit defines the address map of the following registers ,seeTable 12-2 :

Table 12-2IO address map

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

This register is defined to be 00h after reset.

Programming notes

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

12.4.6 INPUT STATUS REGISTER #0 INP0 3C2H (R)

Bit 7 **Vertical Retrace Flag.** This bit is set at the beginning of the vertical retrace period if bit 4 of CR11 (Vertical Retrace End register) is set to one. Once set, this bit is cleared when bit 4 of CR11 is reset to 0. This recording of the vertical retrace interrupt is independent of bit 5 (disable vertical interrupt) of CR11.

See the description of CR11 for more details.

Bits 6-5 *Reserved.* These bits read as ones.

Bit 4 **RAMDAC Sense.** This bit is connected to the SENSE signal of the RAMDAC. It is used by the BIOS to auto-detect the monitor type.

Bits 3-0 *Reserved.* These bits read as zero.

12.4.7 INPUT STATUS REGISTER #1 INP1 3XAH (R)

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

Bits 5-4 **Diagnostic Use.** These bits reflect 2 of the 8 bit video output data during display periods and overscan color data during non-display periods. Selection of one of four pairs of bits is controlled by bits 5-4 of the AR12 as in Table 12-3 :

Table 12-3Input status register diagnostics

AR12		Diagnostic Bits	
Bit 5	**Bit 4**	**Bit 5**	**Bit 4**
0	0	Video2	Video0
0	1	Video5	Video4
1	0	Video3	Video1
1	1	Video7	Video6

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

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

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

12.5 VGA SEQUENCER REGISTERS

12.5.1 SEQUENCER INDEX REGISTER SRX 03C4H (RW)

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

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

The contents of this register are not defined after reset.

Programming notes

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

12.5.2 SEQUENCER RESET REGISTER SR0 03C5H INDEX 0 (RW)

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

Bit 1 **Synchronous Reset.** When set to '0' terminates display memory accesses. This bit, as well as bit 0 of this register, must be set to '1' to enable sequencer operations. The Clocking Mode register (SR1) bits 0 and 3, and Miscellaneous Output register bits 2-3 must not be changed unless this bit is set to '0' to avoid loss of memory contents.

Bit 0 **Asynchronous Reset.** This bit performs the same function as bit 1 except when set from '1' to '0', it also clears the Character Map select register (SR3) to '0'.

The contents of this register are not defined after reset.

Programming notes

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

12.5.3 SEQUENCER CLOCKING MODE REGISTER SR1 03C5H INDEX 1 (RW)

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

Bit 5 **Screen Off.** Setting this bit to '1' blanks the screen by driving black color (not overscan) on the screen. This facilitates the CPU to access video memory at maximum possible bandwidth.

Bit 4 **Shift4.** Along with Shift Load (bit 2) this bit controls the loading of the video serializers as in Table 12-4 :

Table 12-4Screen off video serializer loading

Bit4	Bit2	Video Serializer Load clock	Resolution
0	0	Every character	720 dots/line
0	1	Every second character	360 dots/line
1	X	Every fourth character	180 dots/line

Bit 3 **Dot Clk** When '0' sets the dot clock to be the same as the input dot clock. When '1', divides the input dot clock by 2 to derive the dot clock. The input dot clock is divided by 2 for 320 and 360 horizontal pixel modes 0, 1, 4, 5, D and 13. This is can not be used when using an external DCLK

Bit 2 **Shift Load,** see Bit 4 - Shift4.

Bit 1 *Reserved,* reads as ‘0’.

Bit 0 **8/9 Dot Clock.** When ‘0’, this bit causes the character clock to be 9 dots wide. When ‘1’, an 8-dot wide character clock is selected.

This register is defined to be 00h after reset.

Programming notes

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

12.5.4 SEQUENCER PLANE MASK REGISTER SR2 03C5H INDEX 2 (RW)

Bits 7-4 *Reserved,* read as ‘0’.

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

Bit 2 **Enable Plane 2.**

Bit 1 **Enable Plane 1**.

Bit 0 **Enable Plane 0.**

The planes are used in different manners by the various modes. These are shown in Table 12-5 :

Table 12-5Sequenncer Plane mask Modes

Mode	Plane 0	Plane 1	Plane 2	Plane3
Text Modes 0, 1, 2, 3, 7	Character Data	Attribute Data	Font Data	Unused
16-bit Color Graphics Modes D, E, 10, 12	Pixel Bit 0	Pixel Bit 1	Pixel Bit 2	Pixel Bit 3
4-Color Mono Graphics Mode F	Video	Ignored	Intensity	Ignored
4-Color Modes 4, 5	Even Byte	Odd Byte	Unused	Unused
2-Color Mono Graphics Mode 6	Even Byte	Odd Byte	Unused	Unused
2-Color Mono Graphics Mode 11	All Bytes	Unused	Unused	Unused
256-Color Graphics Mode 13	Byte 0	Byte 1	Byte 2	Byte 3

This register is defined to be 00h after reset.

Programming notes

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

12.5.5 SEQUENCER CHARACTER MAP REGISTER SR3 03C5H INDEX 3 (RW)

Bit 5 **Secondary Font Block Select bit 0**
Bit 4 **Primary Font Block Select bit 0**
Bit 3 **Secondary Font Block Select bit 2**
Bit 2 **Secondary Font Block Select bit 1**
Bit 1 **Primary Font Block Select bit 2**
Bit 0 **Primary Font Block Select bit 1**

This register is defined to be '0' after reset.

Programming notes

Used in text mode to select the primary and secondary font tables when the attribute bit 3 is '0' (for primary) or '1' (for secondary) as per the Table 12-6 *and* Table 12-7 *following :*

Table 12-6Sequencer Character Map Bit 3 = 0

Primary Font				
Bit1	**Bit0**	**Bit4**	**Font block #**	**Table Location**
0	0	0	0	0K
0	0	1	4	8 K
0	1	0	1	16 K
0	1	1	5	24 K
1	0	0	2	32 K
1	0	1	6	40 K
1	1	0	3	48 K
1	1	1	7	56 K

Table 12-7Sequencer Character Map Bit 3 = 1

Secondary Font				
Bit3	**Bit2**	**Bit5**	**Font block #**	**Table Location**
0	0	0	0	0 K
0	0	1	4	8 K
0	1	0	1	16 K
0	1	1	5	24 K
1	0	0	2	32 K
1	0	1	6	40 K
1	1	0	3	48 K
1	1	1	7	56 K

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

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

12.5.6 SEQUENCER MEMORY MODE REGISTER SR4 03C5H INDEX 4 (RW)

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

Bit 3 **Chain-4 Addressing.** When set to '1', this bit forces the two least significant host address bits to select the display buffer plane to be accessed by a host read or write. HA1-0 = '00' selects plane 0, HA1-0 = '01' selects plane 1, etc. For writes, the plane selected by the two address bits still must be enabled via the Plane Mask Register (SR2) in order for the writes to take place.

During read transfers, when this bit is set to '1', the Graphics Control Read Map register (GR4) is 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'.

This register is defined to be 04h after reset.

Programming notes

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

12.5.7 EXTENDED REGISTER LOCK/UNLOCK SR6 03C5H INDEX 6 (RW)

Bits 7-0 **Extended Registers Lock/Unlock.** When written to with 57h, all extended registers are unlocked. When written to with any value other than 57h, all extended registers are locked.

When the extended registers are in the locked state, reads to this register return a zero. When the extended registers are in the unlocked state, reads to this register return a '1'.

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

Programming notes

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

12.6 GRAPHICS CONTROLLER REGISTERS

12.6.1 GRAPHICS CONTROLLER INDEX REGISTER GRX 03CEH (RW)

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

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

The contents of this register are not defined after reset.

Programming notes

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

12.6.2 GRAPHICS SET/RESET REGISTER GR0 03CFH INDEX 0 (RW)

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

Bits 3-0 **Graphics Controller Set/Reset.** These bits define the value written to the four memory planes. In Write Mode 0, only the planes enabled by the Enable Set/Reset Register (GR1) are written to. In Write Mode 3, the contents of the Set/ Reset register are always enabled.
Bit 0 corresponds to memory plane 0, bit 1 to memory plane 1, etc.

The contents of this register are not defined after-reset.

Programming notes

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

12.6.3 GRAPHICS ENABLE SET/RESET REGISTER GR1 03CFH INDEX 1 (RW)

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

Bits 3-0 **Graphics Controller Enable Set/Reset.** These bits define which memory planes are to be written to with the value of the corresponding Set/ Reset Register (GR0) in Write Mode 0. In Write Mode 3, the Enable Set/Reset register has no affect.
Bit 0 corresponds to memory plane 0, bit 1 to memory plane 1, etc.

The contents of this register are not defined after reset.

Programming notes

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

12.6.4 GRAPHICS COLOR COMPARE REGISTER GR2 03CFH INDEX 2 (RW)

Bits 7-4 *Reserved,* read as ‘0’.

Bits 3-0 **Graphics Controller Color Compare Register.** These bits are compared with the 4-bit color of up to 8 pixels in Read Mode 1. The 8-bit (1-bit per pixel) result of the comparison is returned to the host. (A bit of ‘1’ is returned for a match, and ‘0’ for a non-match.) Only those bits enabled by the Color Don’t Care Register (GR7) are matched.

The contents of this register are not defined after reset.

Programming notes

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

12.6.5 RASTER OP/ROTATE COUNT REGISTER GR3 03CFH INDEX 3 (RW)

Bits 4-3 **Graphics Controller Raster Op.** These bits define the logical operation to apply to the Host data with the data in the Graphics Controller data latch. The possible values of this field are shown in Table 12-8

:

Table 12-8Graphics Controller Raster Op Field Values

Bit4	Bit3	Raster Operation
0	0	NOP - Host data passes through unmodifed
0	1	Logical AND of Host and latched data
1	0	Logical OR of Host and latched data
1	1	Logical XOR of Host and latched data

Bits 2-0 **Graphics Controller Rotate Count.** These bits specify the number of bits that the Host data is rotated before the Raster Op is applied. A count of 0 passes the data through unmodified, a count of 1 rotates the Host data 1 bit to the right.

The contents of this register are not defined after reset.

Programming notes

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

12.6.6 GRAPHICS READ MAP SELECT REGISTER GR4 03CFH INDEX 4 (RW)

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

Bits 1-0 **Graphics Controller Read Map Select.** These bits define the memory plane from which the CPU reads data in Read Mode 0.
A value of '00' selects plane 0, '01' selects plane 1, etc. This field also selects one of the 4 Bytes of the Graphics Control Read Data latches.

The contents of this register are not defined after reset.

Programming notes

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

12.6.7 GRAPHICS MODE REGISTER GR5 03CFH INDEX 5 (RW)

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

Bits 6-5 **Shift mode** These values are given in Table 12-9 .

Table 12-9Graphic Mode Shift Register Behaviour

Bit6	Bit5	Shift Register Behaviour
1	X	The shift registers are loaded in the manner to support 256 colors. This bit should be set to "1" for mode 13 operation.
0	1	2-bit packed pixel mode (modes 4 and 5) support. The data in the four serial shift registers are formatted as ATR0-3
0	0	Normal shift mode. M0d7-0, M1d7-0, M2d7-0 and M3d7-0 are shifted out with address to the Attributed Controller.

2-bit packed pixel modes:
ATR0:
M1d0 M1d2 M1d4 M1d6 M0d0 M0d2 M0d4 M0d6
ATR1:
M1d1 M1d3 M1d5 M1d7 M0d1 M0d3 M0d5 M0d7
ATR2:
M3d0 M3d2 M3d4 M3d6 M2d0 M2d2 M2d4 M2d6
ATR3:
M3d1 M3d3 M3d5 M3d7 M2d1 M2d3 M2d5 M2d7

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

Bit 3 **ReadMode.** If this bit is set to '0', a host read transfer returns the data Byte corresponding to the plane selected by the Read Map Select 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 in Table 12-10

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

Programming notes

Table 12-10Graphic Mode Write Behaviour

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

Where:

Write Mode 0: each of the four display memory planes are written with the host data rotated by the rotate count value specified in GR3.
If the Enable Set/Reset register (GR1) enables any of the four planes, the corresponding plane is written with the data stored in the Set/Reset register (GR0). The raster operation specified in GR3 and the bit mask register (GR8) contents alter data being written.

Write Mode 1: each of the four display memory planes are written with the data from the Graphics Controller read data latches. These latches should be loaded by the host via a previous read. The Raster Operation, Rotate Count, Set/Reset Data, Enable Set/Reset and Bit Mask registers have no effect.

Write Mode 2: memory planes 3-0 are filled with the value of the host data bits 3-0, respectively.

Data on the host bus is treated as the color value. The Bit Mask register (GR8) is effected in this mode. A “1” in a bit position in the Bit Mask register sets the corresponding pixel in the addressed Byte to the color specified by the host data bus. A “0” set the corresponding pixel in the addressed Byte to the corresponding pixel in the Graphics Controller read latches. The Set/Reset, Enable Set/Reset and Rotate Count register have no effect.

Write Mode 3: each of the four video memory planes is written with 8-bits of the color value contained in the Set/Reset register for that plane. The Enable Set/Reset register as no effect, all bits are enabled. The host data is rotated and ANDed with the Bit Mask register to form an 8-bit value that performs the same function as the Bit Mask register in Write Modes 0 and 2. This write mode can be used to fill an area with a single color or pattern.

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

12.6.8 GRAPHICS MISCELLANEOUS REGISTER GR6 03CFH INDEX 6 (RW)

Bits 7-4 *Reserved,* read as ‘0’.

Bits 3-2 **MemoryMap.** These bits specify the map of the display memory buffers in the CPU address space. They are defined as follows in Table 12-11

Table 12-11Graphics Miscellenous Memory Map

Bit3	Bit2	Address Map
0	0	A0000h to BFFFFh (128K)
0	1	A0000h to BFFFFh (128K)
1	0	B0000h to B7FFFh (32K)
1	1	B8000h to BFFFFh (32K)

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

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

The contents of this register are not defined after reset.

Programming notes

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

12.6.9 GRAPHICS COLOR DON’T CARE REGISTER GR7 03CFH INDEX 7 (RW)

Bits 7-4 *Reserved,* read as ‘0’.

Bits 3-0 **Dont_care Color Plane Selects.** One bit per plane determine whether the corresponding color plane becomes a don’t care when a CPU read from the video memory is done in Read Mode 1. A ‘1’ makes the corresponding plane a don’t care plane.

The contents of this register are not defined after reset.

Programming notes

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

12.6.10 GRAPHICS BIT MASK REGISTER GR8 03CFH INDEX 8 (RW)

Bits 7-0 **Bit Mask.** Any bit programmed to a ‘0’ in-this register will cause the corresponding bit in each of the four memory planes to be left unchanged by all operations. The data written into memory in this case will be the data which was read in the previous read operation and stored in the Graphics Controller’s read latch.
The bit mask is applicable to any data written by the host. The bit mask applies to all four planes simultaneously.

The contents of this register are not defined after reset.

Programming notes

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

12.7 ATTRIBUTE CONTROLLER REGISTERS

12.7.1 ATTRIBUTE CONTROLLER INDEX ARX 3C0H (RW)

The Attribute Controller Index register is used to index into the Attribute Data register array.

Port 3C0h is used for write access to both this index register and, in a subsequent write to this address, to the data register pointed to by the index. There is a flipflop which changes state after each write to this port. The state of the flipflop determines whether the next IO write to 3C0h will be to the index register or to a data register. The flipflop may be initialized - to point to the index register - by performing a read from Input Status Register #1 (IO address 3XAh).

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

Bit 5. **Palette Address Source.** When set to zero, allows host write access to the Attribute Palette Registers. The CRT display is turned off while this bit stays zero and overscan color is displayed. Setting this bit to one allows normal video pixel display and disables host write access to the Palette registers.

Bits 4-0 **Attribute Controller Index.** Points to the data register which will be accessed by the next write to port 3C0h or the next read from port 3C1h. A sample program could be as follows;

```
mov DX, 3DAh
inAL, DX
movAL, Index
movDX, 3C0h
outDX, AL
movAL, Data
outDX, AL
```

12.7.2 ATTRIBUTE PALETTE REGISTERS AR0-ARF 3C1H/3C0H (R/W)

These sixteen registers provide one level of indirection between the color data stored in the display frame buffer and the displayed color on the CRT screen. In all modes except 256 color mode, the (maximum) 4-bit raw color values select one of these sixteen Palette registers. The six bit output of the Palette registers is combined with bits 3-2 of AR14 to form the 8-bit output of the VGA controller. In addition, bits 5-4 of the VGA output may come from either Palette register bits 5-4 or from AR14 bits 1-0 depending on the state of the V54 bit (bit 7) of register AR10.

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

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

12.7.3 ATTRIBUTE CTRL MODE REGISTER AR10 3C1H/3C0H (R/W)

Bit 7 **V54 Select.** This bit determines whether bits 5-4 of the VGA pixel output come from the Video54 field of AR14 (bits 1-0) or from the normal output of the VGA Palette registers. Setting this bit to one selects the Video54 field.

Bit 6 **Pixel Width.** When this bit is set to one, pixels are clocked at half the normal rate. The effect is to double the width of pixels displayed on the CRT.

Bit 5 **Pixel Panning Compatibility.** When VGA split screen is in effect, this bit controls whether both screens or just the top one are affected by Pixel and Byte Panning fields. When set to zero, both screens pan together.

Bit 4 *Reserved.*

Bit 3 **Blink Enable.** Setting this to one enables blinking in both text and graphics modes. When this bit is set to one in text mode, character attribute bit 7 is used on a character by character basis to enable or disable blinking. When this bit is set to zero in text mode, character attribute bit 7 controls character intensity. The blinking rate is equal to the vertical retrace rate divided by 32 (about twice per second).

Setting this bit to one in graphics modes causes the VGA palette input bit 3 to toggle (approx twice per second) if the incoming pixel bit 3 is high.

Bit 2 **Line Graphics Enable.** Setting this bit to one forces the ninth pixel of a line graphics character (ascii codes C0h through DFh) to be the same as the eighth pixel. Setting it to zero forces the ninth pixel to be displayed as the background color. Ninth pixels of all other ascii codes are always displayed as background color.
This bit has no meaning when character width is not set to nine or during graphics modes.

Bit 1 **Mono Attributes Enable.** Setting this bit to one in graphics modes while Bit 3 of this register is also one causes the VGA palette input bit 3 to toggle regardless of the incoming pixel's bit 3.

Bit 0 **Graphics Mode.** Set this bit to one for graphics mode, zero for text mode.

All the bits in this register reset to zero.

12.7.4 ATTRIBUTE CTRL OVERSCAN COLOR AR11 3C1H/3C0H (R/W)

Bits 7-0 **Border Color.** These bits define the color of the CRT border if there is one. The border or overscan region is that part of the display between where active pixels are displayed and those where the blank signal is active.

12.7.5 ATTRIBUTE COLOR PLANE ENABLE AR12 3C1H/3C0H (R/W)

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

Bits 5-4 **Video Status Mux Control.** These bits select two of the eight output bits of the Attribute Controller to be read via Input Status Register #1 (3XAh) bits 5-4. The selection is as follows in Table 12-12 :

Table 12-12Video Status Mux Control Values

AR12[5]	AR12[4]	ISR[5]	ISR[4]
0	0	PD[2]	PD[0]
0	1	PD[5]	PD[4]
1	0	PD[3]	PD[1]
1	1	PD[7]	PD[6]

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

12.7.6 ATTRIBUTE HORIZONTAL PIXEL PANNING AR13 3C1H/3C0H (R/W)

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

Bits 3-0 **Horizontal Pixel Panning.** These bits specify the number of pixels by which to shift the display left. see Table 12-13

Table 12-13Horizontal Pixel Panning values

H Pixel Pan	Shift		
	9 pixels/chr	8 pixels/chr	mode 13
0	1 pixel left	0 pixels	0 pixels
1	2 pixels left	1 pixel left	0 pixels
2	3 pixels left	2 pixels left	1 pixel left
3	4 pixels left	3 pixels left	1 pixel left
4	5 pixels left	4 pixels left	2 pixels left
5	6 pixels left	5 pixels left	2 pixels left
6	7 pixels left	6 pixels left	3 pixels left
7	8 pixels left	7 pixels left	3 pixels left
8-15	0 pixels	undefined	undefined

The bits in this register reset to zero.

12.7.7 ATTRIBUTE COLOR SELECT REGISTER AR14 3C1H/3C0H (R/W)

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

Bits 3-2 **Video76.** In all modes except 256 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.

12.8 CRT CONTROLLER REGISTERS

The STPC implements an extension of the VGA CRTC controller. The CRTC controller supports up to 1024x768 display resolutions at 75HZ refresh rates as defined by VESA Monitor Timing Standard. The horizontal timing control fields are all VGA compatible. The vertical timings are extended by 1-bit to accommodate above display resolution. The address registers are extended to allow locating the frame buffer in anywhere within the first 4Mb of physical main memory.

12.8.1 INDEX REGISTER CRX 3X4H (RW)

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

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

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

This register resets to zero.

12.8.2 HORIZONTAL TOTAL REGISTER CR0 3X5H INDEX 0 (RW)

The horizontal total register defines the total number of characters in a horizontal scan line, including the retrace time. The characters displayed on the screen are counted by a character counter. A count of 0 corresponds to the first displayed character at the left side of the screen. The value of the character counter is compared with the value in this register to provide the horizontal timing. A character is composed of 8 or 9 pixels as defined in Sequencer clocking mode register. All horizontal and vertical timing is based on the contents of this register.

The maximum horizontal resolution possible with this field is approximately 260*8*0.8 = 1664. (260 is 255+5, 0.8 is the fraction of a horizontal scan period during which active pixels are displayed.)

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

This register resets to zero.

12.8.3 HORIZONTAL DISPLAY END REGISTER CR1 3X5H INDEX 1 (RW)

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

This register is in unknown state after reset.

12.8.4 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.

12.8.5 HORIZONTAL BLANKING END REGISTER CR3 3X5H INDEX 3 (RW)

Bit 7 *Reserved.* This readable and writable bit must be written to as ‘1’ to ensure proper VGA 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.

Programming notes

This field controls the width of the horizontal blanking signal as follows:
Horizontal Blanking start register + width of the blanking signal = 6-bit Horizontal blanking end value.

The blanking signal set in CR2 and CR3 should start at least 23 GCLKs prior to the start of video window.

12.8.6 HORIZONTAL RETRACE START REGISTER CR4 3X5H INDEX 4 (RW)

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

This register is in unknown state after reset.

12.8.7 HORIZONTAL RETRACE END REGISTER CR5 3X5H INDEX 5 (RW)

Bit 7 **Horizontal Blanking End Value Bit 6.** This is the sixth bit of the Horizontal Blanking end field. Refer to CRTC Horizontal Blanking end register for more details.

Bits 6-5 **Horizontal Retrace Skew Control.** This field delays the start of the horizontal sync by the specified number of character clocks. For text mode operation, this field should be programmed to ‘1’.

Bits 4-0 **Horizontal Retrace Width Value.** These 5-bits specify the width of the horizontal sync signal as follows:

Horizontal Retrace Start register + width of the horizontal sync = 5-bit Horizontal retrace end value.

This register is in unknown state after reset.

12.8.8 VERTICAL TOTAL REGISTER CR6 3X5H INDEX 6 (RW)

This register contains the least significant 8-bits of the 11-bit wide Vertical Total value. Next most significant 2-bits are located in CRTC overflow register CR7 and the 11th bit is located in Repaint Control Register 4.

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

12.8.9 OVERFLOW REGISTER CR7 3X5H INDEX 7 (RW)

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

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

end register.

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

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

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

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

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

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

This register is in unknown state after reset.

12.8.10 SCREEN A PRESET ROW SCAN REGISTER CR8 3X5H INDEX 8 (RW)

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

Bits 6-5 **Display Shift.** This field is added to the memory address generated during the display. As a result, the display shifts left by one, two or three Bytes. For both alphanumeric and graphics modes, this implies a left shift by 8, 16 or 24 pixels respectively. This field is encoded as follows in Table 12-14 :

Table 12-14Display shift encoding values

Bit-6	Bit-5	Byte Panning
0	0	0 Byte (display shifts 0 pixels left)
0	1	1 Byte (display shifts 8 pixels left)
1	0	2 Bytes (display shifts 16 pixels left)
1	1	3 Bytes (display shifts 24 pixels left)

When the line compare condition becomes true and pixel panning compatibility bit (AR10 bit 5) is a ‘1’, the outputs of bits 5 and 6 are forced ‘0’ until the start of the next vertical sync pulse.

Bits 4-0 **Smooth Scroll.** This field can be used to implement smooth vertical scrolling. It specifies the starting row scan count of the character cell after a vertical retrace (assuming the scan lines of a character row are numbered starting with 0). Smooth vertical scrolling can be implemented by setting this register to a value between 1 and the value in CR9. As a result, after a vertical retrace, the display will start from the scan line specified in this field instead of 0.

This field is effective only for the top half of the screen (Screen A) if split screen mode is in effect.

Each horizontal scan increments the horizontal row scan counter and is reset to 0 when it reaches the character cell height value programmed in CR9. If this field is programmed to a value 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.

12.8.11 CHARACTER CELL HEIGHT REGISTER CR9 3X5H INDEX 9 (RW)

Bit 7 **Scan Double.** When set to a ‘1’, this bit allows a 200-line mode to be displayed on 400 display scan lines by dividing the row scan counter clock by 2 to duplicate each scan line. Thus all row scan address counter based timing (including character height and cursor and underline locations) double, as measured in scan lines, when scan doubling is enabled.
Scan doubling only effects the way in which data is displayed; it does not effect display timing. If this bit is set without changing anything else, data currently displayed will appear twice as tall; horizontal and vertical sync, blanking etc., will remain the same.

Bit 6 **Bit-9 of the 11 bit wide Line Compare field.**

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

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

This register is unknown state after reset.

12.8.12 CURSOR START REGISTER CRA 3X5H INDEX A (RW)

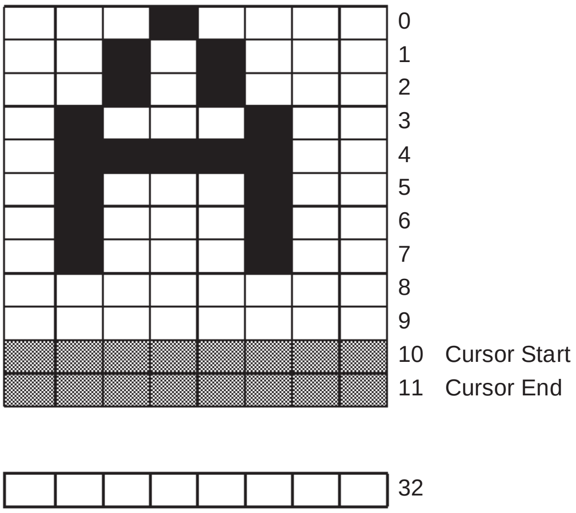
Bits 7-6 *Reserved.* Must be written to ‘0’s. Read back is undefined.

Bit 5 **Cursor Display Enable.** When set to a ‘0’, this bit enables displaying the cursor. Cursor is displayed only in alphanumeric mode. In graphics mode, the cursor is always disabled and this bit has no effect.

Bits 4-0 **Cursor Start Scan Line.** This field, in conjunction with the Cursor End scan line, defines the shape of the cursor. The hardware cursor is represented as a block of pixels occupying a character position. This field determines the first scan line within the character box that should be filled in (the first scan line is numbered as 0). If the cursor start and end scan line numbers are the same, one scan line wide cursor will be displayed. If starting scan line number is larger than the end, no cursor will be displayed. This is illustrated in Cursor start and end scan lineFigure 12-1

This register is in unknown state after reset.

Figure 12-1Cursor start and end scan line



12.8.13 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. Ths is illustrated in Figure 12-1.

This register is in unknown state after reset.

Figure 12-1 illustrates the cursor start and end registers

12.8.14 START ADDRESS HIGH REGISTER CRC 3X5H INDEX C (RW)

This 8-bit register specifies bits 15-8 of the 20-bit display buffer address which will be displayed on the screen after a vertical retrace. Register 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).

12.8.15 START ADDRESS LOW REGISTER CRD 3X5H INDEX D (RW)

This 8-bit register specifies bits 7-0 of the 20-bit display buffer address which will be displayed on the screen after a vertical retrace. Register CRC contains bits 15-8 and the CRTC extended register CR19 contains the upper 4 bits.

The start address in the CRC, CRD and CRIG does not define the offset within the frame buffer of the 1st displayed pixel but this value divided by 4.

12.8.16 TEXT CURSOR OFFSET HIGH REGISTER CRE 3X5H INDEX E (RW)

Bits 7-0 **Cursor offset bits 15-8.** This field contains the upper half of the 16-bit cursor offset. The offset is relative to the left-most character on the top of the screen and is specified in terms of character positions. For example, an offset of 0 will place the cursor on the left-most character on the top. An offset of 2 will place the cursor at the third character from the left in the top most row and so on.

Since the information is stored in the display memory as character-attribute pairs, the address of the character under the cursor will be exactly twice the cursor offset + the screen base address.

This register is undefined after reset.

12.8.17 TEXT CURSOR OFFSET LOW REGISTER CRF 3X5H INDEX F (RW)

This register is the VGA compatible Cursor Offset Low register.

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

This register is undefined after reset.

12.8.18 VERTICAL RETRACE START REGISTER CR10 3X5H INDEX 10 (RW)

This register contains the lower 8 bits of the 11-bit wide vertical retrace start value. Register CR7 contains bits 8 and 9. Repaint Control Register 4 contains the msb of this 11-bit field. The retrace value is specified in horizontal scan lines where top most scan line on the screen is line 0.

12.8.19 VERTICAL RETRACE END REGISTER CR11 3X5H INDEX 11 (RW)

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

Bit 6 *Reserved.* This bit is both readable and 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 the width of the vertical retrace output as follows:

Value in the Vertical Retrace Start register (CR10) + Width of the vertical retrace pulse = 4-bit value to be programmed into this field.

This register defaults to binary 0x10xxxx after reset.

12.8.20 VERTICAL DISPLAY END REGISTER CR12 3X5H INDEX 12 (RW)

This register contains the lower 8-bits of the 11-bit wide Vertical display end value which specifies the scan line position where the display on the screen ends. Bits 8 and 9 are specified in CRTC overflow register and the bit-10 in the Repaint Control Register 4.

Programming notes

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

12.8.21 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.

12.8.22 UNDERLINE LOCATION REGISTER CR14 3X5H INDEX 14 (RW)

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

Bit 6 **Double Word Mode.** If this bit is set to a ‘1’, the address generated by the CRTC memory address counter is shifted up two bits to provide the frame buffer address and bits 1-0 of the frame buffer address are driven from CRTC memory address counter bits 13 and 12 respectively. The logical screen width is multiplied by 8 and added to the starting address of the current scan line to compute the starting address of the next scan line.

Bit 5 **Count by 4.** Setting this bit to one causes the memory address counter to increment every four character clocks.

Bits 4-0 **Underline Location.** This field specifies the horizontal row scan of the character cell at which the underlining will occur assuming that top line of the character cell is numbered 0. Underlining occurs in text (alphanumeric) modes only when an attribute value of ‘b000i001’ binary is detected (where b indicates blink and i indicates intensified).

Bit 0 **Underline Enable.** Setting this bit to ‘1’ enables underlining.

This register is undefined after reset.

Programming notes

Underlining is enabled only in alphanumeric mode and then it is meaningful only for monochrome display (mode 7). For color modes the bit 0 of the attribute Byte is interpreted as foreground color. There is no explicit bit to disable underlining for color alphanumeric modes. Instead, it is disabled by programming this field to a value larger that the character cell height programmed in CR9.

12.8.23 VERTICAL BLANKING START REG CR15 3X5H INDEX 15 (RW)

This register contains the lower 8-bits of the 11-bit scan line valued where the vertical blanking is to begin. The 9th and 10th bits are located in CR7 and CR9 and the 11th bit is located in Repaint Control Register 4.

This register is undefined after reset.

12.8.24 VERTICAL BLANKING END REGISTER CR16 3X5H INDEX 16 (RW)

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

This register is undefined after reset.

Programming notes

Start Vertical Blank value + width of the blanking pulse = Value programmed in this register.

While the register is 8-bits wide, and all bits are readable/writeable, only the least significant 7 bits are used in the generation of the vertical pulse.

12.8.25 MODE REGISTER CR17 3X5H INDEX 17 (RW)

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

Bit 6 **Byte/Word# Mode.** A ‘1’ value in this bit position selects the Byte mode and a ‘0’ selects the word mode. Following Table 12-15 lists the memory address generation for Byte, word and double-word addressing modes. iA24-0 refer to the output of the internal memory address counter and the memory address is the address presented to the address lines of a 4-Byte wide display buffer memory, that is, each memory address selects 4-Bytes.

The least significant memory address bit in Word mode is selected between iA13 and iA15 based on bit 5 of this register.

Table 12-15Memory addressing for Byte, Word and Dword Generation

Internal Memory Address counter	Byte Mode	Word Mode	Double Word Mode
iA24	iA24	iA23	iA22
iA23	iA23	iA22	iA21
:	:	:	:
:	:	:	:
iA3	iA3	iA2	iA1
iA2	iA2	iA1	iA0
iA1	iA1	iA0	iA13
iA0	iA0	iA13/iA15	iA12

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.

12.8.26 LINE COMPARE REGISTER CR18 3X5H INDEX 18 (RW)

This register contains the lower 8 bits of the 10-bit wide Line Compare field. The 9th and 10th bits of this field are held in CR7 and CR9 registers respectively. When the horizontal scan line counter value is equal to the contents of the Line Compare register, the memory address generator and the character row scan count are cleared. As a result the display is split into the two halves. The top half, Screen A displays the contents of the display buffer starting from Start address (CRC and CRD registers) while the bottom half, Screen B, displays the contents of the display buffer starting from address 0.

Screen A can be smooth scrolled vertically but Screen B can not. Control is provided via bit 5 of AR10 register to allow Screen B to pan 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.

12.8.27 GRAPHICS CONTROL DATA CR22 3X5H INDEX 22 (R)

Bits 7-0 **Graphics Controller Data Latch N.** These bits, when read, provide the state of one of the 4 Graphics Controller’s Data Latches. The Graphics Controller Read Map Select register (GR4) specify which latch is read.

The contents of this register is not defined after reset.

Programming notes

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

12.8.28 ATTRIBUTE ADDRESS FLIPFLOP CR24 3X5H INDEX 24 (R)

Bit 7 **Attribute Flipflop.** This read-only bit indicates the state of the Attribute Controller index flipflop. When this bit is zero, the next access to IO port 3C0h will be to the Attribute Index register. When this bit is one, the next access will be to an Attribute data register.

Bits 6-0 *Reserved.* Read as zero.

12.8.29 ATTRIBUTE INDEX READBACK CR26 3X5H INDEX 26 (R)

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

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

Bits 4-0 **Attribute Controller Index.** This is a read-only copy of bits 4-0 of the Attribute Controller Index register.

12.9 VGA EXTENDED REGISTERS

The following registers are additions to those found in the standard VGA specification. They can only be accessed after register SR6 has been written to with 57h.

A typical sequence in 80X86 assembly could be;

```
maxDX, 3C4h
movOX, 5706h
outDX, AX
```

12.9.1 REPAINT CONTROL REGISTER 0 CR19 3X5H INDEX 19 (RW)

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

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

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

This register defaults to 00h after reset.

12.9.2 REPAINT CONTROL REGISTER 1 CR1A 3X5H INDEX 1A (RW)

Bit 7 **Hsync Toggle Disable.** When set to a ‘1’, this bit forces the Hsync to inactive state (high or low as programmed in the bit 6 of the Miscellaneous output register). See note below.

Bit 6 **Vsync Toggle Disable.** When set to a ‘1’, this bit forces the Vsync to inactive state (high or low as programmed in the bit 7 of the Miscellaneous output register). See note below.

Bit 5 *Reserved.* This bit always reads as a one.

Bit 4 **Compatible Text Mode.** When this bit is set to one, the CRT controller expects the font data format within the frame buffer to be identical to that used by the standard VGA chip. Setting this bit to zero enables “Enhanced Text Mode” as described under CR1C bit 7.
Note, though that this bit has the opposite sense of CR1C bit 7.

Bit 3 *Reserved.* This bit always reads as a one.

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

Bit 1 **Six Bit Palette.** Set this bit to one to enable VGA compatible 6 bit palette functionality and zero to enable the 8 bit palette.

Bit 0 **VGA Address Wrap.** Setting this bit to one enables VGA compatible address wrapping such that the CRTC will only address 256kb of frame buffer memory (address bits 16 and above are 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.

Programming notes

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

12.9.3 REPAINT CONTROL REGISTER 2 CR1B 3X5H INDEX 1B (RW)

Bits 7-3 **FIFO Low Water Mark.** When the FIFO occupancy falls below twice this value, the CRTC will restart frame buffer read cycles to refill the FIFO. This field should only ever be set by the BIOS during mode switches.
Setting this field too small results in random pixels being displayed to the screen; setting it too large results in decreased CPU - DRAM bandwidth. 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.

12.9.4 REPAINT CONTROL REGISTER 3 CR1C 3X5H INDEX 1C (RW)

Bit 7 **Enhanced Text Font Load.** This bit should be set to one prior to loading fonts for 132 column high speed text mode. It warps the frame buffer addresses such that what appear to be standard text font writes actually get stored into frame buffer plane 2 in a more optimized manner. Specifically, frame buffer address bits 15-5 are swapped down to become bits 10-0 and bits 4-0 are moved up to become bits 15-11.
The sequence of events to load 132 column enhanced text fonts is as follows. First, Odd-Even and Chain 4 modes should be turned off, then this bit should be set to one. Fonts should then be loaded in the normal VGA manner and finally this bit should be reset to zero and Text/Odd-Even mode entered.

Note that the corresponding address warp for the CRT Controller is performed during font table look-ups when bit 4 of CR1A is set to zero.

Bits 6-5 *Reserved,* read as ‘0’s.

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

Bit 2 **Sequential Chain-4.** When this bit is set to ‘1’, allows the display buffer memory to appear as a normal memory with a Byte address in the host address space mapping into a Byte address in the display buffer address space. Chain-4 in SR4 must be set for Sequential Chain-4 to work

Bit 1 **Page Select Control.** This bit provides control over whether the cycle type (read or write) or the upper address bit controls the page selection. The VGA implements two 7-bit page registers, Page 0 and Page 1, to allow mapping the VGA address space anywhere in the 4 MB address space.

If this bit is ‘1’, the page selection is based on bit 15 or 16 of host address and the Memory Map bits of GR6 as follows in Table 12-16

Table 12-16Page Select Control values

GR6			Page Selection
Bit3	Bit2	Size	

0	0	128K	HA16=0 → Page 0;
			HA16=1 → Page 1

0	1	64K	HA15=0 → Page 0;
			HA15=1 → Page 1

1	0	32K	Not allowed
1	1	32K	Not allowed

Bit 0 **Enable Overlapped Paging.** This bit should be turned on to solve the broken line problem. When software wants to draw a line that crosses the current page boundary it turns this bit on to form a page out of half of the current page and half of the next page. Since the hardware adds half page to the address when this bit is on, the software should subtract half page for passing on the address.

When this bit is ‘1’, the memory address bits MA17 and MA18 are changed as follows for Normal, Odd/Even and Chain-4 cases, see Table 12-17 :

Table 12-17Enable Overlapped Page Bit MA18 & MA17

GR6			Added to MA18	Added to MA17
Bit3	Bit2	Size		
0	0	128K	1 (256K added)	0
0	1	64K	0	1 (128K added)
1	0	32K	Not allowed	
1	1	32K	Not allowed	

for Sequential Chain-4, MA16 and MA15 are changed as follows in Table 12-18 :

Table 12-18Enable Overlapped Page Bit MA16 & MA15

GR6			Added to MA16	Added to MA15
Bit3	Bit2	Size		
0	0	128K	1 (64K added)	0
0	1	64K	0	1 (32K added)
1	0	32K	Not allowed	
1	1	32K	Not allowed	

This register is defined to be 00h after reset.

Programming notes

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

12.9.5 PAGE REGISTER 0 CR1D 3X5H INDEX 1D (RW)

Bit 7 *Reserved,* read as ‘0’.

Bits 6-0 **Page 0 Bits 6-0.** 7-bit Page register 0 is used to extend the host address to allow the VGA buffer to be located anywhere in the 4 MB frame buffer space. The pages are located on 32K boundaries for Normal, Odd/Even and Chain-4 modes and on 8K boundaries for Sequential Chain-4 mode. This is illustrated in Table 12-2

Programming notes

Register = A000h mapped to the frame buffer and can be either 8 KBytes or 32 KBytes

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

The contents of this register is not defined after reset.

12.9.6 PAGE REGISTER 1 CR1E 3X5H INDEX 1E (RW)

Bit 7 *Reserved,* read as ‘0’.

Bits 6-0 **Page 1 Bits 6-0.** 7-bit Page register 1 is used to extend the host address to allow the VGA to be located anywhere in the 4 MB frame buffer space. The pages are located on 32K boundaries for Normal, Odd/Even and Chain-4 modes and on 8K boundaries for Sequential Chain-4 mode. This is in Table 12-2

Register = A800h mapped to the frame buffer and can be either 8 KBytes or 32 KBytes

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

The contents of this register is not defined after reset.

12.9.7 GRAPHICS EXTENDED ENABLE REGISTER CR1F 3X5H, INDEX 1F (RW)

Bit 7 **Exen.** Writing a ‘1’ in this bit enables the GE extended functionality and also direct access to the frame buffer as defined by GBASE (CR20). Writing a ‘0’ disables it. After reset, this bit is set to ‘0’.

Bits 6-0 *Reserved,* read as ‘0’s

This register is defined to be 00h after reset.

Programming notes

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

12.9.8 GRAPHICS EXTENDED GBASE REGISTER CR20 3X5H, INDEX 20 (RW)

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

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

This register is defined to be 00h after reset.

Programming notes

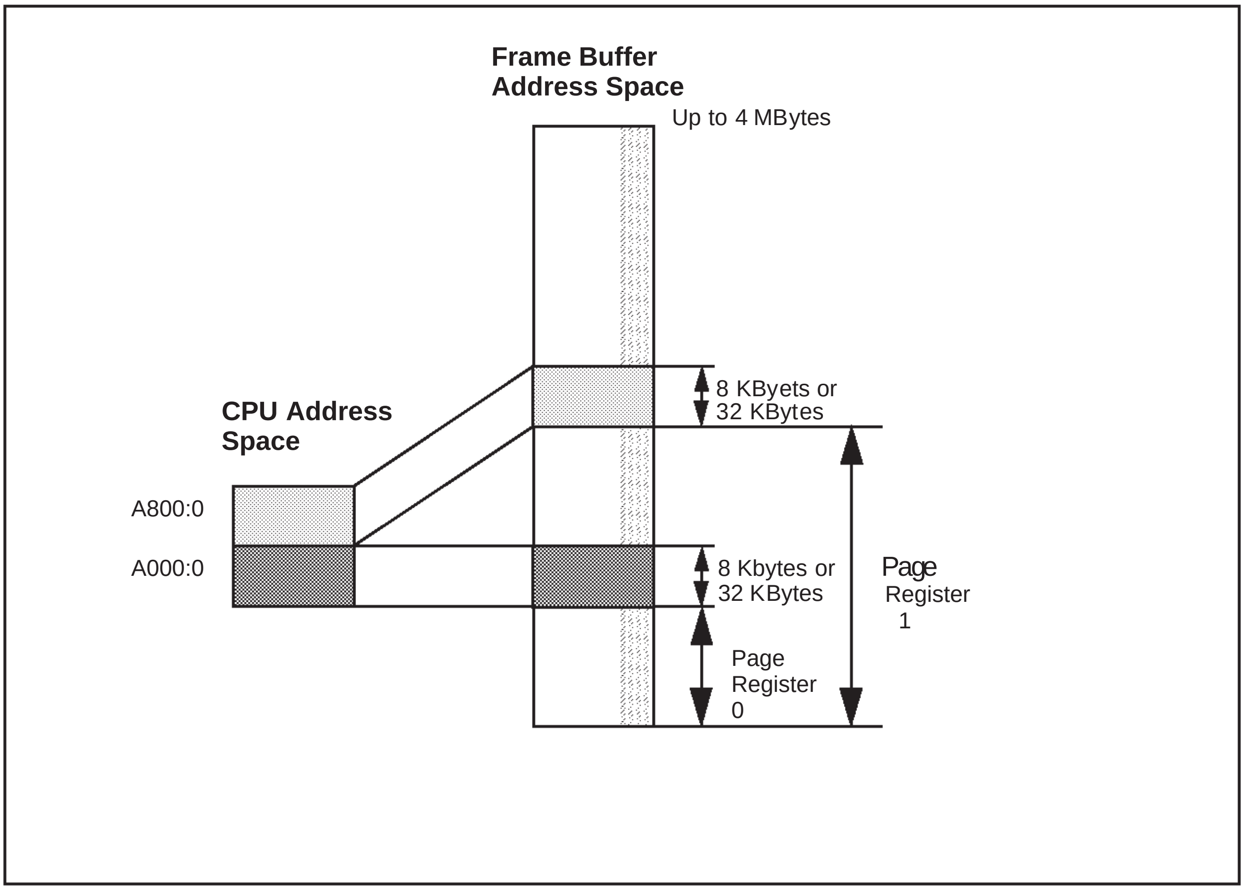
The contents of this register are not altered by drawing operations.

12.9.9 GRAPHICS EXTENDED APERTURE REGISTER CR21 3X5H, INDEX 21 (RW)

Bits 7-0 **Aperture.** The lower 6 address bits are prepended to the 16 least significant address bits in A0000h addresses to form a 22-bit address. This address is then used to map into the 4 MBytes Extended GE Register space. To use this feature, bits 7-6 must be set to ‘01’. Setting this register to FFh disables the aperture. Other values of this register can cause undefined results.

The purpose of the aperure is to enable access to the extended memory mapped register in real mode.

Figure 12-2. Illustration of Page Register 0 and Page Register 1



This register is defined to be FFh after reset.

Programming notes

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

12.9.10 REPAINT CONTROL REGISTER 4 CR25 3X5H INDEX 25 (RW)

Bits 7-6 *Reserved,* read as ‘0’s

Bit 5 *Reserved.*

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

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

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

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

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

This register is set to 00h after reset.

12.9.11 REPAINT CONTROL REGISTER 5 CR27 3X5H INDEX 27 (RW)

Bits 7-3 **FIFO High Water Mark**. When the FIFO occupancy rises above this value, the CRTC will stop filling the FIFO. This field should only ever be set by the BIOS during mode switches. Do not set this register to a value greater than the default value D0h - nothing will work.

Bits 2-0 *Reserved*, read as ‘0’.

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

12.9.12 PALETTE CONTROL REGISTER CR28 3X5H INDEX 28 (RW)

Bit 7 **DAC Power Down.** Setting this bit to one turns off the digital to analog converters. This is useful for when a second graphics card is installed in the system and power needs to be saved by turning the motherboard graphics off.

Bit 6 **DAC Setup.** This bit specifies the blanking pedestal. Zero indicates a blanking pedestal of 0 IRE, one indicates 7.5 IRE.

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

Bit 4 *Reserved.* Must be writtezn to ‘1’.

Bit 3 **LUT Bypass.** Setting this bit to one bypasses the RAMDAC look up table (LUT) and allows 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 in Table 12-19 :

Table 12-19Palette Control Pixel Format encoding

Bit-2	Bit-1	Bit-0	Pixel Format
0	0	0	VGA standard 8 bit
0	0	1	8 bit color (non-VGA)
0	1	0	15-bit (555) direct color
0	1	1	16-bit (565) direct color
1	0	0	24-bit (888) direct color
1	0	1	32-bit (8888) direct color
	1	X	Reserved

This register resets to 08h after reset.

Programming notes

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

The following resolutions are supported at 75 Hz refresh rate for each of the above color depths when a 64 bit bank of DRAM is used for the frame buffer (see Table 12-20 *):*

Table 12-20Supported resolutions at 75 Hz with 64 bit DRAM banks

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

Interlaced monitors and timings are supported.

12.9.13 CURSOR HEIGHT REGISTER CR29 3X5H INDEX 29 (RW)

For the description of this register, see section section 13.11.1

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

12.9.14 CURSOR COLOR 0 REGISTER A CR2A 3X5H INDEX 2A (RW)

For the description of this register, see section section 13.11.2

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

12.9.15 CURSOR COLOR 0 REGISTER B CR2B 3X5H INDEX 2B (RW)

For the description of this register, see section section 13.11.3

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

12.9.16 CURSOR COLOR 0 REGISTER C CR2C 3X5H INDEX 2C (RW)

For the description of this register, see section section 13.11.4

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

12.9.17 CURSOR COLOR 1 REGISTER A CR2D 3X5H INDEX 2D (RW)

For the description of this register, see section section 13.11.5

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

12.9.18 CURSOR COLOR 1 REGISTER B CR2E 3X5H INDEX 2E (RW)

For the description of this register, see section section 13.11.6

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

12.9.19 CURSOR COLOR 1 REGISTER C CR2F 3X5H INDEX 2F (RW)

For the description of this register, see section section 13.11.7

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

12.9.20 GRAPHICS CURSOR ADDRESS REGISTER 0 CR30 3X5H INDEX 30 (RW)

For the description of this register, see section section 13.11.8

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

12.9.21 GRAPHICS CURSOR ADDRESS REGISTER 1 CR31 3X5H INDEX 31 (RW)

For the description of this register, see section section 13.11.9

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

12.9.22 GRAPHICS CURSOR ADDRESS REGISTER 2 CR32 3X5H INDEX 32 (RW)

For the description of this register, see section section 13.11.10

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

12.9.23 URGENT START CR33 3X5H INDEX 33 (RW)

Bits 7-0 **Urgent Start Position.** These bits represent the horizontal character count value at which urgency information will start to be generated for CRTC fetch requests. Prior to this position and after display enable negates, any CRTC fetches performed will be generated at low priority - ie. any CPU or blit operation will take precedence over CRTC regardless of CRTC FIFO occupancy. Once the horizontal character counter reaches this value, if CRTC FIFO occupancy is still below its low water mark then urgent fetches will be performed. Thereafter (and until the next display enable drops), as the CRTC FIFO drains, CRTC fetches will be marked urgent whenever the FIFO occupancy drops below its low water mark.

Programming notes

A value of 0xFFh means “always urgent” and should be used if the VGA screen is showing “Glitches”. By setting this value, the CPU and GE bandwidth will be reduced.

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

12.9.24 DISPLAYED FRAME Y OFFSET 0 CR34 3X5H INDEX 34 (RW)

Bits 7-0 **Frame Y Offset 0 Bits 7-0.** These bits represent bits 7-0 of a (2's complement) 16 bit scan line offset of the displayed frame relative to the Graphics Engine destination base.

This register is undefined after reset.

12.9.25 DISPLAYED FRAME Y OFFSET 1 CR35 3X5H INDEX 35 (RW)

Bits 7-0 **Frame Y Offset 1 Bits 7-0.** These bits represent bits 15-8 of the displayed frame scan line offset relative to the Graphics Engine destination base.

This register is undefined after reset.

12.9.26 INTERLACE HALF FIELD START CR39 3X5H INDEX 39 (RW)

Bits 7-0 **Interlace Horizontal Count.** This register defines the horizontal character count at which vertical timing is clocked during odd frames. When using interlaced operation, this register should be programmed to approximately half of the horizontal total value (CR0).
There is no explicit interlace enable bit. Rather, when this register is programmed to FFh, interlace is disabled. The value of this register is defined to be FFh after reset (interlace disabled).

12.9.27 IMPLEMENTATION NUMBER REGISTER CR3A 3X5H INDEX 3A (R)

Bits 7-0 **Implementation Number.** Indicates the hardware implementation number for the graphics drawing and display subsystem. Table 12-21 below describes the interpretation of each value

.

Table 12-21Implementation Number

VALUE	IMPLEMENTATION
01h	STPC Implementation

12.9.28 GRAPHICS VERSION REGISTER CR3B 3X5H INDEX 3B (R)

Bits 7-0 **Graphics Version Number.** Indicates the hardware version number for the graphics drawing and display subsystem. Table 12-22 below describes the interpretation of each value.

Table 12-22Graphics Version Number

VALUE	IMPLEMENTATION
01h	STPC Implementation

12.9.29 DRAM TIMING PARAMETER REGISTER CR3C 3X5H INDEX 3C (RW)

See section 4.4.1 for the description of this register

12.9.30 DRAM ARBITRATION CONTROL REGISTER 0 CR3D 3X5H INDEX 3D (RW)

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

12.9.31 DRAM ARBITRATION CONTROL REGISTER 1 CR3E 3X5H INDEX 3E (RW)

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

12.9.32 DDC CONTROL REGISTER CR3F 3X5H INDEX 3F (RW)

Bits 7-6 **DDC Write Data.** These two bits drive the DDC[1:0] open collector outputs. Writes to these bits affect the DDC[1:0] pins.
The DDC[1:0] pins are open collector outputs which are externally pulled up. Thus, programming either of these bits to a one disables the output driver and allows the pin to act as an input whose status can be read via bits 5-4 of this register.

Note that reads from these bits return the value of data last written to this register. This may not be the same as the data actually on the bus if another master is driving it. Bits 5-4 of this register accurately reflect the data on the bus no matter who is driving it.

Bits 5-4 **DDC Read Data.** These read-only bits return the read status of the DDC[1:0] pins.

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

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

This register defaults to FFh after reset.

12.10 ADDITIONAL MODES

12.10.1 FAST 132 CHARACTER WIDE TEXT MODE.

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

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

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

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

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

12.11 INTERLACED MONITOR SUPPORT

Section 4.7.6.26a describes the “interlace half field start” register field. Setting this field to a value other than FFh (the power on reset default) enables interlaced CRT timing generation.

In interlaced timing mode, the horizontal and vertical timing parameters (CR0-CR7, CR10-CR12, CR15, CR16) should be programmed to values equal to what they would otherwise take in non-interlaced modes with the following modifications:

- Horizontal period must be an even number of character clocks. This results in the requirement that CR0[0] must equal ‘1’.

- Interlace half field start (CR39) must be set equal to CR4 - (CR0 + 5)/2.

- Vertical period must be an odd number of scan lines. That is, CR6[0] must be set to 1.

- Vertical overscan period should be an even number of scan lines. That is the vertical blank start field must be odd (CR15[0] = ‘1’) and vertical blank end field must be even (CR16[0] = ‘0’). If this is not observed the top and bottom lines of the border will be only half a scan line wide on alternate fields. If no border will be displayed then there is no retriction on vertical blank start and end.

All other registers should be programmed as they would for the same resolution and color depth in non-interlaced mode

12.12 RAMDAC REGISTERS

12.12.1 PALETTE PIXEL MASK REGISTER 3C6H (RW)

This eight-bit mask register is ANDed with the pseudo-color pixel before doing the palette look-up. This provides an alternate way of altering the displayed colors without changing the display memory or color palette.

This register defaults to FFh after reset.

12.12.2 PALETTE READ INDEX REGISTER 3C7H (W)

This register contains the index value for the read access to the 256 entries of the color palette. Each entry is 24-bits wide (8-bits each for R, G and B) and is read as sequence of 3-Bytes. After writing the index of the entry to be read, the actual contents of the selected palette entry are read by doing 3 consecutive Byte reads from the DAC Data port (3C9h) in sequence: 1) Red, 2) Green and 3) blue. This 3-Byte read sequence is aborted and a new one is started if either the Read or Write Index register is written before reading the third Byte.

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

This register is a write only register. Reads from this address do not return the contents of the index register. The Palette state register contents are returned instead.

12.12.3 PALETTE STATE REGISTER 3C7H (R)

This is a read only register and contains the two least significant bits of the last IO writes to IO 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.

12.12.4 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.

12.12.5 PALETTE DATA REGISTER 3C9H (RW)

This register is used in conjunction with the Read and the Write index register to access the look-up table. Reads from this port return the contents of the entry pointed to by the Read Index 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.

12.13 DCLK CONTROL REGISTERS

These registers control the Dot Clock or pixel clock which the VGA uses to display the pixels on the screen.

12.13.1 DCLK CONTROL REGISTER 00 DCLK00 INDEX 42H

This is one of the 4-pairs of Dot Clock control registers. This pair (00 and 01)is selected when bits 3-2 of VGA Miscellaneous Output register is set to 00.

Bit 7 *Reserved.*

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

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

This register defaults to 0x76h at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 25.18 MHz assuming 14.318 MHz oscillator clock as the reference input.

12.13.2 DCLK CONTROL REGISTER 01 DCLK01 INDEX 43

This is one of the 4-pairs of Dot Clock control registers. This pair (00 and 01) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 00.

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

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

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

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

12.13.3 DCLK CONTROL REGISTER 10 DCLK10 INDEX 44H

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

Bit 7 *Reserved*.

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

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

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

12.13.4 DCLK CONTROL REGISTER 11 DCLK11 INDEX 45

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

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

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

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

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

12.13.5 DCLK CONTROL REGISTER 20 DCLK20 INDEX 46H

This is one of the 4-pairs of Dot Clock control registers. This pair (20 and 21) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 10.

Bit 7 *Reserved.*

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

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

This register defaults to 0x5Bh at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 40 MHz assuming 14.318 MHz oscillator clock as the reference input.

12.13.6 DCLK CONTROL REGISTER 21 DCLK21 INDEX 47H

This is one of the 4-pairs of dot clock control registers. This pair (20 and 21) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 10.

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

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

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

This register defaults to 0x6D at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 40 MHz assuming 14.318 MHz oscillator clock as the reference input.

12.13.7 DCLK CONTROL REGISTER 30 DCLK30 INDEX 48H

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

Bit 7 *Reserved*

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

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

This register defaults to 0x6E at reset. This value when combined with the default value of the other half of this pair results in a dot clock of 60 MHz assuming 14.318 MHz oscillator clock as the reference input.

12.13.8 DCLK CONTROL REGISTER 31 DCLK31 INDEX 49

This is one of the 4-pairs of Dot Clock control registers. This pair (00 and 01) is selected when bits 3-2 of VGA Miscellaneous Output register is set to 11.

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

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

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

This register defaults to 0x69h at reset. This value when combined with the default value of the other half of this pair results in a Dot Clock of 60 MHz assuming 14.318 MHz oscillator clock as the reference input. This register can be accessed using IO cycles to index register 4Ah.

12.14 UPDATE HISTORY FOR VGA CONTROLLER CHAPTER

The following changes have been made to the VGA Controller Chapter on 20/10/99.

Section	Change	Text
12.8.1	Replaced	“~~Five~~” With “Seven”

The following changes have been made to the VGA Controller Chapter on 12/10/99.

Section	Change	Text
12.13	Added	DCLK Control registers descriptions

Update History for VGA Controller chapter

13. GRAPHICS ENGINE

13.1 INTRODUCTION

The Graphics Engine (GE) performs limited graphics drawing operations. The results of these operations changes the contents of the on-screen or off-screen frame buffer areas of DRAM memory.

Pixel depths of 8, 16, 24 and 32 bits are fully supported by the GE.

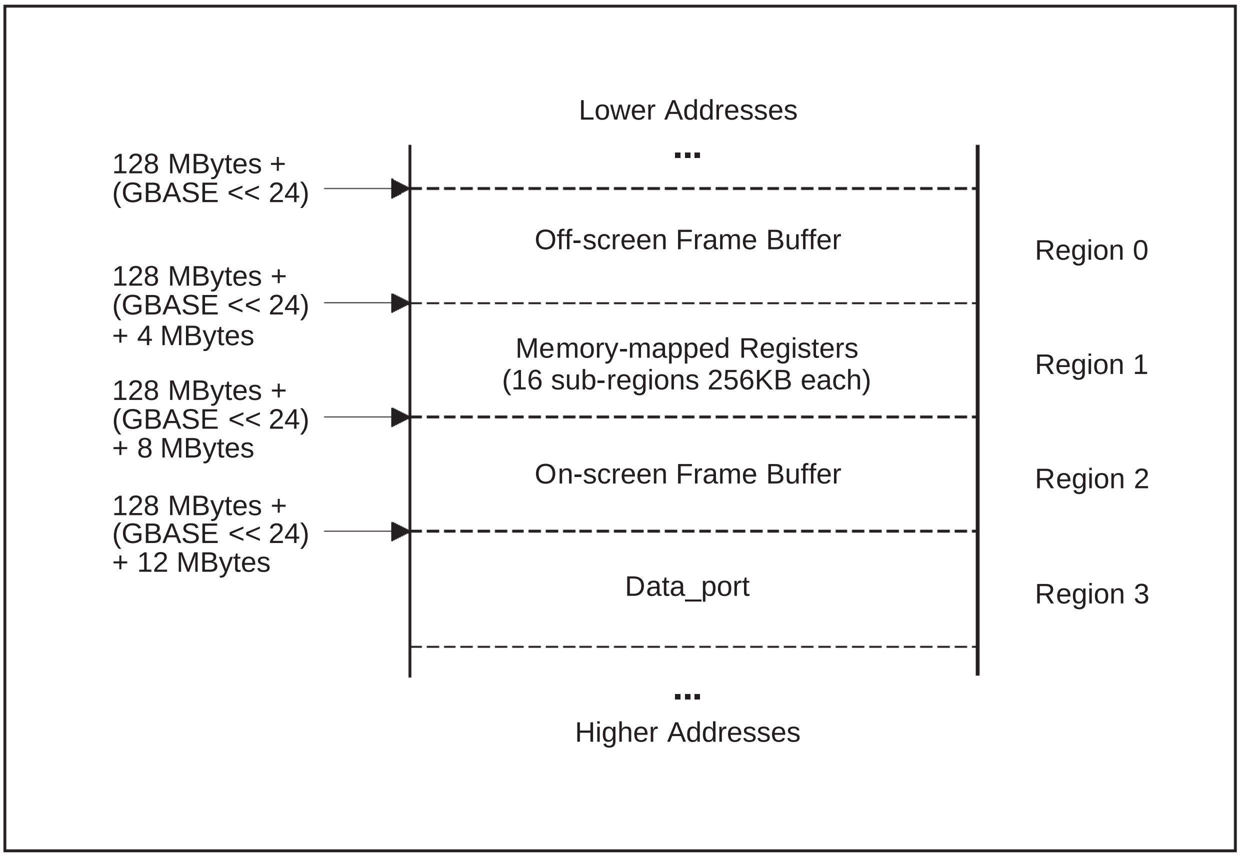
13.2 MEMORY ADDRESS SPACE

The extended (non-VGA) graphics and video functions of Graphics Engine occupy 16 MBytes of memory address space. This space can be located anywhere in the memory on any 16 MByte boundary between 128 MBytes and 256 MBytes. The 16 MByte region is divided into four parts as shown in Figure 13-1:

In this figure, the GBASE is Extended CRTC Register 20 (CR20) and provides bits 26 through 24 of the starting address where the CPU sees the extended graphics and video functionality.

This 16-MByte space can be linearly (one to one) mapped in the CPU address space or can be accessed via 64K aperture located at A0000h-AFFFFh. The aperture access method is described in more detail in section “CR21 Graphics Extended Aperture Register” and facilitates the access to extended functionality in real mode.

Figure 13-1GE 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 the frame buffer. The frame Buffer reads from either: 128MBytes + (GBASE<<24) or 128MBytes + (GBASE<<24) + 8MBytesand they are done identically.

However, writes to any area of the Frame Buffer that might be displayed should be done to the region 128MBytes + (GBASE<<24) + 8MBytes. Writes to areas of the Frame Buffer that are not displayed should be done to 128MBytes + (GBASE<<24). The Frame Buffer address used by the host also informs the GE whether the access is being made to an on-screen or off-screen drawing area. It is important that the software never draws using an off-screen area address to a portion of the Frame Buffer that is currently being displayed in order to be compatible with future versions of the GE.

The Frame Buffer addresses loaded into GE registers are from 0 to 4MBytes. The source, pattern or destination of a GE operation can be located anywhere in Frame Buffer DRAM. However, the DRAM physical address must be known; the entire operand must be contiguous in physical memory (the GE does not do scatter/gather), and the operands must not move from the specified memory location until the drawing operation is completed.

All registers needed for the extended graphics and video functionality are mapped in a 4-MByte region of its own. This region is further divided in 16 256KByte sub-regions illustrated in Table 13-1 below:

Table 13-1Graphic memory subdivisions

Sub-region	Region function
0	2-D Graphics Engine registers
1	Reserved
2	Video overlay registers
3-7	Reserved for future functionality
8	Video Input Port Registers
9-15	Reserved for future functionality

Writes done to any double word between (128MBytes + (GBASE << 24) + 12 MBytes) to (128MBytes + (GBASE << 24) + 16 MBytes) will be the same as a write to the Data_port register of the 2-D graphics engine. This region of 1 million aliases of the Data_port is provided to allow the use of string move instructions for Host-to-screen BitBlts.

All registers can be read with accesses of any width. The CPU can read any register via Byte (8-bit), word (16-bit), or double-word (32-bit) accesses. Writes, must be done using double-word (32-bit) transfers.

Note that the contents of all GE registers are not defined after reset.

Software must initialize all registers upon power-up before attempting any drawing operation.

13.3 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.

13.4 ADDRESSING

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

The Frame Buffer address used by the host also informs the GE whether the access is being made to an on-screen or off-screen drawing area. It is important that the software never draws using an off-screen area address to a portion of the Frame Buffer that is currently being displayed.

The addresses that are loaded into GE registers are the physical DRAM addresses after the OS address translation and GE host address remapping is done. The Frame Buffer addresses loaded into GE registers are from 0 to 4 MBytes. The source, pattern or destination of a GE operation can be located anywhere in the Frame Buffer space.

13.5 VGA OPERAND SOURCES

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

1) The frame buffer memory (i.e. a location in the DRAM memory that is dedicated to the graphics sub-system, and which may or may not be currently displayed by the CRT controller)

2) The host-supplied data

3) The on-chip color registers

13.5.1 OPERAND SELECTION

Some operands are color pixels and others are monochrome bitmaps. In general, the data written to the Destination address is the result of a Raster Operation (ROP) performed upon three pixel-depth color inputs:

1) The Source, which can originate from frame buffer memory (for Screen-to-screen BitBlts), from the Host (for Host-to-screen BitBlts), or from the Foreground and Background color registers

2) The pattern, must originate from the frame buffer

3) The destination, must originate from the frame buffer

When one or more of these operands are the inputs to an 8-bit Windows' ROP, the result is written to the destination.

If the ROP does not use a source operand, then the “Source” field of the ROP register must be set to CONSTANT_FILL (see ROP register 9.5.5.11) to prevent wasted performance due to needless operand fetching.

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

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

13.5.2 TRANSPARENT MODE

Transparent mode drawing leaves some of the destination pixels untouched. The GE supports four types of transparent mode drawing:

1) Bitmap transparency, where bits that are ‘1’ are expanded to the Foreground color register value and drawn, but bits that are expanded to ‘0’ are not drawn.

2) Pattern transparency, where any Pattern Bytes that are ‘0’ suppresses writing to the corresponding destination pixel Bytes.

3) Source transparency, where any Source pixel which either matches the value or do not match the value of the source transparency register is not drawn.

4) Destination transparency, where any Destination pixel that either matches the value or does not match the value of the source transparency register is not drawn.

These modes are controlled by fields in the ROP register.

13.6 VGA OPERAND FRAME BUFFER ADDRESSES

The GE fetches needed data from the frame buffer area. The software identifies these areas with an operand base address, unsigned X and Y Indeces from this base address, and a pitch for that region. The pitch is the Byte distance between two pixels which are in the same X position of adjacent scan lines.

The frame buffer is addressed using DRAM linear addresses. These are the addresses that the DRAMs are presented with. The frame buffer starts at DRAM linear address 0 and continues until the top of the frame buffer. The system's physical addresses are mapped to above the frame buffer. To accommodate a more natural view of the frame buffer, the GE implements X-Y addressing. An operand's base address, pitch, X and Y components are combined in the GE, to form the associated DRAM linear address. The base component of an operand is the DRAM linear address at the start of that operand. That address can range from 0 to the maximum size of the frame buffer, depending upon where the operand is located in the frame buffer.

A pixel's X coordinate is usually expressed as an unsigned Byte quantity, the number of Bytes from the left edge of a scan line.

If the X_dir field of the Pixel_depth register is '0', advancing from left-to-right, then X points to the least-significant Byte of the starting pixel.

If the X_dir field of the Pixel_depth register is '1', advancing from right-to-left, then X points to the most-significant Byte of the starting pixel.

Mathematically, consider a BitBlt region that starts at (x0, y0), where "x0" is in pixels. This region is W+1 Bytes wide, is H+1 scan lines high and has BPP (Bytes-per-pixel).

Then the starting address that must be programmed into the GE is depend on the X_dir and Y_dir fields of the Pixel_depth register. This is illustrated in Table 13-2 :

Table 13-2 Detail GE starting address register

X_dir	Y_dir	Starting Address
0	0	(x0 * BPP, y0)
0	1	(x0 * BPP, y0 + H)
1	0	(x0 * BPP + W, Y0)
1	1	(x0 * BPP + W, Y0 + H)

Table 13-2 Detail GE starting address register

Note that movement in the negative X direction (i.e. X_dir set to '1') is only defined for Screen-to-screen color BitBlts.

When bitmap expansion is enabled, the X field of the Src_XY register is a bit address and not a Byte address. In other words, the least significant three bits of Src_XY.X refers to the bit within a Byte of bitmap data.

Internally, the GE performs its calculations using the X and Y coordinates. When a DRAM linear address is needed, for example to write a destination pixel, the address is computed using:

linear_address = operand_base + (Y * pitch) + X *BPP

The multiplication by pitch is done using hardwired shifts and adds. The pitch is actually specified as a group of 4 shift codes. For each non-zero shift code, the Y address is shifted by a corresponding number of bits and then added to the total. The resulting sum is then added to X and the base address to obtain the DRAM linear address. The shift values supported are shown in Table 13-3 :

Table 13-3Shift values supported

Value	Shift0	Shift1	Shift2	Shift3
000	0	0	0	0
001	0	0	32 * Y	1024 * Y
010	0	64 * Y	64 * Y	2048 * Y
011	128 * Y	128 * Y	128 * Y	4096 * Y
100	256 * Y	256 * Y	256 * Y	n/a
101	512 * Y	512 * Y	512 * Y	n/a
110	1024 * Y	1024 * Y	0	n/a
111	2048 * Y	0	0	n/a

The operand base addresses must be aligned to 32 Bytes (that is, the 5 least significant bits of the address must be zeros).

Table 13-4Encoded Dst_XY registers

4-MByte region 1 (memory mapped regs)

256KByte sub-region 0

31 27	26 24	23 22	21 18	17 - 16	15 14	13 2	1 - 0
00001	<GBASE>	0 1	0000	0 - 1	<Cmd>	<Count>	1 - 0

The supported pitches (in Bytes) are:

0, 32, 64, 96, 128, 160, 192, 224, 256, 288, 320, 352, 384, 416, 448, 512, 544, 576, 608, 640, 672, 704, 768, 800, 832, 896, 1024, 1056, 1088, 1120, 1152, 1184, 1216, 1248, 1280, 1312, 1344, 1376, 1408, 1440, 1472, 1536, 1568, 1600, 1632, 1664, 1696, 1728, 1792, 1824, 1856, 1920, 2048, 2080, 2112, 2144, 2176, 2208, 2240, 2272, 2304, 2336, 2368, 2400, 2432, 2464, 2496, 2560, 2592, 2624, 2656, 2688, 2720, 2752, 2816, 2848, 2880, 2944, 3072, 3104, 3136, 3168, 3200, 3232, 3264, 3328, 3360, 3392, 3456, 3584, 3616, 3648, 3712, 3840, 4096, 4128, 4160, 4192, 4224, 4256, 4288, 4320, 4352, 4384, 4416, 4448, 4480, 4512, 4544, 4608, 4640, 4672, 4704, 4736, 4768, 4800, 4864, 4896, 4928, 4992, 5120, 5152, 5184, 5216, 5248, 5280, 5312, 5376, 5408, 5440, 5504, 5632, 5664, 5696, 5760, 5888, 6144, 6176, 6208, 6240, 6272, 6304, 6336, 6400, 6432, 6464, 6528, 6656, 6688, 6720, 6784, 6912, 7168, 7200, 7232, 7296, 7424, 7680

13.6.1 COMMAND INITIATION

The destination coordinate register, Dst_XY, appears multiple times in the address space. Reading from any of these appearances, or aliases, is equivalent. Writing to most of these aliases also has the effect of initiating a drawing command. Which command is begun depends upon the address written to. There is also an address that just provides write access to the Destination register, with no other side-effects.

The operations are encoded in the Dst_XY register shown in Table 13-4 :

Where “GBASE” is the contents of the Extended CRTC Register 20 (CR20). Bits 23 through 16 are ‘01000001’ to identify this as a Dst_XY register access. This is shown in Table 13-5 :

Table 13-5CMD operations

CMD	Operation
00	Simple BitBlt, all registers must be set up before this command is issued (the Count field is ignored)
01	Width-specified BitBlt, the Count field of the address is used as the width for the operation, all other relevant registers (height, ROP, etc.) must be set up before this command is issued
10	Height-specified BitBlt, the Count field of the address is used as the height for the operation, all other relevant registers (width, ROP, etc.) must be set up before this command is issued
11	Write to Dst_XY without starting a BitBlt operation, (used for diagnostic applications)

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.

13.7 DRAWING ENGINE REGISTERS

The software controls the graphic drawing by writing to the GE's registers to set-up and initiate an operation. Any data that must be provided by the host is written to the Data_port. The Data_port can be referenced via the Data_port "register" or via the 4 MB window of aliases.

All registers can be read with accesses of any width. The CPU can read any register via Byte (8-bit), word (16-bit), or double-word (32-bit) accesses.

A register that is exclusively for the use of software, "Xtra", is included in the GE but has no influence on any drawing operation or on the display.

13.8 REGISTER ACCESS

Except for the Dst_XY register, discussed in the previous section, the memory-mapped GE registers and the Data Port are accessed by reading or writing to an address of the form. This is illustrated in Table 13-6 :

Where "Index" specifies the offset of the register to be accessed from the start of the GE memory-mapped register address space. The least significant 2 bits of Index will always be '00'. The following sections will list the "Index" value along with a description of each register.

Reads may be done in any width, but writes must be done as 32 or 64 bit transfers.

In general, the CPU should not write to any of the GE registers when the GE is busy. If such an access is done, the CPU may be held for a long period of time, possibly for the duration of a large BitBlt. Reads of GE registers (except for Status) may return invalid data if the GE is busy.

The Dst_XY, Src_XY, Width and Height registers are double buffered and the CPU may write the next values to these registers while a prior operation is being performed. The last write to any double buffered register done before a write to Dst_XY will be the one used for the next operation. Any writes done to a GE register after a write is done to the Dst_XY while the GE is busy will hold the CPU until the first operation is completed and the pending register values are used for the second operation. During normal operation, the CPU writes to the Dst_XY register for text and line segments, reducing the hold period as much as possible.

The GE Status register may be accessed at any time.

13.8.1 DATA PORT ACCESS

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

In normal operation for text drawing (done as a bitmap expanded Host-to-screen BitBlts), the CPU-writes exact amount of data to the Data Port for the current character, then starts on the next text character by writing to the Dst_XY register and finally writes data for the next text character. The current operation reads from the Data Port FIFO until its needs are met. Then the next BitBlt operation reads its data from the Data Port FIFO. To assure correct results, the software must write the correct number of 32-bit double words to the Data Port FIFO for all BitBlt operations.

The CPU can write to the Data Port at any time. If the GE is unable to accept any additional writes to the Data Port, the CPU is held off until the write command can be accepted. If a PCI master requests bus access when the CPU has been held off for a long period of time (128 clocks cycles), then the GE forces a CPU retry via the backoff (BOFF) mechanism.

Table 13-6GE and Data_Port access

4-MByte region 1 (memory mapped regs)

256KByte sub-region 0

31 27	26 24	23 22	21 18	17 12	11 0
00001	<GBASE>	0 - 1	0000	000000	<Index>

13.9 REGISTER SPECIFICATION

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

13.9.1 BACKGROUND COLOR REGISTER INDEX = 004H (BACKGROUND)

This register contains the full-color value(s) that a ‘0’ bit is expanded to. This expansion is done for BitBlt operations with bitmap expansion specified in the Expand field of the ROP register, and for operations with the Source field of the ROP register set to CONSTANT_FILL.

For 8-bit pixels, only bits 7-0 are significant. For 16-bit pixels, only bits 15-0 are significant. For 24-bit pixels, only bits 23-0 are significant.

Bits 31-0 **Background Color.** This is the color to be used as the background when expanding bitmap ‘0’ values or when using CONSTANT_FILL as the source operand.

The content of this register is defined to be zero after reset.

Programming notes

The content of this register is not altered by drawing operations.

13.9.2 CURSOR COORDINATE REGISTER INDEX = 11CH (CURSOR_XY)

This register contains the address of the upper-left-hand corner of the cursor. To eliminate the cursor, its address should be set to a value large enough so that none of the cursor is on the displayed screen. Note that when set to (0,0), the entire cursor may be displayed on the upper-left hand corner of the display.

Bits 31-29 *Reserved.*

Bits 28-16 **CYUL**. The Y location of the upper-left-hand corner of the cursor.

Bits 17-13 *Reserved.*

Bits 12-0 **CXUL**. The X location of the upper-left-hand corner of the cursor.

The contents of this register are not defined after reset.

Programming notes

To suppress cursor display, enter one more than the number of display scan lines into the Y field.

The contents of this register remain unaltered throughout drawing and display operations.

13.9.3 TOP OF DATA FIFO INDEX = 804H (DATA_PORT)

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

Bits 31-0 **Data_Port.**

Programming notes

The CPU can write to the Data Port at any time. If the GE is unable to accept any additional writes to the Data Port, the CPU will be held off until the write can be accepted.

Note that writing to this address is the same as writing to any double word between (128MBytes+(GBASE << 24) + 12 MBytes) to (128MBytes+(GBASE << 24) + 16 MBytes).

The Data FIFO is empty after reset.Destination Operand Base Address Register Index = 018h (DST_Base)Destination Operand Base Address Register Index = 018h (DST_Base)

13.9.4 DESTINATION OPERAND BASE ADDRESS REGISTER INDEX = 018H (DST_BASE)

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

Bits 31-21 *Reserved.*

Bits 21-0 **DstOp_Base.** Base DRAM linear address of the destination operand with 16 Byte alignement. Lower five Bytes are reserved and are set to ‘0’.

The contents of this register are not defined after reset.

Programming notes

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

13.9.5 DESTINATION PITCH REGISTER INDEX = 028H (DIST_PITCH)

This register specifies the number of Bytes needed to advance from a pixel in one scan line of the Destination to the corresponding pixel in the next scan line. This value is always positive. The Y_dirfield of the Pixel_depth register controls how the operation advances to the next scan line.

Only a limited number of pitches are supported. The supported ones (in Bytes) are listed in the Src_pitch description.

Bits 31-11 *Reserved.*

Bits 10-9 **Dst_shift3.** These bits specify an amount to multiply Dst_XY.Y, this result along with the other shift results, is added to the Dst_base and Dst_XY.X to compute the DRAM linear address of the destination pixel. See Table 13-7 for the multiplication values that this field can specify.

Table 13-7DRAM address multiplication factor

Value	Shift0	Shift1	Shift2	Shift3
000	0	0	0	0
001	0	0	32 * Y	1024 * Y
010	0	64 * Y	64 * Y	2048 * Y
011	128 * Y	128 * Y	128 * Y	4096 * Y
100	256 * Y	256 * Y	256 * Y	n/a
101	512 * Y	512 * Y	512 * Y	n/a
110	1024 * Y	1024 * Y	0	n/a
111	2048 * Y	0	0	n/a

Bits 8-6 **Dst_shift2.** See Dst_shift3, above.

Bits 5-3 **Dst_shift1.** See Dst_shift3, above.

Bits 2-0 **Dst_shift0.** See Dst_shift3, above.

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

The contents of this register are not defined after reset.

Programming notes

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

13.9.6 DESTINATION OPERAND COORDINATE REGISTER INDEX = 10000H FOR STANDARD USE (DST_XY)

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

Bits 31-16 **Dst_Y.** The unsigned Y coordinate of the starting corner of the destination operand.

Bits 15-0 **Dst_X**. The unsigned X location of the starting corner of the destination operand. This value must be a multiple of Pixel_depth.

The complete addressing of this register is described in Section 13.6.1. Command Initiation.

The contents of this register are not defined after reset.

Programming notes

The address of this register is also used to determine which of the BitBlt operations is to be performed (Simple BitBlt, Width-specified BitBlt or Height-specified BitBlt). Writing to this register initiates a graphics operation.

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

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

13.9.7 FOREGROUND COLOR REGISTER INDEX = 034H (FRG_CL)

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

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

The contents of this register are defined to be FFFFFFFFh.

Programming notes

This expansion is done for BitBlt operations with bitmap expansion specified in the Expand field of the ROP register.

For 8-bit pixels, only bits 7-0 are significant. For 16-bit pixels, only bits 15-0 are significant. For 24-bit pixels, only bits 23-0 are significant.

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

13.9.8 HEIGHT REGISTER INDEX = 048H (HEIGHT)

This register contains one less than the number of scan lines in the Source and Destination areas. The contents of this register will not change during the execution of a command.

Bits 31-16 *Reserved.*

Bits 15-0 **Height.** The value set in these bits must be one less than the height, in scan lines, of the source and destination areas.

The contents of this register are not defined after reset.

Programming notes

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

This register can be loaded by writing to the Dst_XY register using one the Height-specified address alias. For this case, the Height register is loaded with the value in the Count field of the address described in Table 13.6.1

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

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

13.9.9 PATTERN BASE ADDRESS OPERAND INDEX = 058H (PATTERN)

This register contains the starting DRAM linear address of the Pattern operand, including the aligned base address, the first row to be displayed and a starting Byte number for 24-bit pixels.

Bits 31-22 *Reserved.*

Bits 21-8 **Pattern base.** These bits specify the starting DRAM physical address of the Pattern, operand, aligned to a 256 Byte boundary

Bits 7-5 *Reserved.*

Bits 4-3 **Pat_X_start.** For 24-bit pixels, these bits must be set to:

(Dst_X / 8) modulo 3

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

Bits 2-0 *Reserved.*

The contents of this register are not defined after reset.

Programming notes

The start of Pattern data must be aligned to a 256-Byte boundary. Advancing to the next Pattern data row will be done modulo 8 rows. Regardless of the number of Pixel_depth, the Pattern row is 32 Bytes long.

The Pattern register can be loaded with the address of the last row of Pattern data and the GE will wrap-around to the start of the pattern on the second row. Note that the Pattern register advances by increasing the address regardless of the X_dir, Y_src_dir or Y_dst_dir fields of the Pixel_depth register.

For further discussion of the Pattern Data, see Section 13.10.1*, “Pattern Data”.*

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

13.9.10 PIXEL DEPTH OPERAND INDEX = 07CH (PIXEL_DEPTH)

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

Bits 31-8 *Reserved.*

Bit 7 **Y_src_dir.** When this bit is set to ‘0’, source pixels advance from upper scan lines to lower scan lines (from smaller linear to larger linear addresses). Setting this bit to ‘1’ reverses the direction of BitBlt source operations. This bit should be set to ‘0’ for all operations other than reverse-direction non-bitmap-expanded screen-to-screen BitBlts.

Bit 6 **Y_dst_dir.** When this bit is set to ‘0’ destination pixels advance from upper scan lines to lower scan lines (from smaller to larger linear addresses). Setting this field to ‘1’ reverses the direction of BitBlt destination operations. This field should be set to ‘0’ for all operations other than reverse-direction non-bitmap-expanded screen-to-screen BitBlts.

Bit 5 **X_dir.** When this bit is set to ‘0’, pixels advance from left to right, and when set to ‘1’ they advance from right to left. This field can be set to ‘1’ only for Screen-to-screen BitBlts and horizontal scan line fills.

Bits 4-2 *Reserved.*

Bits 1-0 **Pixel depth.** The only supported values for this field are shown in Table 13-8 :

Table 13-8Supported Pixel depth values

Bit1	Bit0	Pixel Depth
0	0	1 Byte per pixel
0	1	2 Bytes per pixel
1	0	3 Bytes per pixel
1	1	4 Bytes per pixel

The contents of this register are not defined after reset.

Programming notes

Note that the 4th Byte of 4-Byte pixels is not used in the display, but is processed by drawing operations. Zeros should be written to this 4th Byte to *preserve compatibility with future versions of this architecture.*

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

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

13.9.11 RASTER OPERATION REGISTER INDEX = 08CH (ROP)

This register contains the ROP code to be applied during processing a pixel, enables bitmap expansion, selects transparent modes, and controls the source operand. This is summarise in Table 13-9

Table 13-9Summary of ROP Functions

Bits	Function
31:30	SRC operand type
29	Use PAT operand
28	Use DST operand
27	GE Diagnostic mode
26:16	Unused/Reserved
15	DST transparency mode
14	DST transparency match
13	SRC transparency mode
12	SRC transparency match
11	Packed SRC data
10	SRC bitmap expansion
9	PAT transparency mode
8	SRC bitmap transparency
7:0	Raster operation code

Bits 31-30 **Source.** These bits determine the SRC operand. Possible values are shown in Table 13-10 :

Table 13-10Detail of SRC operand functions

Bit 31	Bit 30	Function	Source
0	0	CONSTANT_FILL	Background color register
0	1	SCREEN	screen or frame buffer
1	0	HOST	host CPU
1	1	Reserved	

This field MUST be set to CONSTANT_FILL if the Raster Operation requires no SRC operand (such as in inverting the destination as well as constant fills). Failure to set this field correctly can result in a degradation of performance.

Note that CPU writes to the Data Port will 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 this bit should be set to '0'. When set to '1', GE register reads will be done from an alternative path for diagnostic verification.

Bits 26-16 *Reserved.*

Bit 15 **DST transparency mode.** When this is set to '1', pixels are selectively modified based upon a comparison of the DST data from the frame buffer vs. the DST transparency compare register. The results of the comparison are interpreted based upon the DST transparency match bit. Note that the DST present bit must also be set to '1' when this bit is set. This mode is not valid when the pixel depth is 3 Bytes per pixel.

Bit 14 **DST transparency match.** This mode applies only when DST transparency mode is set. When this bit is set to '1', pixels with DST data that match the DST transparency compare register will be modified. When it is set to '0', pixels with DST data that do not match the DST transparency compare register will be modified.

Bit 13 **SRC transparency mode.** When this is set to '1', pixels are selectively modified based upon a comparison of the SRC data vs. the SRC transparency compare register. The results of the comparison are interpreted based upon the SRC transparency match bit. This mode is only meaningful when using non-bitmap screen or host data as the source. Transparency for bitmap source data should not use this mode, but rather the SRC bitmap transparency mode. This mode is not valid when the pixel depth is 3 Bytes per pixel.

Bit 12 **SRC transparency match.** Applies only when SRC transparency mode is set. When this bit is set to '1', pixels with SRC data that match the SRC transparency compare register will be modified. When this is set to '0', pixels with SRC data that do not match the SRC transparency compare register will be modified.

Bit 11 **Packed.** If set to ‘1’, the source will be read in packed mode. Effectively, the source is viewed as a continuous stream of data. At the end of a destination scan line, any data remaining in the last-used source Dword is applied to the start of the next destination scan line.

When this bit is set to ‘0’, any remaining source data is discarded at the end of a destination scan line. New source data is read from the next source scan line to apply to the start of the next destination scan line.

Bit 10 **Expand.** If set to ‘1’, the bitmap expansion will be enabled and source data from screen or host is assumed to be bitmap data. If set to ‘0’, source data is assumed to be color data with the depth specified in the Pixel_depth register.

Bit 9 **PAT transparency mode.** When this bit is set to ‘1’, pixels are selectively modified based upon the value of corresponding pattern data. Pattern Bytes that are set to zero are not modified. Note that the PAT present bit must also be set to ‘1’ when this bit is set.

Bit 8 **Bitmap transparency mode.** When this bit is set to ‘1’, pixels are selectively modified based upon the pre-expanded bitmap value. Pixels with corresponding bitmap values of zero are not modified. Pixels with corresponding bitmap values of one are written with the foreground value. Note that the expand bit must also be set when using this mode.

Bits 7-0 **ROP**, the raster operation used when computing a pixel result value.

The contents of this register are not defined after reset.

Programming notes

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

13.9.12 SOURCE OPERAND BASE ADDRESS REGISTER INDEX = 098H (SCR_BASE)

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

Bits 31-22 *Reserved.*

Bits 21-5 **SrcOp_Base.** Base linear address of the source operand.

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

The contents of this register are not defined after reset.

Programming notes

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

13.9.13 SOURCE PITCH REGISTER INDEX = 0ACH (SRC_PITCH)

This register specifies the number of Bytes needed to advance from a pixel in one scan line of the source to the corresponding pixel in the next scan line. This value is always positive. The Y_src_dir field of the Pixel_depth register controls how the operation advances to the next scan line.

Only a limited number of pitches are supported. The supported pitches (in Bytes) are described in Section 13.6.

Bits 31-11 *Reserved.*

Bits 10-9 **Src_shift3.** These bits specify an amount to multiply Src_XY.Y, this result along with the other shift results, is added to the Src_base and Src_XY.X to compute the DRAM linear address of the source pixel. See the Table 13-11 below for the multiplication values that this field can specify.

Table 13-11Scr_shift3 multiplication factors

Value	Shift0	Shift1	Shift2	Shift3
000	0	0	0	0
001	0	0	32 * Y	1024 * Y
010	0	64 * Y	64 * Y	2048 * Y
011	128 * Y	128 * Y	128 * Y	4096 * Y
100	256 * Y	256 * Y	256 * Y	n/a
101	512 * Y	512 * Y	512 * Y	n/a
110	1024 * Y	1024 * Y	0	n/a
111	2048 * Y	0	0	n/a

Bits 8-6 **Src_shift2.** See Src_shift3, above.

Bits 5-3 **Src_shift1.** See Src_shift3, above.

Bits 2-0 **Src_shift0.** See Src_shift3, above.

The contents of this register are not defined after reset.

Programming notes

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

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

13.9.14 SOURCE COORDINATE REGISTER INDEX = 0BCH (SCR_XY)

This register contains the coordinate address of the starting corner of the source operand.

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

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

The contents of this register are not defined after reset.

Programming notes

The “starting” corner is controlled by the X_dir and Y_src_dir fields of the Pixel_depth register.

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

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

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

13.9.15 STATUS REGISTER INDEX = 908H (STATUS)

Bit 31 **GE_Busy.** This read/write bit is set to ‘1’ when the GE is busy, GE register accesses that are done when this bit is set may result in the CPU being held for the duration of the current operation.

Bit 30 **Pending Busy.** This read-only bit is set to ‘1’ when the Dst_XY pending register has data in it. If GE writes to that registers when this bit is set it may result in the CPU being held for the duration of the current operation. GE register reads always return data without holding the CPU, but the data returned from the read may not be valid. The Status register may be read at any time and the operation will return valid data. Note that Pending Busy implies Busy, that is the Pending Busy field can be set to ‘1’ only if the Busy field is also set to ‘1’.

Bits 29-0 *Reserved.* These may read as one or zero.

After reset, bits 31-30 read as zero. All other bits are undefined.

13.9.16 WIDTH REGISTER INDEX = 0C8H (WIDTH)

This register contains the one Byte width less than the destination operand.

This register can also be loaded by writing to the Destination register using one of the Text or Line Segment commands. For these cases, the Width register is loaded with the value in the Count field of the address described in Section 13.6.1, “Command Initiation”.

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

Bits 31-16 *Reserved.*

Bits 15-0 **Width.** These bits should be set to one less than the number of Bytes across the destination area. This value should be a multiple of Pixel_depth, because only the number of Bytes specified in this field will be modified.

The contents of this register are not defined after reset.

Programming notes

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

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

13.9.17 EXTRA USE REGISTER INDEX = 0D4H (XTRA)

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

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

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

The contents of this register are not defined after reset.

Programming notes

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

13.9.18 SRC TRANSPARENCY COMPARE INDEX = ECH (ST_COMP)

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

13.9.19 DST TRANSPARENCY COMPARE INDEX = FCH (DT_COMP)

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.

13.9.20 NOTES ON: INTERACTIONS BETWEEN BLT OPERATIONS AND VGA FRAMEBUFFER ACCESSES

The GE performs two major classes of operations: BitBlts and standard VGA Framebuffer accesses. These two types of operations share resources in the hardware. This imposes certain requirements on driver software.

The state of all standard VGA registers is unchanged by BitBlt and extended register reads/writes with the exception of the CR22 data latch. The state of this register is undefined after a BitBlt.

The state of all extended registers is unchanged by VGA read/write operations.

Between a BitBlt operation and a VGA read/write operation, the software must ensure that no BitBlt operation is in progress by means of the Status register.

Before performing any VGA read/write operations, the software must ensure the Foreground register has the value FFFFFFFh and the background register has the value 00000000h. These are also the reset values of these registers.

Between a VGA write operation and a BitBlt operations, the software must ensure the VGA write pipeline is flushed by performing a VGA read operation.

13.10 GE OPERATIONS

13.10.1 PATTERN DATA

If the ROP register value specifies that pattern data is used in the computation of the destination results, then one row of the pattern data is read at the start of each scan line processed. This row of data is repeatedly applied to the result computation across scan line. The Pattern register points to the start of an 8-pixel-by-8-pixel color area that is aligned to the destination. The GE does not perform any horizontal alignment to the pattern data.

When the pixel depth is 3 Bytes, the least significant three bits of the Pattern register must indicate which Byte starts the pattern row. This field should be set to:

(Dst_X / 8) modulo 3

where Dst_X is the Byte address of the first 24-bit pixel in the destination row. (The same value that is written to the X field of the Dst_XY register).

Bitmap patterns are not directly supported. To use a bitmap pattern, first allocate off-screen frame buffer memory for a color version of the pattern. Then set up the GE to perform a Host-to-screen BitBlt with bitmap expansion into this allocated memory. The bitmap pattern is then written to the Data Port. The expanded pattern can now be used by pointing the Pattern register to the allocated memory.

13.10.2 BITMAP CONSIDERATIONS

Screen-to-screen and Host-to-screen operations can optionally expand single-bit-per-pixel bitmaps into color pixels. Each ‘1’ bit is replaced by the contents of the Foreground color register and each ‘0’ bit is replaced by the contents of the Background color register.

Bitmaps from the frame buffer (during Screen-to-screen BitBlts) must be aligned on a quad-word (64-bit) boundary. Bitmaps from the Host can be aligned on a double-word (32-bit) boundary. Leading bits of the bitmaps may be skipped by setting the least significant bits of the X field of the Src_XY register to the number of bits in the Byte to be ignored. When in bitmap expansion mode, the X field of the Src_XY address can be thought of as a bit address instead of a Byte address. For Host-to-screen bitmap expanded BitBlts only the least significant 5 bits of the Src_XY.X register are 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.

13.10.3 BITBLT OPERATIONS

Using the GE's BitBlt commands it is possible to implement the following six operations:

1) Rectangular Fill
2) Screen-to-screen BitBlt
3) Host-to-screen BitBlt
4) Packed Text
5) Microsoft Font Text
6) Line Segments

13.10.4 RECTANGULAR FILL

A rectangular fill operation is used to fill rectangular areas in the frame buffer with solid or patterned colors. The function performed during the fill operation is:

ROP ((Background), (Pattern), (Destination)) -> (Destination)

The specified rectangle is filled with the contents of the Background color register, the pattern data and the existing destination contents, as modified by the ROP. The CPU provides the rectangle's upper-left (Dst_XY) coordinates, the width and the height of the rectangle. Destination and pattern data can be anywhere in the frame buffer. ROP may be any of the 256 standard raster operations.

The Rectangular Fill operation is optimized to run at the memory bandwidth.

To perform a rectangular fill (except for the last write, order is unimportant):

1) Set the starting base address of the screen or region into Dst_base
2) Set the pitch of the screen or region into Dst_pitch
3) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to '0',
4) Set Width.Width to one less than the width of the rectangular area, in Bytes
5) Set Height.Height to one less than the height of the rectangular area, in Bytes
6) Set ROP.ROP to the raster operation, ROP.Expand to "0", and ROP.Transparent OPAQUE
6) Set ROP.Source to CONSTANT_FILL
7) Set the Background color register to the 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. 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.

13.10.5 SCREEN-TO-SCREEN BITBLT

The Screen-to-screen BitBlt operation is used to copy data from one rectangle in the frame buffer (either on-screen or off-screen areas) to another with the identical geometry. The pixel depth of the source region must match that of the destination region, or it may be a bitmap (if bitmap expansion is specified by setting ROP.Expand).

The function performed during the BitBlt operation is:

ROP ((Source), (Pattern), (Destination)) -> (Destination)

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

* *Source region is below the destination:*

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

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

* *Source region is above the destination:*

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

> the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register should be “1”

The Source, destination and pattern data can be anywhere in the frame buffer. The ROP may be any one of the 256 standard raster operations.

To perform a Screen-to-screen BitBlt (except for the last write, order is unimportant):

1) Set the starting base address of the destination screen or region into Dst_base

2) Set the pitch of the destination screen or region into Dst_pitch

3) Set the starting base address of the source region into Src_base

4) Set the pitch of the source region into Src_pitch

5) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to ‘0’ or to ‘1’, depending upon direction of the BitBlt

6) Set Width.Width to one less than the width of the destination rectangular area, in Bytes

7) Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values

8) Set ROP.Source to SCREEN

9) If Color Transparency mode is set, the set Background color register to the color to be made transparent

10) Set the Pixel_depth for the number of Bytes-per-pixel

11) If a pattern is to be applied, set the Pattern register with the starting address of the pattern data

12) If ROP.Expand is set, then also set the Foreground and Background color registers

13) Write the starting coordinates of the source area to be copied into the Src_XY register

14) Write the starting coordinates of the destination area into the Dst_XY register. This write starts the BitBlt.

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

The implementation of this operation is performance optimized and can drive the DRAM 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.

13.10.6 HOST-TO-SCREEN BITBLT

The Host-to-screen BitBlt is used to copy data from the Host CPU to the frame buffer (either on-screen or off-screen areas). Note that if the CPU has built a rectangle in the frame buffer memory area with the Host data, then the Screen-to-Screen BitBlt operation can be used instead of this operation.

The pixel depth of the Host data must match that of the Destination region, unless it is a bitmap (if bitmap expansion is specified).

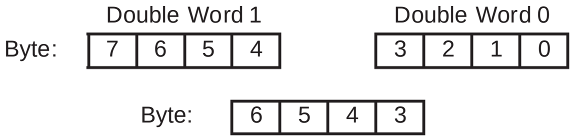
The function performed during the BitBlt is:

ROP ((Host), (Pattern), (Destination)) -> (Destination)

The host area data is supplied by the CPU, which writes its data into the Data Port. The destination and pattern data can be anywhere in the frame buffer.

ROP may be any one of the 256 standard raster operations.

The CPU specifies the number of least significant Bytes of the first double-word that should be discarded, via the least significant 3 bits of the X field in the Src_XY register. The GE then merges Bytes of two double-words at a time, in order to build a double-word to operate on. For example, if the X field was set to 3, then the last Byte of the first double-word and the first three Bytes of the second double-word would be combined to form the first Host data double-word:



The CPU must provide the number of words required for Height * Width pixels. At the end of a scan line, the GE will discard the excess Host Bytes or bits that may be left in the last double-word and advance to the next scan line, unless Src_pitch is set to 0. In this case, data for adjacent scan lines are contiguous in the host data stream.

To perform a Host-to-screen BitBlt (except for the last write, order is unimportant):

1) Set the starting base address of the destination screen or region into Dst_base

2) Set the pitch of the destination screen or region into Dst_pitch

3) Set Src_pitch to 0 if packed source data or to a non-zero value, otherwise

4) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, and Pixel_depth.Y_dst_dir to ‘0’

5) Set Width.Width to one less than the width of the destination rectangular area, in Bytes

6) Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values

7) Set ROP.Source to HOST

8) If Color Transparency mode is set, set the Background color register to the color to be made transparent

9) Set the Pixel_depth for the number of Bytes-per-pixel

10) If a pattern is to be applied, set the Pattern register with the starting address of the pattern data

11) If ROP.Expand is set, then also set the Foreground and Background color registers

12) Set Src_XY to indicate alignment of the Host data (only the least 3 significant bits are important)

13) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.

14) The image data is then written to the GE’s Data Port, if the GE is unable to accept additional data the CPU will be held off (invisibly to the software)

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

13.10.7 PACKED TEXT

The Packed Text operation is used to efficiently expand packed bitmap fonts into full color representations in the frame buffer (either on-screen or off-screen areas). This operation is implemented as a Host-to-screen BitBlt with bitmap expansion and packed source data. The next section discusses how to handle Microsoft Font Text operations.

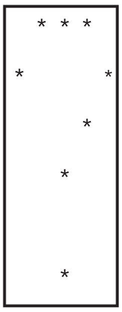
The function performed during the Packed Text operation is:

ROP ((Host), (Pattern), (Destination)) -> (Destination)

The Host packed bitmap data is supplied by the CPU via writes to the Data Port and is organized as double-words containing 32 bits of information. Each bit corresponds to a pixel. This data is expanded into Background and Foreground colors, unless the bitmap expansion transparent mode is on. If the transparent mode is set, then Host data bits of ‘0’ suppress any changes to the corresponding destination pixels. The Destination and Pattern can be anywhere in the frame buffer.

ROP may be any one of the 256 standard raster operations.

In a standard bitmap, the start of each scan line is aligned to a pitch-specified boundary. This is acceptable for wide bitmaps, however text font bitmaps are usually not very wide. To increase the amount of information provided to the GE per Host write, the Text operands are bit-packed. Each 32-bit write contains only useful font data, except possibly for the trailing bits of the last write. For example, question mark character might appear in a fictitious font as:



Or in Binary form:

0	1	1	1	0
1	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	0	0	0
0	0	1	0	0

This would appear in memory as in Table 13-12

Table 13-12Bit representation

top line					bottom line
01110	10001	00010	00100	00000	00100

Increasing memory addresses ------>

Breaking this up into Bytes see Table 13-13

Table 13-13Byte representation

	01110100	01000100	01000000	000100XX
Byte	0	1	2	3

Breaking this up into a double-word as inTable 13-14 :

Table 13-14Double word representation

	000100XX	01000000	01000100	01110100
Byte	3	2	1	0
Word	0		1	

In this example, the entire character bitmap fits into a single 32-bit double-word. This is a big savings over having to possibly send one 32-bit double-word for each font row. Note that 2 bits of don't cares exist at the (top) end of the double word. Since this character is 5 bits wide and 6 lines high, it only needs 30 bits of storage. The remaining 2 bits will not be displayed.

After setting up the registers, the CPU writes the Host data, in 32-bit quantities, to the Data Port.

If a Pattern is applied to the text operation, a row of the pattern data will be read at the start of each character scan line.

To perform a Packed Text BitBlt (except for the last write, order is unimportant):

1) Set the starting base address of the destination screen or region into Dst_base

2) Set the pitch of the destination screen or region into Dst_pitch

3) Set the ROP.Packed to 1 for packed source data

4) Set Pixel_depth.X_dir, Pixel_depth.Y_src_dir, Pixel_depth.Y_dst_dir to ‘0’

5) Set Height to the height of the destination character, in scan lines

6) Set ROP.ROP, ROP.Expand and ROP.Transparent to the appropriate values

7) Set ROP.Source to HOST

8) Set the Pixel_depth for the number of Bytes-per-pixel

9) If a pattern is to be applied, set the Pattern register with the starting address of the pattern data

10) set the Foreground and Background color registers

11) Set Src_XY to indicate alignment of the Host data (only the least 5 significant bits are important)

12) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.

13) The character font data is then written to the GE's Data Port, if the GE is unable to accept additional data the CPU will be held off (invisibly to the software)

To draw the next character, its starting address (taking into account inter-character spacing) is written to the Dst_XY register, along with that character's width encoded into the address of the Dst_XY register. This write can be done even if the GE is busy, as the Destination and Width registers are double buffered. The CPU then writes all the bitmap data that corresponds to the second character, the third Dst_XY/Width, the third bitmap data, etc.

Frequently, some of these parameters will already be set up from prior operations, such as Dst_pitch, Foreground color, Background color, and Pixel_depth. These will not need to be written each time.

13.10.8 MICROSOFT FONT TEXT

Microsoft fonts (consisting of 8-bit strips of a character) can be handled as simple 8-pixel-wide Host-to-screen BitBlts with bitmap expansion, but no packed data. The last strip of a character is handled in a different manner. The background color for the last strip is first filled into its rectangular area. Then the strip data is drawn in transparent mode with the unused bits filled with zeros.

13.10.9 LINE SEGMENTS

The line segment operations are used to draw horizontal or vertical line segments. The segments are runs of pixels that start from a specified coordinate address (via the Dst_XY register) and whose length is specified in the address used when writing to the Dst_XY register.

The function performed during the line draw is:

ROP ((Background), (Pattern), (Destination)) -> (Destination)

ROP may be any one of the 256 standard raster operations.

Simple and complex curves can be efficiently drawn. The software on the CPU must generate all points or scan lines to be drawn and then use the GE to draw the line segments.

Two different types of line segments are supported: horizontal and vertical. For horizontal line segments, the Height register should be programmed to ‘0’, to indicate a single pixel high line. The length of the line segment (in Bytes) will then be stored into the Width register when the Dst_XY register is written to. (The length is encoded into the Count field of the Dst_XY register’s address.) For vertical lines, the Width register should be programmed to one less than the number of Bytes per pixel, to indicate a single pixel wide line. The length of the line segment is stored into the Height register when the Dst_XY register is written to.

It is possible to draw thicker line segments, by programming the Height register (for horizontal segments) or the Width register (for vertical segments) to other values.

For horizontal line segments, the X_dir, Y_src_dir and Y_dst_dir fields of the Pixel_depth register must be set to “0”.

The Background color register should be set to the color of the line segment to be drawn.

13.11 CURSOR SUPPORT

The GE supports a 64x64x2 cursor. The cursor is actually two 64x64x1 arrays: an AND array and an XOR array. For any given pixel that is within the cursor's active region, the displayed pixel depends on the frame buffer's pixel, the AND array value, the XOR array value and the Cursor_color0 and Cursor_color1 registers as shown in Table 13-15 :

Table 13-15Cursor Arrays

AND Value	XOR Value	Displayed Pixel
0	0	Cursor_color0
0	1	Cursor_color1
1	0	Frame Buffer Pixel
1	1	Inverted Frame Buffer Pixel

The AND array is stored in off screen memory, starting at Cursor. The XOR array is stored in off screen memory starting at (Cursor + 512). Two 64-bit on-chip registers hold one scan line of each of these arrays. Before a scan line that possibly includes a cursor is displayed, these two registers are loaded from the appropriate off-screen locations.

Note that for 8-bit and 16-bit pixel depths, the above cursor operation is performed AFTER the data has been expanded by the color look-up-table (LUT). Thus, the Inverted Frame Buffer Pixel, is the complement of the full-color pixel that would otherwise be displayed.

The cursor address (Cursor_XY) refers to the upper-left-hand corner of the cursor and specifies the distance, in pixels, from the upper-left-hand corner of the screen. So, if the cursor address were to be set to (0,0), then the entire cursor could be displayed in the upper-left-hand corner of the screen. The cursor's active region thus may extends from:

(Cursor_XY.X, Cursor_XY.Y)

to

(Cursor_XY.X + 63, Cursor_XY.Y + height - 1)

as controlled by the Cursor Height register (CR29).

Note: To suppress cursor display, enter one more than the number of display scan lines into the Y field.

13.11.1 CURSOR HEIGHT REGISTER
CR29 3X5H INDEX 29 (RW)

Bit 7 **Cursor XOR Pre/Post Look Up Table.** When this bit is set to one, the graphics 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.

This register is set to 00h after reset.

Programming notes

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

13.11.2 CURSOR COLOR 0 REGISTER A
CR2A 3X5H INDEX 2A (RW)

Bits 7-0 **Cursor Color 0 Red.** These bits are the red component of cursor color 0.

This register is undefined after reset.

13.11.3 CURSOR COLOR 0 REGISTER B
CR2B 3X5H INDEX 2B (RW)

Bits 7-0 **Cursor Color 0 Green.** These bits are the green component of cursor color 0.

This register is undefined after reset.

13.11.4 CURSOR COLOR 0 REGISTER C

CR2C 3X5H INDEX 2C (RW)

Bits 7-0 **Cursor Color 0 Blue.** These bits are the blue component of cursor color 0.

This register is undefined after reset.

13.11.5 CURSOR COLOR 1 REGISTER A

CR2D 3X5H INDEX 2D (RW)

Bits 7-0 **Cursor Color 1 Red.** These bits are the red component of cursor color 1.

This register is undefined after reset.

13.11.6 CURSOR COLOR 1 REGISTER B

CR2E 3X5H INDEX 2E (RW)

Bits 7-0 **Cursor Color 1 Green.** These bits are the green component of cursor color 1.

This register is undefined after reset.

13.11.7 CURSOR COLOR 1 REGISTER C

CR2F 3X5H INDEX 2F (RW)

Bits 7-0 **Cursor Color 1 Blue.**These bits are the blue component of cursor color 1.

This register is undefined after reset.

13.11.8 GRAPHICS CURSOR ADDRESS REGISTER 0

CR30 3X5H INDEX 30 (RW)

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

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

This register is undefined after reset.

Programming notes

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

13.11.9 GRAPHICS CURSOR ADDRESS REGISTER 1

CR31 3X5H INDEX 31 (RW)

Bits 7-0 **Cursor AND Address Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the cursor's AND mask. For a discussion of DRAM linear addresses, see section 5.4.5.

This register is undefined after reset.

13.11.10 GRAPHICS CURSOR ADDRESS REGISTER 2

CR32 3X5H INDEX 32 (RW)

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

Bits 2-0 **Cursor AND Address Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the cursor's AND mask. For a discussion of DRAM linear addresses, see section tbc.

This register is undefined after reset.

13.12 GRAPHICS CLOCK REGISTERS

The GCLK is used to the Graphics Engine operations

13.12.1 GCLK CONTROL REGISTER 0 GCLK00 INDEX 40

Bit 7 This is unused.

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

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

This register defaults to 0x5B at reset. This value when combined with the default value of the other half of this pair results in a graphics clock of 80.05 MHz assuming 14.318 MHz oscillator clock as the reference input.

13.12.2 GCLK CONTROL REGISTER 1 GCLK01 INDEX 41

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

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

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

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

13.13 UPDATE HISTORY FOR GRAPHICS ENGINE CHAPTER

The following changes have been made to the Graphics Engine Chapter.

Section	Change	Text
13.12	Added	GCLK register descriptions

The following changes have been made to the Graphics Engine Chapter.

Section	Change	Text
13.1	Replaced	“~~above~~” With “between”
13.2	Replaced	“~~displayed~~” With “displayed in order to be compatible with future versions of the GE.”
13.2	Removed	“~~Except for the VGA registers~~,”
13.2	Removed	“~~VGA registers may be written to via byte, word or double-word accesses~~.”
14.6	Removed	~~“The GE fully supports VGA operations. The standard two read modes and four write modes of the VGA frame buffer are implemented. While, there is no “GE mode” bit that places the drawing operations in a VGA mode, the Foreground and Background color registers must be set to all ‘1’s and all ‘0’s, respectively, before issuing standard VGA write operations.”~~
13.5	Removed	~~“Addressing for reads and writes~~ ~~Host addresses are mapped into display memory addresses according to the VGA addressing mode selected.~~ ~~Multiple byte reads and writes are broken into single byte operations by the GE. For multi-byte reads, all of the bytes are sequenced through by the GE which performs each read and store the resulting byte in correct byte of an VGA read register. After the last read is done, the GE will provide the CPU with the requested data.~~ ~~For multi-byte writes, all the GE sequences through the CPU data bytes, performing a 32-bit write to memory for each byte. No optimization is done by the GE to merge 32-bit Dword writes. (Such optimization would only be possible in the Chain-4 addressing modes.)”~~
13.5.1	Removed	~~“Read Mode 0~~ ~~Read data is read into the GE. The byte selected by the read-map-select register is replicated into all four byte positions to form the resulting Dword, which is presented to the CPU.”~~
13.5.2	Removed	~~“Read Mode 1~~ ~~Read data is read into the GE. The color-compare assembler takes the appropriate 4 bit sets from the output and groups them into one byte. This byte is then replicated into all four byte positions to form the resulting Dword, which is presented to the CPU.”~~
13.5.2	Replaced	“~~pixels that~~” With “pixel which either matches the value or do not”
13.5.2	Replaced	“~~pixels that~~” With “pixel that either matches the value or does not”

Update History for Graphics Engine chapter

Section	Change	Text
14.5.1.3	Removed	~~“Write Mode 0~~ ~~One of the bytes of the write data, depending upon the byte address, is selected and rotated 0-7 bits. The resulting byte is then output as all four bytes of the result. In parallel, the four set/reset mask bits are each expanded to a byte. The four set/reset enable bits select either the rotated write data or the expanded set/reset bytes, on a byte-by-byte basis. The VGA ROP is then applied to the selected data and the latched read data. A bit mask controls which bits of all bytes are altered and a map mask controls which bytes are written.”~~
14.5.1.4	Removed	~~“Write Mode 1~~ ~~Write mode 1 is a subset of Write Mode 0. No CPU-supplied write data is used. The read data latched from a previous read operation is written. The bit mask is disabled. The map-masks are implemented as they are for Write Mode 0.”~~
14.5.1.5	Removed	~~“Write Mode 2~~ ~~Write mode 2 is similar to write mode 0, except that only data from the bit-alignment is used. In the bit alignment unit, the four least significant bits of the current byte are selected. Since the bit ordering is different for this operation than Windows bitmaps, the stutterer/selector must reverse the order of the four bits. Bit-expander expands the bits into the FG or BG, which software will have set to all ‘1’s or ‘0’s, respectively.”~~
14.5.1.5	Removed	~~“The bit-mask, VGA logic-operation, and map-masks are implemented in the same way as Write Mode 0.”~~
14.5.1.6	Removed	~~“Write Mode 3~~ ~~Write mode 3 is similar to write mode 0, except that the CPU data byte is rotated and anded with the contents of the bit mask register to form the bit mask. The set/reset mask bits are expanded to one byte each and used regardless of the state of the set/reset enable bits.”~~
13.6	Replaced	“~~A pixel’s X coordinate is usually expressed as an unsigned byte quantity, the number of bytes from the left edge of a scan line. If the X_dir field of the Pixel_depth register is ‘0’, advancing from left-to-right, then X points to the least-significant byte of the starting pixel. If the X_dir field of the Pixel_depth register is ‘1’, advancing from right-to-left, then X points to the most-significant byte of the starting pixel~~” With “A pixel’s X coordinate is usually expressed as an unsigned Byte quantity, the number of Bytes from the left edge of a scan line. If the X_dir field of the Pixel_depth register is ‘0’, advancing from left-to-right, then X points to the least-significant Byte of the starting pixel. If the X_dir field of the Pixel_depth register is ‘1’, advancing from right-to-left, then X points to the most-significant Byte of the starting pixel.”
13.6.1	Replaced	“~~Cmd is one of the following~~:” With “ This is shown in Table 13-5 CMD operations:”
13.7	Removed	~~“Except for the VGA registers, Writes, must be done using double-word (32-bit) transfers. There are a few extended registers for which only the lower 16 bits are defined. For these, 16-bit writes are acceptable. VGA registers may be written to via byte, word or double-word accesses.”~~
13.8	Replaced	“~~accepting the short possible periods of being held~~.” With “reducinghold period as much as possible.”
13.8.1	Replaced	“~~enough~~” With “exact amount of”
13.8.1	Added	“command”
13.9.3	Replaced	”**~~Top of Data FIFO~~”** With **“Data_Port”**

Section	Change	Text
13.9.3	Added	“Destination Operand Base Address Register Regoffset = 018h (DST_Base)Destination Operand Base Address Register Regoffset = 018h (DST_Base)”
13.9.6	Added	“implemented”
13.9.6	Added	The complete addressing of this register is described in Section 9.5.3.1. Command Initiation.
13.9.6	Replaced	“~~The address of this register is described in~~ Section 9.5.5.” With “The contents of this register are not altered by drawing operations.”
13.9.7	Added	The contents of this register are defined to be FFFFFFFFh.
13.9.10	Replaced	“~~to ‘1’~~ if “ With “when set to ‘1’”
13.9.14	Removed	“~~corner which is the~~ “
13.9.14	Replaced	“~~Bits 32-16 **Src_Y.** The unsigned Y coordinate of the starting corner of the source operand~~.” With “Bits 32-16 **Src_Y.** The source operand starting corner of the unsigned Y coordinate.”
13.9.14	Replaced	“Bits 15-0 **Src_X.** The unsigned X location of the starting corner of the source operand. When bitmap expansion is not enabled, this is a Byte address. When in bitmap expansion is enabled, this is a bit address.” With “Bits 15-0 **Src_X.** The source operand starting corner of the unsigned X location. When bitmap expansion is not enabled, this is a Byte address. When in bitmap expansion is enabled, this is a bit address.”
13.10.4	Added	“4) Set Width.Width to one less than the width of the rectangular area, in Bytes”
13.10.4	Replaced	“~~5) Set Width. Width to one less than the width of the rectangular area, in bytesBytes~~” With “5) Set WidthHeight.Width Height to one less than the width height of the rectangular area, in bytesBytes”
13.10.4	Replaced	“~~10) Write the upper-left coordinates of the area to be filled into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address~~.” With “10) Write the upper-left coordinates of the area to be filled into the Dst_XY register.”
13.10.5	Replaced	“~~14) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address~~.” With “14) Write the starting coordinates of the destination area into the Dst_XY register.”
13.10.6	Replaced	“~~13) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Height-specified address alias and the height of the rectangular region is encoded in the Count field of the address, starting the BitBlt~~.” With “13) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt.”

Section	Change	Text
13.10.7	Replaced	"~~12) Write the starting coordinates of the destination area into the Dst_XY register is written to using the Width-specified address alias and the width of the destination character (in bytes) in the Count field of the address, starting the BitBlt~~." With "12) Write the starting coordinates of the destination area into the Dst_XY register. This writes the BitBlt."
13.11	Replaced	"~~(Cursor_XY.X + 63, Cursor_XY.Y + 63)~~" With "(Cursor_XY.X + 63, Cursor_XY.Y + height - 1)"
13.11.1	Added	"Cursor Height Register CR29 3X5h Index 29 (RW) Bit 7 **Cursor XOR Pre/Post Look Up Table.** When this bit is set to one, the graphics cursor XOR operation is performed before the look up table. The default behavior, when this bit is set to zero, is for the XOR operation to happen after the look up table. This is correct for 15, 16, 24 bit per pixel modes but not 8bpp. Bits 6-0 **Cursor height.** This field represents the vertical extent of the graphics cursor in scan lines. Setting this to zero effectively turns the graphics cursor off. Values greater than 40h (decimal 64) are meaningless and produce unpredictable results. Note: there is no cursor width register - the width is always 64 pixels. If a narrower cursor is required, pad the bitmap on the right with transparent cursor color (pad the AND plane with '1's on the right and the XOR plane with '0's). This register is set to 00h after reset."
13.11.2	Added	Cursor Color 0 Register A CR2A 3X5h Index 2A (RW) Bits 7-0 **Cursor Color 0 Red.** These bits are the red component of cursor color 0. This register is undefined after reset.
13.11.3	Added	Cursor Color 0 Register B CR2B 3X5h Index 2B (RW) Bits 7-0 **Cursor Color 0 Green.** These bits are the green component of cursor color 0. This register is undefined after reset.
13.11.4	Added	Cursor Color 0 Register C CR2C 3X5h Index 2C (RW) Bits 7-0 **Cursor Color 0 Blue.** These bits are the blue component of cursor color 0. This register is undefined after reset.

Section	Change	Text
13.11.5	Added	Cursor Color 1 Register A CR2D 3X5h Index 2D (RW) Bits 7-0 **Cursor Color 1 Red.** These bits are the red component of cursor color 1. This register is undefined after reset.
13.11.6	Added	Cursor Color 1 Register B CR2E 3X5h Index 2E (RW) Bits 7-0 **Cursor Color 1 Green.** These bits are the green component of cursor color 1. This register is undefined after reset.
13.11.7	Added	Cursor Color 1 Register C CR2F 3X5h Index 2F (RW) Bits 7-0 **Cursor Color 1 Blue.**These bits are the blue component of cursor color 1. This register is undefined after reset.
13.11.8	Added	Graphics Cursor Address Register 0 CR30 3X5h Index 30 (RW) Bits 7-1 **Cursor AND Address Bits 15-9.** These bits represent bits 15-9 of the DRAM linear address of the cursor's AND mask. The cursor's XOR mask begins at this address + 512. This memory must be aligned on a 1 KByte boundary. For a discussion of DRAM linear addresses, see section tbc. Note that the cursor bitmap is ordered such that the top left hand corner of the cursor is represented by bit 7 of the Byte addressed by this field (AND) and bit 7 of the Byte at 512 plus this address (XOR plane). The next pixel right is represented by bit 6 of these Bytes and so on until the bottom right hand pixel is represented by bit 0 of the Byte locat ed at this address plus 511 (AND) and bit 0 of the Byte at 1023 plus this address. Bit 0 *Reserved.* This bit should be written as zero. This register is undefined after reset.

Update History for Graphics Engine chapter

Section	Change	Text
13.11.9	Added	Graphics Cursor Address Register 1 CR31 3X5h Index 31 (RW) Bits 7-0 **Cursor AND Address Bits 23-16.** These bits represent bits 23-16 of the DRAM linear address of the cursor's AND mask. For a discussion of DRAM linear addresses, see section 5.4.5. This register is undefined after reset.
13.11.10	Added	Graphics Cursor Address Register 2 CR32 3X5h Index 32 (RW) Bits 7-3 *Reserved.* These bits should be written as zero. Bits 2-0 **Cursor AND Address Bits 26-24.** These bits represent bits 26-24 of the DRAM linear address of the cursor's AND mask. For a discussion of DRAM linear addresses, see section tbc. This register is undefined after reset.

14. VIDEO CONTROLLER

14.1 INTRODUCTION

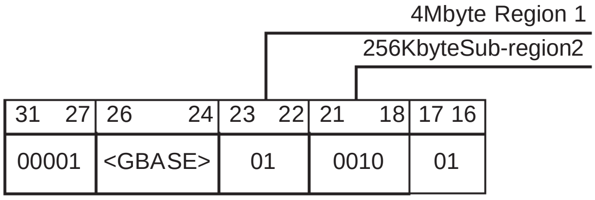
The STPC Industrial controls video signal input, buffering and output through the Video Controller. The Video Input and buffering is controlled by the Video Input Port, while the Video Output Port controls the video output in several standards.

14.1.1 The Video Pipeline Registers

The video input and display is controlled through the Video Pipeline registers. These registers provide the settings for the display buffer areas, filter control, color mixing and color space mixing.

14.2 VIDEO PIPELINE REGISTERS

The Video Pipeline registers, similar to the extended graphics (non-VGA) registers, are located in the 4-MByte memory-mapped registers region of the 16-MByte memory space occupied by the Graphics Controller. The Video Pipeline registers are located at the 256-KByte wide sub-region 2. The figure below shows the address format for the Video Pipeline registers.



31 27	26 24	23 22	21 18	17 16
00001	<GBASE>	01	0010	01

All registers can be read with accesses of any width. The CPU can read any register via byte (8-bit), word (16-bit), or double-word (32-bit) accesses. Writes must be done using double-word (32-bit) transfers.

14.2.1 Source Specification registers

14.2.1.1 Video_Src_Base
Index 00h

This register specifies the DRAM linear starting address of the source image, aligned to an 8 byte boundary. This address may specify either the top left corner or bottom left corner, depending on the state of the Y_Vid_Src_dir bit in the Video_Src_Pitch register.

This register is double buffered, the active register is only updated during vertical blanking.

Bits 31-22 *Reserved.*

Bits 21-0 Base linear address of the Video Source Image. Lower address bits 2-0 are reserved and when read return a value of ‘0’.

This register is cleared to zero after reset

14.2.1.2 Video_Src_Pitch Index 04h

This register contains the Video_Src_pitch field, which specifies the number of bytes which must be added to the address of a pixel on one line of the video source image to compute the address of the corresponding pixel on the line below.

This register also contains the Y_Vid_Src_dir bit which specifies the Y direction in which the Video Source Image should be read, and the Video_Color_fmt field which defines the the 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 10-0 Specifies the amount to add to the current address to get to the address of the corresponding pixel in the next line. Lower address bits 2-0 are reserved and when read return a value of ‘0’.

This register is cleared to zero after reset

14.2.1.3 Vid_Src_dim Index 08h

This register contains the dimensions of the Video Source Image relative the the starting corner.

This register is double buffered. The active 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 15-10 *Reserved.*

Bits 9-0 **dX,** the width in pixels, of a line .

14.2.1.4 CRTC_Burst_length Index 0Ch

This register contains the CRTC low water mark and burst length.

Bits 31-24 **Delta low water mark, crtc_dlwm.** Together with crtc_dt and crtc_lwm, this field defines a variable low water mark. When the video window starts, the CRTC low water mark is set to crtc_lwm. After that time, for every crtc_dt*8 pixels’ time elapsed, the low water mark will be incremented by crtc_dlwm bytes.

Since this field is represented as a 2’s complement number, setting bit 31 results in a low water mark which is a decreasing function of time. A decreasing or constant function will be the normal mode of operation of the CRTC low water mark during the video window.

Note that this CRTC low water mark is distinct from the one described in CR1B. This one is valid during the video windows only.

For normal CRTC operation (scanlines or pixels outside the video window), the pertinent CRTC low water mark is specified by CR1B.

Guarantee of the CRTC ownership can be achieved by the Setting of this field to zero. This causes the CRTC low water mark to remain at a constant value of crtc_lwm.

Bits 23-16 **crtc_lwm,** the (initial) low water mark for the CRTC FIFO in bytes. During the video window, if the CRTC FIFO occupancy rises above the low water mark (defined as a function of time by crtc_dlwm and crtc_dt) and the video occupancy rises above the video low water mark then ownership of the system DRAM can be given back to the CPU.

The value of this register field can only be a multiple of eight (bits 18-16 are not writable and read as zeroes).

Bits 15-11 **crtc_dt, delta t.** See the description of crtc_dlwm above.

Bit 10 *Reserved*.

Bits 9-0 **Minimum CRTC burst length, crtc_bl.** This is the minimum number of bytes that will be sent in one transfer to fill the CRTC FIFO (during the active video window only). This value can only be a multiple of eight (bits 2-0 are not writable and read as zeroes) since pixels are fetched 8 bytes at a time.

This register is cleared to zero after reset

14.2.1.5 Vid_Burst_Length Index 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

14.2.2 Destination Specification Registers

14.2.2.1 Vid_Dst_XY Index 14h

This register contains the coordinates of the top left corner of the video window.

This register is double buffered. The active register is only updated during vertical sync.

Bits 31-26 *Reserved.*

Bits 25-16 **Y,** the Y coordinate of the top edge of the video window, relative to the display. The first display line is line 0.

Bits 15-11 *Reserved.*

Bits 10-0 **X,** the X coordinate of the left edge of the video window, relative to the display. The first pixel of each display line is pixel 0.

This register is cleared to zero after reset

14.2.2.2 Vid_Dst_dim Index 18h

This register contains the dimensions of the video window.

This register is double buffered. The active register is only updated during vertical sync.

Bits 31-26 *Reserved.*

Bits 25-16 **dY,** the height of the video window in screen lines - 1 is entered in this field.

Bits 15-11 *Reserved.*

Bits 10-0 **dX,** the width of the video window in screen pixels.

This register is cleared to zero after reset

14.2.3 Filter Control Registers

14.2.3.1 Horiz_Scale
Index 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:-

```
hpi = ((source_width/hdf_tmp) *
        4096)/dest_width
```

Note that the maximum value hpi is 4096.

This register is cleared to zero after reset

14.2.3.2 Vert_Scale
Index 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

14.2.3.3 Color space converter specification Index 2Ch

This register contains the control for the Color Space Converter.

Bits 31-1 *Reserved.*

Bit 0 **Color Space Converter Enable.** When set, YUV data is converted to RGB according to the formula:-

R = 1.164(Y - 16) + 1.591(V - 128)

G = 1.164(Y - 16) - 0.700(V - 128) - 0.336(U - 128)

B = 1.164(Y - 16) + 1.733(U - 128)

When clear, pixels are passed through unchanged.

This register is cleared to zero after reset

14.2.4 Video and Graphics mixing control registers

14.2.4.1 Mix Mode Register Index 30h

This register contains the Mix_Mode field which defines the method used to mix graphics and video.

Bits 31-2 *Reserved.*

Bits 1-0 **Mix_Mode,** controls the way in which graphics and video are mixed:-

1	0	Mix Mode
0	0	Video Window only. The video always appears in a rectangular window which is defined by the Destination Specification registers.
0	1	Video Window with Color Key. The Destination specification is further qualified by the Color Key register. Within the specified video window, if the graphics pixel (pre color palette) is equal to the value specified by the Color Key register, then the corresponding video pixel is displayed, otherwise the graphics pixel is displayed. Note that in 8-bit graphics modes, only Color_Key[7:0] are used in the comparison and in 16-bit graphics modes, Color_Key[15:0] are used.
1	0	Video Window with Chroma Key. The destination specification is qualified by the Chroma key registers. Chroma key compares each of the pixel components to independent ‘high’ and ‘low’ values (between limits compare). If all the selected components are between their limits, then the corresponding graphics pixel is displayed, otherwise the video pixel is 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

14.2.4.2 Color Key Register Index 34h

This register contains the color key value used in color keying mixing.

Bits 31-24 *Reserved.*

Bits 23-0 **Color_Key,** this value is compared to the graphics pixel to determine whether to display the video pixel in color key mode. When the graphics is operating in 8-bit per pixel mode, Color_key[7:0] is compared, when the graphics is operating in 16-bits per pixel, Color_Key[15:0] is compared and when the graphics is operating in 24-bits per pixel, Color_Key[23:0] is compared.

This register is cleared to zero after reset

14.2.4.3 Chroma Key Low Register Index 38h

This register contains the chroma key low limits, the component ignore bits and the color mode bit used in chroma keying mixing.

Bits 31-28 *Reserved.*

Bit 27 **Chroma key mode.**

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

14.2.4.4 Chroma Key High Register Index 3Ch

This register contains the chroma key high limits used in chroma keying mixing.

Bits 31-24 *Reserved.*

Bits 23-16 **ch2high,** the high limit against which component 2 is compared during chroma key 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
```

14.2.4.5 Status Register Index 40h

This register contains enable bit for the video scaler.

Bit 31 **vid_enable.** Setting the enable bit turns on the video scaler.

Bits 30-0 *Reserved.*

This register defaults to 00h after reset.

15. TFT INTERFACE

15.1 INTRODUCTION

The TFT interface works as a converter block which converts the signals from the CRT controller signals into control signals for a TFT (Thin Film Transistor) Flat Panel. This interface extends the capability of the VGA controller, allowing generation of the display on an active matrix LCD Flat Panel.The flat panel standards supported are given in Table 15-1.

The TFT interface supports a programmable panel size up to 1024 by 1024 pixels with programmable image positioning and sizing.

The TFT Interface also supports the PanelLink TM high speed serial transmitter externally for interfacing high resolution panel.

Table 15-1. Flat Panels supported

Video Standard	Supported interface
VGA, SVGA	9, 12, 18-bit (1 pixel/clock)
XGA, SXGA	2 x 9-bit (2 pixels/clock)

15.1.1. FUNCTIONAL DESCRIPTION

The TFT interface receives the basic timing informations through the HSYNC, VSYNC signals from the CRTC interface. It also receives the dot clock signal from the CRTC. All the internal timings of the TFT interface is based on this dot clock. The video data from the CRTC data output is also internally taken by the TFT Interface and is stored in a local FIFO buffer. The Flat Panel driver then generates the horizontal and vertical timing signals, data enable signal, data clock and the RGB data to the Flat Panel.

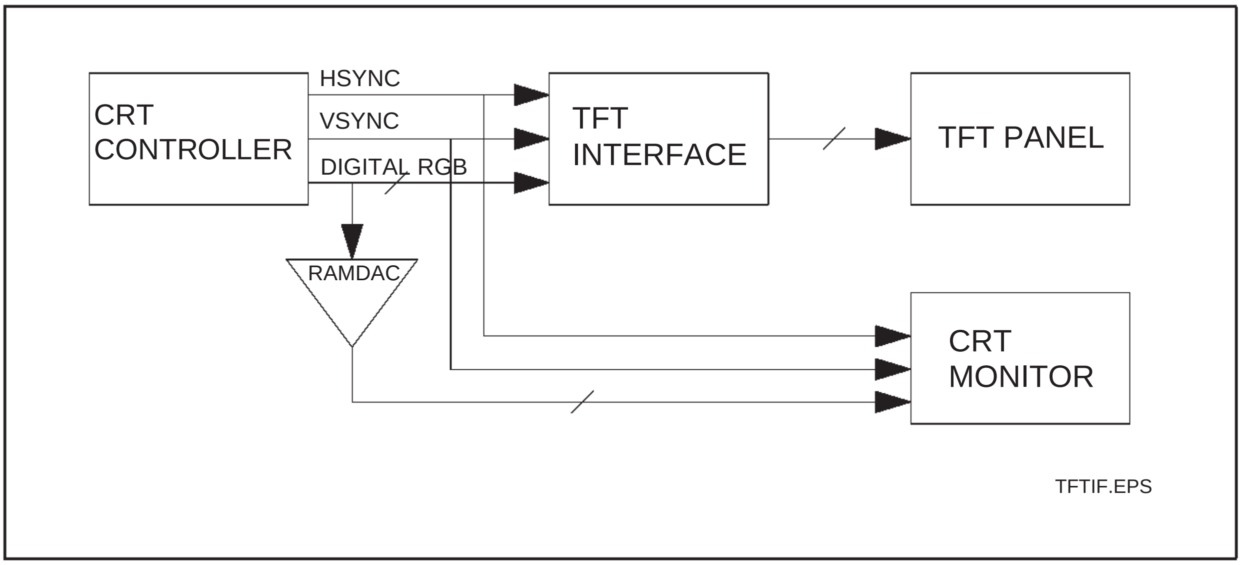
The built-in power control block provides PWM signals for brightness, contrast control and also for the switching of the panel power supply

15.1.2. TFT INTERFACE INTERFACE DESCRIPTION

Table 15-1. below shows how the TFT interface connects to a CRTC and a TFT panel.

.

Figure 15-1. TFT Interface



15.1.2.1 Programmable panel size

The height and width of the flat panel is defined in terms number of active pixels per scan line and number of active scan lines per frame respectively. These parameters are programmable through the module's configuration registers. The maximum supported panel size is 1024 by 1024.

15.1.2.2 Pixels per clock

TFT panels accept digital RGB data as input. For example, a TFT with 9 bits per pixel uses 3 bits to represent red, 3 bits for green and 3 bits for blue. There are two major standards for the data interface of the TFT flat panels (1) single pixel/clock in which on every dot clock data for a single pixel is sent to the panel and (2) two pixels/clock in which data for two pixels are sent to the panel. The advantage of two pixels per clock is that it requires a slower pixel clock which comes at the cost of larger data bus width. The TFT interface module supports 9, 12 and 18-bit interface in single pixel per clock mode and 2 x 9 bit interface for two pixels per clock mode.

The following table shows the colors disolayed by the Screen in single and two pixel per clock mode.

Table 15-2. Pixel Color Depth

1 Pixel/Clock	2 Pixel/Clock	Signal Type
R[5]	R[2]	odd
G[5]	G[2]	odd
B[5]	B[2]	odd
R[4]	R[2]	even
G[4]	G[2]	even
B[4]	B[2]	even
R[3]	R[1]	odd
G[3]	G[1]	odd
B[3]	B[1]	odd
R[2]	R[1]	even
G[2]	G[1]	even
B[2]	B[1]	even
R[1]	R[0]	odd
G[1]	G[0]	odd
B[1]	B[0]	odd
R[0]	R[0]	even
G[0]	G[0]	even
B[0]	B[0]	even

15.1.2.3 Programmable image positioning

Modern multi-sync CRT monitors automatically stretch images of different resolution such as to fill up the entire screen with the image. But unlike CRT monitors, TFT panels have a fixed resolution. So when a lower resolution image (say VGA 640 by 480 graphics) is displayed on a higher resolution TFT (say 800 by 600) the image covers only part of the screen. By default the image gets positioned at the top left corner of the screen. This is not very appealing from aesthetic stand point. One of the ways of improving the image appearance is to position the image at the center of the screen.

The TFT interface has configuration registers to position the image any where on the screen. Changing position of the image also requires reprogramming of some of the CRT Controller's timing registers. For compatibility reasons it is preferable to use the VGA Controller CRTC which supports shadowing of its timing control registers.

15.1.2.4 Programmable blank-space insertion in text modes

As described in the last section, by default lower resolution images cover only a part of the larger TFT panels. One way to fill up the entire panel (in text mode) is to introduce blank space between characters. To expand the image vertically, blank lines should be introduced between two character lines; this can be easily done by programming the CRTC character height control register. To expand the image horizontally, the TFT interface module can be programmed to introduce arbitrary number of blank pixels every 8 or 9 (or whatever is the character width) input pixels.

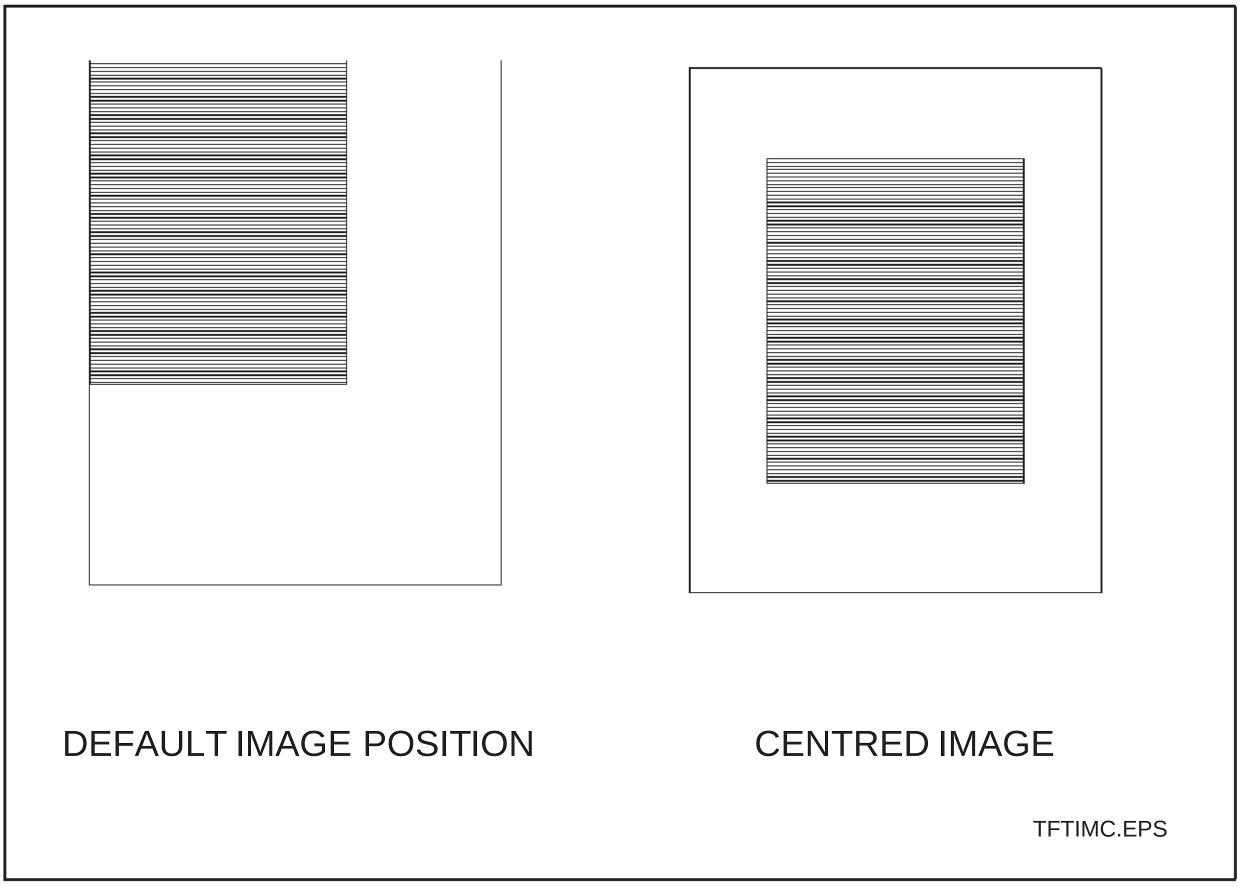
15.1.2.5 Image expansion in graphics mode

Another approach to fill up the larger TFT panels is done through image expansion by pixel replication. This approach works better with graphics mode rather than text modes.

In this method the image is expanded vertically by repeating last scan line every N lines. For example, with this method a 480 line image can be expanded to a 576 line image by repeating a scan line every 5 scan lines.

A similar approach is used to expand the image horizontally. The interface module can be programmed to repeat every Nth pixel M times. For example, with N=5 and M=1 a 640 pixel wide image can expand to 768 pixel wide image. Therefore by doing such vertical and horizontal expansion the entire screen can almost be filled up while displaying a 640 by 480 image on a 800 by 600 TFT panel.

Figure 15-2. Image centering



15.1.2.6 Brightness control using PWM

A programmable Pulse Width Modulated (PWM) signal is supported in the interface. This signal can be used to directly control the brightness or contrast of the flat panel without requiring external components. The total pulse width (period) and the duty cycle of this signal are individually programmable through the configuration registers.

15.1.2.7 PanelLink TM

PanelLink TM is a proprietary interconnect protocol defined by Silicon Image, Inc. It consists of a transmitter (Sil100) that takes parallel video/ graphics data from the host LCD graphics controller and transmits it serially at high speed to the receiver (SIL1101) which controls the TFT panel. The interface uses an adjustable voltage, high speed, serial, DC-balanced, transition minimized, differential transmission technique which reduces the line count, increases physical seperation between the panel and the host controller and most importantly lowers EMI.

The TFT interface is designed to support connection of its control signals to the PanelLink transmitter.

15.1.3. FLAT PANEL INTERFACE SIGNALS

DCLK, *Dot Clock Output.*

FPFRAME, *Vertical Sync. pulse Output.*

FPLINE, *Horizontal Sync. Pulse Output.*

DE, *Data Enable.*

R5-0, *Red Output.*

G5-0, *Green Output.*

B5-0, *Blue Output*.

ENAVDD *Enable VDD of Flat Panel.*

ENVCC *Enable VCC of Flat Panel.*

PWM *PWM Back-Light Control.*

.

Table 15-3. TFT interface configuration registers

Name	Index	Description
Input scan line active pixel count	0000h	Number of pixels in the active part of the input scan line
Input horizontal back porch	0001h	Input scan line back porch delay
FP horizontal sync width	0002h	Flat panel horizontal synchronizer active pulse width
FP horizontal back porch	0003h	Flat panel horizontal back porch
FP horizontal active pixel count	0004h	Number of pixels in the active part of the output scan line
FP horizontal front porch	0005h	Minimum flat panel horizontal front porch
FP vertical sync width	0006h	Flat panel vertical synchronizer active pulse width
FP vertical back porch	0007h	Flat panel vertical back porch
FP vertical active line count	0008h	Number of active scan lines
Interface control	0009h	Interface control register
PWM control	000Ah	Pulse width modulation control
Power control	000Bh	Power control for the Flat Panel
Blank red	000Ch	Red component of a blank pixel
Blank green	000Dh	Green componet of a blank pixel
Blank blue	000Eh	Blue component of a blank pixel
Polarity control	000Fh	Polarity control signal

15.2 TFT CONFIGURATION REGISTER ACCESS

The TFT module configuration registers are acessed by an index and data registers. The index and data registers are 16 bits wide, they are located in the I/O space and their addresses are programmable.

15.2.4. TFT BASE ADDRESS INDEX 0X1DH

The index register base address is part of the chipset register and programmed with the host configuration register 0x1Dh. This register is 16 bits wide but is accessed using two 8 bit I/O accesses.

The data register base address is the index address +4h.

Once the base address is programmed, the TFT registers are accessed as shown in the following sample coding;

```
mov dx, TFT Index base
mov ax, TFT Register number
out dx, ax
mov ax, TFT Register data
add dx, 4
out dx, ax
```

All TFT configuration registers are 16 bits wide.

15.3 CONFIGURATION REGISTER DESCRIPTION

There are 12 configuration registers in the module illustrated in Table 15-3.

15.3.5. INPUT SCAN LINE ACTIVE PIXEL COUNT INDEX 000H

Bits 15-12 *Reserved.*

Bits 11-0 **Active Pixel Count.** Number of pixels in the active part of an input scan line from the CRTC. A value of zero represents 2048 pixels in the active part of input scan line.

Power On Default: Undefined

15.3.6. INPUT HORIZONTAL BACK PORCH INDEX 001H

Bits 15-12 *Reserved*.

Bits 11-0 **Input Horizontal Front Porch.** Width (in units of input dot clock period) of the input scan line horizontal back porch from the CRTC. Horizontal back porch is the period between horizontal sync going inactive and start of the active display part of the next scan line. A value of zero represents a back porch 2048 dot clock wide.

Power On Default: Undefined

15.3.7. FLAT PANEL HORIZONTAL SYNC PULSE WIDTH INDEX 002H

Bits 15-12 *Reserved*.

Bits 11-0 **FPLINE Pulse Width.** Width (in units of input dot clock period) of the horizontal sync (FPLINE) pulse generated for the TFT panel.

A value of zero makes the pulse width 2048 dot clock wide.

Power On Default: Undefined

15.3.8. FLAT PANEL HORIZONTAL BACK PORCH INDEX 003H

Bits 15-12 *Reserved*.

Bits 11-0 **Horizontal Back Porch Width.** Width (in units of input dot clock period) of the flat panel scan line horizontal back porch of minus three. Therefore a value of one in this register would actually mean a back porch width of four clocks. Horizontal back porch is the period between horizontal sync going inactive and start of DE (Display Enable) of the next scan line. A value of zero represents a back porch 2051 dot clock wide.

Power On Default: Undefined

15.3.9. FLAT PANEL HORIZONTAL ACTIVE PIXEL COUNT INDEX 004H

Bits 15-12 Reserved.

Bits 11 to 0 **Horizontal Active Pixel Count.** The number of pixels in active part of flat panel scan line. Horizontal back porch is the period between horizontal sync going inactive and start of the active display part of the next scan line. A value of zero represents a back porch of 2048 pixels.

Power On Default: Undefined

15.3.10. FLAT PANEL HORIZONTAL FRONT PORCH INDEX 005H

Bits 15-12 *Reserved*.

Bits 11-0 **Minimum Horizontal Front Porch.** The minimum width (in units of input dot clock period) of the horizontal front porch of flat panel scan line. The actual front porch delay may be greater than this value. Normally the total horizontal width of flat panel scan line will be the same as that of the input scan line. So the horizontal front porch will be total input horizontal width minus the sum of FPLINE width, FP back porch and FP active pixel count. When the vertical scaling through line replication mode is enabled, the time of the active scan lines front porch width will be determined by this register value.

Horizontal back porch is the period between DE (Display Enable) going inactive and start of horizontal sync pulse. A value of zero represents a back porch 2048 dot clock wide.

Power On Default: Undefined

15.3.11. FLAT PANEL VERTICAL SYNC WIDTH INDEX 006H

Bits 15-12 *Reserved*.

Bits 11-0 **FPFRAME Pulse Width.** The active pulse width (in units of horizontal scan line input period) of the vertical sync signal (FPFRAME). A Zero value represents a width of 2048 dot clock period.

Power On Default: Undefined

15.3.12. FLAT PANEL VERTICAL BACK PORCH INDEX 007H

Bits 15-12 *Reserved*.

Bits 11-0 **Vertical Back Porch Width.** The vertical back porch width (in units of horizontal scan line input period) for flat panel frames. A Zero value represents a width of 2048 lines. The vertical back porch is the period between FPFRAME going inactive and start of the first active scan line of the next frame.

Power On Default: Undefined

15.3.13. FLAT PANEL ACTIVE LINE COUNT INDEX 008H

Bits 15-12 *Reserved*.

Bits 11-0 **Active Line Count.** The number of scan lines in the active part of flat panel frames. A Zero value represents a width of 2048 lines.

Power On Default: Undefined

15.3.14. FLAT PANEL INTERFACE CONTROL INDEX 009H

Bit 15 **LP, Line Polarity.** Polarity of the flat panel horizontal sync signal (FPLINE). Setting this bit to ’1’ generates an active high FPLINE signal.

Bit 14 **FP, Frame Polarity.** Polarity of the flat panel vertical sync signal (FPFRAME). Setting this bit to ’1’ generates an active high FPFRAME signal.

Bit 13 **VP, Vertical Synch Polarity.** Polarity of the vertical sync input signal (VSYNC) from the CRTC. Setting this bit to ‘1’ indicates an active high VSYNC signal.

Bit 12 **HP, Horizontal Synch Polarity.** Polarity of the horizontal sync input signal (HSYNC) from the CRTC. Setting this bit to ’1’ indicates an active high HSYNC signal.

Bit 11 **EV, Vertical Scaling.** Setting this bit to ’1’ enables vertical scaling. Setting it to zero disables this feature.

Bit 10 **DP, Pixel Clock Data Format.** Setting this bit to ’1’ indicates two pixel/clock data format, otherwise the data format is assumed single pixel/ clock

Bit 9 **IB, Blank Pixel Insert.** If this bit is set to ‘1’ then during the active part of the output scan line, after every N pixels (indicated by real pixel period value), M blank pixels (indicated by stretch pixel period value) will be inserted. Setting it to ‘0’ disables this feature.

Bit 8 **RP, Pixel Repeat.** If this bit is set to ‘1’ then during the active part of the output scan line, every Nth pixel (indicated by real pixel period value), will be replicated M times (indicated by stretch pixel period value). Setting it to ‘0’ disables this feature.

Bits 7-4 **Stretch Pixel Period.**

Bits 3-0 **Real Pixel Period.**

Power On Default: F000H

15.3.15. PWM CONTROL INDEX 00AH

Bits 15-12 *Reserved*.

Bits 11-4 **Duty Cycle.** The period for which the PWM signal will be active. If this value is x then the duty cycle of the PWM signals will be x/255.

Bits 3-0 **Clock Divide Value.** The PWM clock is divided by 1 plus the values specified in bits 3 to 0. The divided clock is used as the base clock for the 8 bit PWM counter. The PWM signal remains high as long as the count is less than the value specified in bits 11 to 4. So to set the PWM signal to permanently high, the duty cycle should be 255. Similarly a duty cycle of zero would permanently disable the PWM signal.

Power On Default: 0000h

15.3.16. POWER CONTROL INDEX 00BH

Bits 15-3 *Reserved*.

Bit 2 **EVCC.** When set ’1’, this bit enables the VCC supply to the flat panel. The VCC supply is used for the digital part of the TFT.

Bit 1 **EVDD.** When set ’1’, this bit enables the VDD supply to the flat panel. The VDD supply is used for the analog part of the TFT.

Bit 0 **Reset.** Resets all the ouput signals going to flat panel including FPLINE, FPFRAME, DE, and RGB. It also forces all the internal logic to go into IDLE state. PWM signals are not affected by this signal

Power On Default: 0001h

To avoid damage to the flat panel be sure to follow the directions for switching on the TFT interface as per the specification of the TFT being used. Normally after powering up the system, all interface signals should be low, so bit 0 should be 1. First the digital part of the flat panel should be activated, so set bit 1 as 0. After about 50ms enable the digital interface signals by setting bit 0 to 0. After this wait for another 50ms and then enable the Vee or the analog power to the TFT. For switching down the power, follow the same sequence but in reverse order.

15.3.17. BLANK RED INDEX 00CH

Bits 15-8, *Reserved.*

Bits 7-0, This register represents the 8bit value of the red component of a blank pixel. This value is used to put blank pixels during image expansion in text mode. Note this value should correspond to the back ground color of the current text mode.

15.3.18. BLANK GREEN INDEX 00DH

Bits 15-8, *Reserved.*

Bit 7-0, This register represents the 8bit value of the green component of a blank pixel. This value is used to put blank pixels during image expansion in text mode. Note this value should correspond to the back ground color of the current text mode.

15.3.19. BLANK BLUE INDEX 00EH

Bits 15-8, *Reserved*

Bit 7-0, This register represents the 8bit value of the blue component of a blank pixel. This value is used to put blank pixels during image expansion in text mode. Note this value should correspond to the back ground color of the current text mode.

15.3.20. POLARITY CONTROL INDEX 00FH

This register is used to set the polarity of control signals from the CRTC and control signals to the flat panel.

Bits 15-5, *Reserved*

Bit 4, When 1 indicates TFT DE ploarity is active high. When 0 indicates active low.

Bit 3, When 1 indicates TFT line ploarity is active high. When 0 indicates active low.

Bit 2, When 1 indicates TFT frame ploarity is active high. When 0 indicates active low.

Bit 1, When 1 indicates CRTC vsync polarity is active high. When 0 indicates active low.

Bit 0, When 1 indicates CRTC hsync polarity is active high. When 0 indicates active low.

15.4 UPDATE HISTORY FOR TFT INTERFACE CHAPTER

The following changes have been made to the Interface Chapter.

Section	Change	Text
15.2	Removed	“~~mov bx, TFT Index base~~ ~~mov al, 0x1Dh~~ ~~out 22h, al~~ ~~mov al, bh~~ ~~out 23h, al~~ ~~mov al, bh~~ ~~out 23h, al~~”
15.2.4.	Added	“TFT BASE address index 0x1Dh”

Update History for TFT Interface chapter

16 PC CARD INTERFACE

The PCCard interface of the STPC Device combines a PCMCIA (16 bit) and a CardBus (32 bit) bridge. These two interfaces share a single socket.

16.1 PCI TO CARDBUS BRIDGE

The architecture of the bridge is mainly the architecture of a PCI to PCI bridge with the support of a different header type (Type 2 defined by Yenta group) and some registers linked to the socket status. In order to provide compatibility with PC Cards 16bit, a PCMCIA controller has been instatiated just behind the primary PCI interface through an ISA bridge.

Table 16-1. PCI to CardBus Register Layout

Device ID		Vendor ID		00
Status Register		Command Register		04h
Class Code Register			Revision ID	08h
BIST	Header Type	Latency Timer	Cache Line Size	0Ch
CardBus Socket Register / ExCA Base Address Register				01h
Secondary Status		*Reserved*		14h
CardBus Latency Timer	Subordinate Bus Number	CardBus Number	PCI Number	18h
Memory Base Register 0				1Ch
Memory Limit Register 0				20h
Memory Base Register 1				24h
Memory Limit Register 1				28h
▨		I/O Base Register 0		2Ch
▨		I/O Base Register 0		30h
▨		I/O Base Register 0		34h
▨		I/O Base Register 0		38h
Bridge Control		Interrupt Pin	Interrupt Line	3Ch
Sub System ID		Sub System Vendor ID		40h
16 bit PCCard Legacy Mode Base Address				44h
Reserved				48h-4Ch
▨	P2H_ENA	▨	ERR_Enable	50h
▨			ERR_Status	54h
▨			ST_RSVD	58h
Reserved				5Ch-7Ch
Vendor Specific				80h-FCh

16.2 PCCARD CONFIGURATION REGISTER DESCRIPTION

16.2.1 Vendor ID Index 0x00h

Bits 15-0 **Vendor ID.** Read only, they are hardwired to 0x104Ah. This is a 16-bit read-only register implemented at configuration space offset 0x0h and 0x1h and contains the Vendor Identifier. Writes to this register have no effect.

16.2.2 Device ID Index 0x02h

Bits 15-0 **Device ID.** Read only, they are hardwired to 0x0220h. This is a 16-bit read only register implemented at configuration space offset 0x2h and 0x3h. It contains the Device Identifier assigned to STPC Device .Writes to this register have no effect.

16.2.3 Command Reg Index 0x04h

This is the 16-bit PCI command register.

Bits 15-10 *Reserved*.

Bits 9 **Fast back to back enable.** This bit is hardwired to 0.

Bit 8 **SERR# enable**

Bit 7 **Address/Data stepping.** This bit is hardwired to 0.

Bit 6 **Parity error response**. This bit is hardwired to 0.

Bit 5 **VGA palette snoop.** This bit is hardwired to 0.

Bit 4 **Memory write/invalidate.** This bit is hardwired to 0.

Bit 3 **Special Cycles.** This bit is hardwired to 0.

Bit 2 **Bus master enable** on the primary side.

0 = Bridge ignores all mem and I/O transactions detected on the secondary side

1 = Bridge processes all transactions detected on the secondary side

Bit 1 **Memory space enable**

0 = Memory decode for this card disabled

1 = Memory decode for this card enabled

Bit 0 **I/O space enable**

0 = I/O decode for this card disabled

1 = I/O decode for this card enabled

This register defaults to 0x0000 after reset.

16.2.4 Status Reg Index 0x06h

This is the 16-bit PCI Status register.

Bit 15 **Detected parity error.** This bit is hardwired to 0.

Bit 14 **Signaled system error**

Bit 13 **Received master abort**

Bit 12 **Received target abort**

Bit 11 **Signaled target abort**

Bits 10-9 **Devsel timing.** These bits are hardwired to 2B01h.

Bit 8 **Data parity error reported.** This bit is, hardwired to 0.

Bit 7 **Fast back to back capable**. This bit is hardwired to 1.

Bit 6 **UDF supported.** This bit is hardwired to 0.

Bit 5 **66 Mhz capable.** This bit is hardwired to 0.

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

This register defaults to 0x0280h after reset.

16.2.5 Revision ID Index 0x08h

Bits 7-0 **Revision ID.** Hardwired to 0x00h.

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

16.2.6 Class Code Index 0x09h

This is a 24 bit read only register.

Bit 23-16 These bits are hardwired to 0x06h to indicate 'Bridge Device'.

Bit 15-8 These bits are hardwired to 0x07h to indicate 'PCI/CARDBUS bridge'.

Bit 7-0 these bits are hardwired to 0x00h.

16.2.7 Cache Line Size Index 0x0Ch

Bit 7-0 These bits are hardwired to 0x00h.

16.2.8 Latency Timer Index 0x0Dh

Bits 7-0 these bits are hardwired to 0x00h.

16.2.9 Header Type Index 0x0Eh

Bits 7-0 These bits are hardwired to 0x02h.

16.2.10 BIST Index 0x0Fh

Bits 7-0 These bits are hardwired to 0x00h.

16.2.11 CardBus Socket Registers Index 0x10h

This register points to the memory mapped I/O space that contains both the CardBus Socket register and ExCA Base Address register for 16-bit PC Card. CardBus Socket interface registers start at offset 000h and the 16-bit card socket interface registers begin at offset 800h. Bits 31-11 are R/W. Bits 11-0 are hardwired to zero. This indicates to configuring software that the bridge wants 4K bytes of non-prefetchable memory space, starting on a 4K boundary, that can be mapped anywhere in memory.

Bits 31-12 **Start address of socket** interface register block (aligned on 4KB boundaries). Mapped in PCI memory space.

Bits 11-0 The bits are hardwired to 0x0h.

16.2.12 Secondary Status Register Index 0x16h

The Secondary Status Register is similar in function to the Primary Status Register but contains information relating to the CardBus. Bit 14 of this register is defined differently to the Primary. When set it indicates that the bridge has detected SERR# asserted on the CardBus. This function is identical to that specified in the PCI to PCI bridge specification.

Bit 15 **Detected parity error,** on Secondary side.

Bit 14 **Signaled system error** on Secondary side. Writing a 1 clears the bit as well as reset.

Bit 13 **Received master abort .**

Bit 12 **Received target abort.**

Bit 11**Signaled target abort.** This bit is hardwired to 0.

Bit 10-9 **Devsel timing.** This bit is hardwired to 2B01h.

Bit 8 **Data parity error detected** on secondary side.

Bit 7 **Fast back to back capable.** This bit is hardwired to 1.

Bit 6 **UDF supported.** THis bit is hardwired to 0.

Bit 5 **66 Mhz capable.** NOT DEFINED for CardBus, this bit is hardwired to 0.

Bits 4-0 *Reserved.* This bit is hardwired to 0.

16.2.13 PCI bus Number Index 0x18h

Each PCI to CardBus bridge interface must implement three bus number registers. The Primary Bus number identifies the number of the PCI bus on the primary side of the bridge. This is set by the appropriate configuration software.

16.2.14 CardBus Bus number Index 0x19h

The CardBus Number identifies the number of the CardBus attached to the socket. This is set by PCI BIOS configuration software or Socket Services software. This register is called the ”Secondary Bus Number” on a PCI to PCI bridge.

Default: 00h

16.2.15 Subordinate Bus Number Index 0x1Ah

The Subordinate Bus Number is a register defined for PCI to PCI bridges. It holds the number of the bus at the lowest part of the hierarchy behind the bridge. Normally, a CardBus bridge will be at the bottom of the bus hierarchy and this register will hold the same value as the CardBus Bus Number register.

16.2.16 CardBus Latency Timer Index 0x1Bh

The CardBus Latency Timer has the same functionality of the primary PCI bus Latency Timer but applies to the CardBus attached to this specific socket. This is set by PCI BIOS configuration software or Socket Services software.

16.2.17 Memory Base Address 0 Index 0x1Ch

The Memory Base Register defines the bottom address of a memory mapped I/O window. The upper 20 bits correspond to address bits AD[31:12]. The bottom 12 bits of this register are read only and return zero when read. This window is enabled by bit 1 of the Command Register. Prefetching within this window is controlled by bit 8 of the Bridge Control Register. The default is enabled and should only be cleared if Memory Reads will cause side effects on the installed card.

Bit 31-12 **Programmable.** Controls where in CPU memory space the registers will be mapped.

Bits 11-0 These bits are hardwired to 0.

16.2.18 Memory Limit Address 0 Index 0x20h

The Memory Limit Register defines the top address of the memory mapped I/O space. The upper 20 bits of this register correspond to AD[31:12]. The bottom 12 bits of this register are read only and return zeros when read. The bridge assumes the bottom address bits 11-0 are ones to determine the range defined. So if the Memory Base and Limit registers are set to the same value a window of 4 KBytes is defined. Both Memory windows are both enabled by Command Register Bit 1. To disable either window individually, the Limit register of that range should be set below the Base. This will cause the bridge to never detect a hit on that window.

Bit 31-12 **Programmable,** specify memory end address

Bits 11-0 These bits are hardwired to 0.

16.2.19 Memory Base Address 1 Index 0x24h

Memory Base 1 has the same functionality as Memory Base 0. This window is enabled by bit 1 of the Command Register. Prefetching within this window is controlled by bit 9 of the Bridge Control Register. The default is enabled and should only be cleared if Memory Reads will cause side effects on the installed card.

Bits 31-12 **Programmable,** controls where in CPU memory space the registers will be mapped.

Bits 11-0 These bits are hardwired to 0.

16.2.20 Memory Limit Address 1 Index 0x28h

The Memory Limit 1 register has the same functionality as the Memory Limit 0 register.

Bits 31-12 **Programmable,** specify memory end address.

Bits 11-0 These bits are hardwired to 0.

16.2.21 I/O Base Register 0 Lower Half Index 0x2Ch

The IO Base Register defines the bottom address of an address range that is used by the bridge to determine when to forward an I/O transaction to the CardBus. The bits in this register correspond to AD[15:0]. Bits AD[1:0] are used to indicate whether the bridge implements 16 or 32 bit I/O addressing. For address decoding, if only 16 bit addressing is implemented the bridge must determine that the upper 16 bits of address are zeros before accepting an access. Address bits AD[15:2] provide the 4 byte granularity required by CardBus. This I/O mapping varies from a PCI to PCI bridge, in that it allows mapping the windows on a 4 byte boundary with a minimum size of 4 bytes. A PCI to PCI bridge maps I/O windows on 4K boundaries with a minimum 4 Kbyte size.

Bits 15-2 **Programmable**

Bits 2-0 These bits are hardwired to 00Bh (16 bit I/O support)

16.2.22 I/O Limit Register 0 Lower Half Index 0x30h

The I/O Limit Register defines the top address of the address range that is used by the bridge to determine when to forward an I/O access to the CardBus. The bits in this register correspond to AD[15:0].

Bits 15-2 **Programmable**

Bits 2-0 These bits are hardwired to 00Bh (16 bit I/O support)

16.2.23 I/O Base Register 1 Lower Half Index 0x34h

The IO Base Register defines the bottom address of an address range that is used by the bridge to determine when to forward an I/O transaction to the CardBus. The bits in this register correspond to AD[15:0]. Bits AD[1:0] are used to indicate whether the bridge implements 16 or 32 bit I/O addressing. For address decoding, if only 16 bit addressing is implemented the bridge must determine that the upper 16 bits of address are zeros before accepting an access. Address bits AD[15:2] provide the 4 byte granularity required by CardBus. This I/O mapping varies from a PCI to PCI bridge, in that it allows mapping the windows on a 4 byte boundary with a minimum size of 4 bytes. A PCI to PCI bridge maps I/O windows on 4K boundaries with a minimum 4 Kbyte size.

Bits 15-2 **Programmable**

Bits 2-0 These bits are hardwired to 00Bh (16 bit I/O support)

16.2.24 I/O Limit Register 1 Lower Half Index 0x38h

The I/O Limit Register defines the top address of the address range that is used by the bridge to determine when to forward an I/O access to the CardBus. The bits in this register correspond to AD[15:0].

Bits 15-2 **Programmable**

Bits 2-0 These bits are hardwired to 00Bh (16 bit I/O support)

16.2.25 Interrupt Line Index 0x3Ch

This register is used in the manner defined in the PCI Local Bus Specification and PCI to PCI Bridge Specification.

Bits 7-0 **Writable by POST.** If not implemented, hardwired to 0xFFh

16.2.26 Interrupt Pin Index 0x3Dh

Read only register.

Bits 7-0 These bits are hardwired to 0x01h (INTA# interrupt used)

16.2.27 Bridge Control Index 0x3Eh

The Bridge Control register provides extensions of the Command Register that are specific to PCI to PCI and PCI to CardBus bridges.

Bits 15-11 *Reserved*, These bits are hardwired to 0.

Bit 10 **Write posting Enable,** not used, reset to 0.

Bit 9 **Memory type for range 1 is prefetchable,** this bit is hardwired to 0.

Bit 8 **Memory type for range 0 is prefetchable,** this bit is hardwired to 0.

Bit 7 **IREQ/INT enable routing for 16 bit cards,** not used, reset to 0.

Bit 6 **CardBus reset assertion/deassertion** (Primary bus reset will override this bit) reset to 1.

Bit 5 **Master Abort Mode,** reset to 0.

Bit 4 *Reserved,* hardwired to 0.

Bit 3 **VGA enable,** hardwired to 0.

Bit 2 **ISA enable,** hardwired to 0.

Bit 1 **SERR# enable,** reset to 0.

Bit 0 **Parity error response enable.** If set to 1, SERR is asserted on the primary bus, when parity error is detected on CardBus.

Note: reserved location will be hardwired to 0. Reads to this location will respond with all 0's.

16.2.28 PC Card 16 Bit IF Legacy Mode Base Address (Optional) Index 0x44h

This optional register points to the index and data registers that resided at 3E1h and 3E0h in the PCMCIA. This register is intended only for legacy mode operation. It is not recommended that this mode be used by new software. New code should use the PC Card Socket Status and Control Registers Base Address space to address the registers directly. This register is cleared and disabled by PCIRST#. It must not respond to PCI cycles unless specifically loaded with a non-zero address after PCIRST# is deasserted. When this register is enabled memory mapped accesses to the Socket and Control Registers, via PC Card Socket Status and Control Registers Base Address Register, are disabled.
This makes the usage of this mode and the memory mapped mode mutually exclusive. This register's bits adhere to the definitions set out in the PCI Local Bus Specification.

Note: The implementation of this optional register does not remove the requirment to support direct memory mapped access to the 16 bit IF Control and Status registers via the PC Card Socket Status and Control Registers Base Address at 10h.

Default value; 03E1h.

16.2.29 ERR_ENABLE Index 0x50h

Bit 15-5 *Reserved***,** These bits are hardwired to 0.

Bit 4 **PERR_RDPE,** Cardbus PERR_ on read data parity error enable.

Bit 3 **PERR_WDPE,** Cardbus PERR_ on write data parity error enable.

Bit 2 **PERR-APE,** Cardbus PERR_ on address parity error enable.

Bit 1 **SERR_PE,** Cardbus SERR_ on PERR_ enable.

Bit 0 **SERR_RTA,** Cardbus SERR_ on received target abort enable.

Default value; 0000h.

16.2.30 P2H_ENABLE Index 0x52h

Bits 15-7 *Reserved***,** These bits are hardwired to 0.

Bit 6 **PCI_V2_1_DIS**, PCI rev.2.1 latency fix disable.

Bit 5 **P2H_PREFETCH_ENA,** PCI to host read prefetch enable.

Bit 4 **P2H_POST_ENA,** PCI to host write posting enable.

Bit 3-0 *Reserved***,** These bits are hardwired to 0.

Default value: 0000h

16.2.31 ERR_STATUS Index 0x54h

Bits 15-5 *Reserved***,** These bits are hardwired to 0.

Bit 4 **PERR_RDPE_STAT_R,** Cardbus PERR_ on read data parity error.

Bit 3 **PERR_WDPE_STAT_R,** Cardbus PERR_ on write data parity error.

Bit 2 **PERR-APE_STAT_R,** Cardbus PERR_ on address parity error.

Bit 1 **SERR_PE_STAT_R,** Cardbus SERR_ on PERR_.

Bot 0 **SERR_RTA_STAT_R,** Cardbus SERR_ on received target abort.

Default: 0000h

16.2.32 ST_RSVD Index 0x58h

This register supports some ST dedicated informations used by the SW.

Bits 31-7 *Reserved***,** These are hardwired to 0.

Bit 6 **DMARD,** The second miscellaneous features byte specifies the DMARD which signal is used to route TC (Terminal Count) signal on the correct socket pin..See Table 16-2.

Table 16-2. DMARD Specification

DMARD	TC Routing
0	DMA write. TC Routed on pin 9 (OE#)
1	DMA read. TC Routed on pin 15 (WE#)

Bit 5 **DMAWIDTH.** The second miscellaneous features byte specifies the width of the DMA transfer. The SW should select the corresponding DMA channel that support the data width indicated by the PC Card. See Figure 16-3.

Table 16-3. DMA Width Specification

DMA Width	Data tranfer
0	8-bit
1	16 bit

Bit 4 **DREQEN1** (also Bit 3 DREQEN0) are used to route DMA signals to the appropriate socket pin . DREQ# can be assigned to anyone of three memory or I/O interface pins, thereby replacing the standard I/O signals. The SW should interrogate the CIS (miscellaneous features field of the configuration table Entry Tuple 1Bh. Bit 7 of the first miscellaneous features extension byte is set to indicate that the 2nd byte is present. Bit 4 and 3 of the second byte specifies which interface pin the PC Card uses for DREQ#. The SW is in charge to copy this coding to bit 4 and 3 of PCI configuration register 22h. See Figure 16-4.

Table 16-4. Support for DREQ#

DMAREQ[1:0]	DREQ ROUTING
00	DMA Not supported
01	DREQ# USES SPKR#
10	DREQ# USES IOIS16#
11	DREQ# USES INPACK#

Bit 3 **DREQEN0** See bit 4.

Bit 2 *Reserved*

Bit 1 **CARD_32** (also Bit 0 CARD_16) are coding the result of the inquiry of the socket management unit on the type of PC Card inserted into the socket. See Figure 16-5. for the meaning of the different codes:

Table 16-5. PC Card type

ST_RSVD[1:0]	Card type
00	No card inserted
01	16 bits card (R2)
10	32 bits Cards
11	*Reserved*

Bit 0 **CARD_16** see Bit 1.

Default value: 0000h

16.2.33 All Remaining Configuration Registers Indeces 0x44h-0x48h and 0x5Ch-0xFCh

Bits 31-0 These bits are hardwired to 0.

16.3 PCMCIA FUNCTIONAL DESCRIPTION

16.3.1 Overview

The PCMCIA provides a single channel PC Card Interface Controller, (PCMCIA), with an EXCA interface ^{TM}. This PCMCIA design not only greatly extends the connectivity options which can be used by the PC User, but also provides a basis for access to an even broader range of future functions.

The PCMCIA design is based on the PCMCIA/JEIDA standard and provides an open standard system interface for PC Cards at the hardware and data interchange level. This ensures that PC Card compatibility can be realised across manufacturers of PC Cards as well as between systems provided by the notebook PC vendors.

16.3.2 Interface Registers

A set of interface registers is provided for PC Card Socket

.The interface decoding of these registers is described in the PC Card Interface. The detailed functions of the interface registers are described in the Register Description. The following sections outline the functionality of the interface registers.

16.3.2.1 General Setup Registers

These registers perform the following functions:

1) Identification and Revision information for the socket.

2) Status of the Interface

3) Power and RESETDRV Control

4) Card Status Change

5) Address Window Enable

6) Card Detect and General Control Register

7) Global Control Register

16.3.2.2 Interrupt Registers

These registers perform the following functions:

1) Interrupt and General Control

2) Card Status Change Interrupt Configuration.

16.3.2.3 I/O Registers

These registers perform the following functions:

1) I/O Control

2) I/O Address window 0-1 Start/Stop Low Byte

3) I/O Address window 0-1 Start/Stop High Byte

16.3.2.4 Memory Registers

These registers perform the following functions:

1) System Memory Address Window 0-4 Start/ Stop Low Byte

2) System Memory Address Window 0-4 Start/ Stop High Byte

3) Card Memory Index Address Window 0-4 Low Byte

4) Card Memory Index Address Window 0-4 High Byte

16.3.3 Interrupt Steering

The PCCard Interface has 1 IRQ line for both the status and PCCard Interrupt.

Due to interrupt routing limitations for the PCMCIA.

16.3.4 Memory Control

16.3.4.1 PC Card Memory Addressing

The PCMCIA enables the System User to map portions of the 64Mbyte (26 bit address), (common and/or attribute), memory spaces on the PC Card into the smaller 16Mbyte (24 bit address), system (ISA) address space. The mapping functions provided will allow for the system side address space to be extended to the full 64Mbyte PC Card capability.

The PCMCIA has 5 (five) independently enabled and controlled system memory address mapping windows. Each of these windows can be set to map either into the common or attribute memory space of the PC Card, (Common/Attribute Memory Address Mapping). Every system memory window has fully independent control of the memory data bus width, system bus wait states, software write protect, and enable.

System Memory windows start and stop on any 4Kbyte boundary of the ISA system memory above 64Kbytes. The PCMCIA will automatically inhibit any mapping of a system memory window below 64Kbyte in the ISA system address. This is due to the fact that the first 64K addresses are reserved for ISA I/O, and also in order to resolve conflicts in accesses to I/O PC Cards that contain memory.

In order to open any of the 5 windows, system memory start address, system memory stop address, and the PC Card memory offset are intialised. Accesses to the PC Card memory will occur when the following conditions have been satisfied:

1) The system memory address mapping window is enabled.

2) The ISA system memory address is greater than or equal to the mapping start register

A23:12 (high and low byte).

3) The ISA system memory address is less than or equal to the mapping stop register

A23:12 (high and low byte).

When the above is satified PCMCIA will add the PC Card memory offset address to the ISA system address in order to generate the PC Card address.

All windows can be configured independently, or alternatively grouped together in order to perform special memory mapping requirements such as for example, LIM/EMS (Lotus-Intel-Microsoft/Extended Memory Specification), or XIP (Execute in Place).

The PC Card memory offset allows for either positive or negative (2's complement) offset from the ISA system address. The PCMCIA does not perform any checks for any window whose size and offset cause it to wrap round from the last PC Card address to the first. The system software has total responsibility for preventing address wrap round effects.

16.3.5 Common/Attribute Memory Address Mapping

As stated in PC Card Memory Addressing Common/Attribute Memory on the PC Card can be accessed through any of the five system memory address mapping windows. This is achieved by setting the ”REG active bit” in the card memory offset address register (high byte), (0 = common memory, 1 = attribute memory). There are no restrictions for the mapping of the system memory window to common/attribute memory from any ISA address to any PC Card address.

Multiple system memory address mapping windows to different common/attribute memory address spaces can be opened simultaneously. All windows can be independently configured for wait states, software write protect and data width.

16.3.6 Memory Paging

The PCMCIA supports the paging of system memory by the use of multiple system memory address mapping windows. When using LIM or XIP, software must assign a window to each page required to support the LIM/XIP function. It is the responsibility of the software to set up the system memory address mapping windows into one contiguous system address space, and to ensure that each window controls a single page in the PC Card memory. In these conditions only the PC Card memory offset address value needs to be altered to change the page pointer and mapping.

16.3.7 I/O Control

The PCMCIA method of I/O addressing is similar to that described in Memory Control. In this case the System User can map portions of the 0 to 64Kbyte I/O address range of a PC Card into the smaller 768 byte I/O system (ISA) address space. The PC Card can request certain common I/O address locations or alternatively only the size of the space required. In the case when only the size of I/O space is requested, the system is able to locate the PC Card anywhere in the 64Kbyte system I/O address space.

The PC Card decodes the CE#2, CE#1, IORD#, and IOWR# signals in order to respond to an I/O access

The PCMCIA has 2 (two) independently enabled and controlled I/O address windows each of which define a 16 Bit address (1 byte resolution). This feature allows for two non-contiguous I/O address windows to be available for each PCMCIA socket. Each of these windows has independent control of I/O data bus width, zero wait state system access and generation of IOCS16#.

An I/O PC Card can be accessed providing that the following conditions are satisfied:

1) The I/O address window is enabled.

2) The ISA system address SA[15:0] is greater than or equal to the I/O address start register (high and low byte).

3) The ISA system address SA[15:0] is less than or equal to the I/O address stop register (high and low byte).

4) The access is not a DMA transfer (AEN = ‘0’ to access the I/O PC Card)

System software is required to be responsible for accounting for each I/O address range assigned to a particular PC Card. Card power consumption can be reduced by the reservation of a particular I/O address range for each PC Card, this is due to the fact that only one PC Card is enabled during each I/O access.

As there is no indirect offset addressing within the PCMCIA, the system's I/O address must be within the PCMCIA's I/O window in order to generate the correct I/O cycle to the PC Card. The I/O addressing bits [15:0] of the PC Cards are directly derived from SA[15:0] with bits [25:16] driven “0”.

The PCMCIA directly maps the system I/O address space to the PC Card I/O ports to a single byte resolution. Each PC Card is guaranteed a reserved system I/O space, and an I/O cycle will be generated to the PC Card within the assigned space. This means that the PCMCIA does not rely on the PC Card to decode the I/O address space and respond with the acknowledge signal. However for systems that generate card enables to the PC Card over a wide range of I/O address space, and rely on the PC Card acknowledge signals to enable any data transreceivers between the PC Slot and the system data bus, the PCMCIA Input Acknowledge (PACK#) will be required. On a read from the PC Card with an address window enabled, the PCMCIA will qualify the data transreceiver direction line and the card enables with a valid access to the PC Card I/O address space. The PCMCIA will not allow overlapping I/O address windows to be enabled concurrently.

16.4 PCMCIA CONTROL

16.4.1 PCMCIA Card Status

The status of the PC Card is accessible through the interface status register. The status information in this register includes PC Card detection, memory write protect status, battery voltage detect, PC Card power, and ready/busy. Changes in PC Card status can cause a card status change interrupt, (for example when a PC Card is inserted or removed). The source of the interrupt is configurable.

16.4.2 Control/Status Signal Multiplexers

PC Card signals have differing designations depending on whether the PC Card is memory or I/O. The PCMCIA contains multiplexers to redirect the appropriate signals based upon the condition of the PC Card type bit in the interrupt and general control register.

16.4.3 Configuration Registers

The PCMCIA provides a register containing interface identity and revision information for the socket.

16.4.4 Power Management

The PCMCIA provides independent power management control signals for each PC Card socket. Socket power management is controlled by programming the POWER and RESETDRV control register.

The actual power buffers are required to be added externally by the System Designer, refer to (2.9).

The PCMCIA automatically enters into a lower power consumption state when the memory and I/O windows are disabled and the sockets are empty. The lowest power consumption level can be achieved by setting the PCMCIA into CS# controlled power down mode. This mode is entered by disabling all the I/O/Memory windows, output buffers, enabling the power down mode, and driving CS# high.

During the CS# power controlled down mode the PCMCIA will still be powered up and the contents of the internal registers will be maintained. Interrupt requests (IRQs) can still be generated to the Host System by the PCMCIA from either card status change, or PC Card interrupt requests. Additionally RI_OUT# can still be set to route either RI_IN#, or Card detect change. Furthermore INTR# can also be set to route either card status change interrupts or PC Card interrupts.

The PCMCIA provides a unique feature to support host system suspend/resume operations. When the host system enters suspend mode the PCMCIA still maintains the ability to route the RI# signals from both sockets to the RI_OUT# signal as the resume indication to the host system.

16.4.5 PC Card Interface

The PCMCIA directly supports one PC Card Sockets (A). The PC Card socket interface has its own complement of memory address mapping, I/O address mapping, configuration and status registers. Additionally a set of ID and revision registers is provided.

All PCMCIA control registers are byte wide and are accessed using the following indirect indexing scheme. Two ISA I/O addresses are used to access the PCMCIA's control registers. The first ISA I/O address is the index register and the second ISA I/O address the data register. The PC Card socket can have up to 64 indirectly addressed registers, thus allowing support of the two separate PC Card Sockets using only 2 ISA I/O addresses.

16.4.6 Interface Decode Logic

The interface decoding logic decodes the ISA I/O addresses 3E0h and 3E1h as the addresses of the PCMCIA index and data registers respectively. To access one of the PCMCIA registers the system must first write the index value to the index register, and then either read or write to the data register. Both the index and data registers are read/write. Providing the index register contains a valid index, the PCMCIA will respond to a data register read/write operation, or to an index register read operation.

A mismatch in the address decode will cause a clearing of all 4 of the index registers in the PCMCIA.

Table 16-6. Index Register Mapping

Socket A	Base Address	Index Range
Slot 0	3E0h	00h to 3Fh

16.5 EXTERNAL CONNECTIONS

The external connections required by the System Designer are illustrated in Fig. 2.9. The PC Card data (SD[15:0]) and address (SA[11:0]) bus is electrically isolated from the internal ISA bus by two sets of transceivers and buffers. The PCMCIA provides control signals to enable the transfer of odd or even data bytes, (CE#1 and CE#2 respectively) between the PC Card and the ISA bus via transceivers. The system address is electrically isolated by a buffer. Power Control signals are provided from the PCMCIA and these control via isolating circuitry VCC/VPP to each card socket.

Note: I/O PC Card Functions in the following table are listed directly below the Memory PC Card functions and are designated in ( ).

16.6 REGISTER DESCRIPTION

16.6.1 PCMCIA On-Chip Register Set

Table 4.1. is a list of all the interface registers contained in the PCMCIA, together with their offset values..

Table 16-7. PCMCIA Interface Register Set

Slot	Description
GENERAL SET UP REGISTERS	
00h	Identification and Revision
01h	Interface Statu
02h	Power and RESET DRV Control
04h	Card Status Change
06h	Address Window Enable
16h	Card Detect and General Control
1Eh	Global Control Register
INTERRUPT REGISTERS	
03h	Interrupt and General Control
05h	Card Status Change Interrupt Configuration
I/O REGISTERS	
07h	I/O Control
08h	I/O Address 0 Start Low Byte
09h	I/O Address 0 Start High Byte
0Ah	I/0 Address 1 Stop Low Byte
0Bh	I/O Address 1 Stop High Byte
0Ch	I/O Address 2 Start Low Byte
0Dh	I/O Address 2 Start High Byte
0Eh	I/O Address 2 Stop Low Byte
0Fh	I/O Address 2 Stop High Byte
MEMORY REGISTERS	
10h	System Memory Address 0 Mapping Start Low Byte
11h	System Memory Address 0 Mapping Start High Byte
12h	System Memory Address 0 Mapping Stop Low Byte
13h	System Memory Address 0 Mapping Stop High Byte
14h	Card Memory Index Address 0 Low Byte
15h	Card Memory Index Address 0 High Byte
17h	*Reserved*
18h	System Memory Address 1 Mapping Start Low Byte
19h	System Memory Address 1 Mapping Start High Byte
1Ah	System Memory Address 1 Mapping Stop Low Byte
1Bh	System Memory Address 1 Mapping Stop High Byte
1Ch	Card Memory Index Address 1 Low Byte
1Dh	Card Memory Index Address 1 High Byte

Table 16-7. PCMCIA Interface Register Set

1Fh	*Reserved*
20h	System Memory Address 2 Mapping Start Low Byte
21h	System Memory Address 2 Mapping Start High Byte
22h	System Memory Address 2 Mapping Stop Low Byte
23h	System Memory Address 2 Mapping Stop High Byte
24h	Card Memory Index Address 2 Low Byte
25h	Card Memory Index Address 2 High Byte
26h	*Reserved*
27h	*Reserved*
28h	System Memory Address 3 Mapping Start Low Byte
29h	System Memory Address 3 Mapping Start High Byte
2Ah	System Memory Address 3 Mapping Stop Low Byte
2Bh	System Memory Address 3 Mapping Stop High Byte
2Ch	Card Memory Index Address 3 Low Byte
2Dh	Card Memory Index Address 3 High Byte
2Eh	*Reserved*
2Fh	*Reserved*
30h	System Memory Address 4 Mapping Start Low Byte
31h	System Memory Address 4 Mapping Start High Byte
32h	System Memory Address 4 Mapping Stop Low Byte
33h	System Memory Address 4 Mapping Stop High Byte
34h	Card Memory Index Address 4 Low Byte
35h	Card Memory Index Address 4 High Byte
36h-3Fh	*Reserved*

16.6.2 General Set Up Registers

16.6.2.1 Identification and Revision Register (R) Index 0x00h

This register is used by the system software to determine the type of PC Cards supported, and also to identify which PCMCIA version is present. The system software reads this register and then compares the result value against existing revision numbers.

Bits 7-6 **PCMCIA Interface Type.** These bits indicate the type of PC Cards supported by the PCMCIA (Table 4.2.1.1.) at the particular socket. These bits do not identify the type of card that is present at the socket.

ID1	ID0	Interface
0	0	I/O Only
0	1	Memory Only
1	0	Memory & I/O
	1	*Reserved*

Note: These bits will be read back as 10.

Bits 5-4 *Reserved.* Bits 5 and 4 will be read back as zero.

Bits 3-0 **PCMCIA Revision.** Bits 3 to 0 indicate the current revision level of the PCMCIA.
Bits 3 to 0 will be read back as 0011.

16.6.2.2 Interface Status Register (R) Index 0x01h

This register provides the current status of the PC Card socket interface signals.

Active BSY# SPKR# STSCHG#

Bit 7 **GPI#.** This bit is hardwired to 1.

Bit 6 **PC Card Power Active.** This bit indicates the current power status of the socket.
If bit 6 = ’0’, power to the socket is turned off, (VCC ,VPP1 and VPP2 are no actively driven).
When bit 6 = ’1’, power is provided to the socket (VCC = 5V, and VPP1 and VPP2 are set according to bits 3-0 in the power control register).

Bit 5 **Ready/BSY#.** This bit indicates the ready condition of the PC Memory Card.
If this bit = 1, the PC Card is ready to accept a new data transfer. When bit 5 = 0, the PC Card is busy processing a previous command, or alternatively performing initialization for an I/O Card. The default state of bit 5 = 1.

Bit 4 **Memory Write Protect.** This bit 4 gives the logic level of the WP pin.

If WP pin = ’0’, bit 4 = ’0’.
If WP pin = ’1’, bit 4 = ’1’.

Memory write access to the slot will not be blocked unless the write protect bit in the associated Card Memory Index Address Register High byte is set to a one.

Bits 3-2 **Card Detect 2 and 1.** These bits indicate whether a card is present at the socket and fully seated.

If CD#1 = 0, bit 2 = ’1’. If CD#1 = 1, bit 2 = ’0’.
If CD#2 = 0, bit 3 = ’1’. If CD#2 = 1, bit 3 = ’0’.

They are set to zero if the PC Card interface are inactive.

Bits 1-0 **Battery Voltage Detect 2 and 1.** The following table shows the possible battery states as indicated by bits 2 and 1 of this register.

BVD1	BVD2	Status
0	0 or 1	Battery Dead
1	0	Warning
1	1	Battery Good

For I/O PC Card, bit 0 indicates the current status of the STSCHG#/RI# signal from the PC Card, while bit 1 indicates the current state of SPKR# from the PC Card.

16.6.2.3 Power and RESETDRV Control Register Index 0x02h

This register provides both the control of PC Card Power, and the resetting of the PCMCIA’s registers.

A RESETDRV clears all bits in this register, unless the RESETDRV is a result of a Host System resume (PWRGOOD=’1’) and the disable resume RESETDRV bit is set to a one. (Output Enable (D6) should not be set until the register has been previously written setting PC Card Power Enable (D4)).

Bit 7 **Output Enable.** If Bit 7 is set to zero, the following PC Card Outputs are tri-stated, and the ENABLE# pin to the corresponding socket is inactive:
CA[25:12], CE#1, CE#2, IORD#, IOWR#, OE#, REG#, RESET, WE#.

The functionality of the Output Enable bit is illustrated in Figure 16-10.. This bit should always be set to zero when a slot is not powered.

Bit 6 **RESETDRV Disable Resume.** This bit specifies the action of RESETDRV signal under resume (PWRGOOD=’1’).r.

Disable RESETDRV	Signal PWRGOOD	Signal RESETDRV Action
0 or 1	0	System reset. Resetable registers **are** reset.
0	1	System resume. Resetable registers **are** reset.
1	1	System resume. Resetable registers **are not** reset.

Bit 5 **Auto Power Switch Enable.** Automatic socket power switching based on card detects is disabled when this bit is set to zero.
Conversely if the bit is set to a one then automatic power switching is enabled. The automatic socket power switching function controls the VCC_EN# and VPP_ENX output pins.

The PCMCIA provides each socket with 5 power control pins. These are physically used to control the power supplies (VCC, VPP2 and VPP1) to the PC Cards. .

Control Bit	Name	Description
0	VPP1_EN0	VPP 1 control, bit 0
1	VPP1_EN1	VPP 1 control, bit 1
2	VPP2_EN0	VPP 2 control, bit 0
3	VPP2_EN1	VPP 2 control, bit 1
4	VCC_EN#	Master Enable

Bit 4 **PC Card Power Enable.** When this bit is set to zero, all power to the PC Card is disabled. However when this bit is set to a one VCC_EN# = 0, VCC= 5V, and VPP1/VPP2 are set in accordace with the states of bits [3:0] in this register.

Bits 3-2 **PC Card VPP 2 Power Control.** Bits 3 and 2 control the VPP2_EN1, VPP2_EN0 output pins of PCMCIA as shown in Table 16-8. .

Table 16-8. VPP2 Outputs

Bit 4	Bit 3	Bit 2	VPP2_EN1	VPP2_EN0	VCC_EN#
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
0	X	X	0	0	1

Note: The 11 combination of Bits 2 and 3 is reserved and should not be used.

Bits 1-0 **PC Card VPP1 Power Control.** Bits 1 and 0 control the VPP1_EN1, VPP1_EN0 output pins of PCMCIA as shown in Table 16-9. .

Table 16-9. VPP1 Outputs

Bit 4	Bit 1	Bit 0	VPP1_EN1	VPP1_EN0	VCC_EN#
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
0	X	X	0	0	1

Note: The 11 combination of Bits 1 and 0 is reserved and should not be used.

Slot Power Control

A PC Card is considered to be present, (detected), in the socket when both CD#2 and CD#1 = '0'. The PC Card Power Active bit of the Interface Register indicates the current power status of the socket. When this bit is zero, power to the socket is turned off (VCC_EN#, VPP1_EN0, VPP1_EN1, VPP2_EN0, and VPP2_EN1 are all inactive). However when the bit is set to a one, power is provided to the socket

(VCC_EN# = 0V and VPP1 and VPP2 are set according to bits 3-0 in the power control register).

Table 16-10. summarises the slot power control function

Note: PC Card Power Active = 0 means VPPX_EN#, VCC_EN#, and NVCC_EN pins are active to provide power supply to VCC, VPP1 and VPP2.

The power control circuitry only switches the applicable voltages and does not provide voltage generation.
For compliance with IEC 950 and UL 1950 It is recommended that a 1A, 125V fuse be installed on each PCMCIA socket voltage line.

.

Table 16-10. Slot Power Control

Power Control Register			PC Card Pins		Logic Outputs	Interface Status Register
Output Enable (D7)	PC Card Power Enable (D4)	Auto Power Switch Enable (D5)	CD#1	0CD#2		PC Card Power Active
X	0	X	X	X	OFF	0
0	1	0	0	0	OFF	1
1	1	0	0	0	ON	1
X	1	0	X	1	OFF	1
X	1	0	1	X	OFF	1
0	1	1	0	0	OFF	1
1	1	1	0	0	ON	1
X	1	1	X	1	OFF	0
X	1	1	1	X	OFF	0

16.6.2.4 Card Status Change Register Index Value (Base + 04h)

This register contains the status of the sources for the card status change interrupts. These sources can be enabled to generate a card status change interrupt by setting the appropriate bit in the Card Status Change Interrupt Configuration Register. The bits in this register, (Card Status Change), will be read back as ’0’, when the Card Status Enable bits are set to zero in the Card Status Change Interrupt Configuration Register for the various sources of the card status change interrupts.

In the condition where the Explicit Write Back Card Status Change Acknowledge bit is set in the Global Control Register, the acknowledgement of the card status change interrupt sources is achieved by writing a one back to the appropriate bit in the Card Status Change Register that was read as a one. Once acknowledged, that particular bit in the Card Status Change Register will be read back as ’0’. The interrupt caused by card status change, (if enabled on a system IRQ line), will be active until all of the bits in this register are zero.

Should the Explicit Write Back Card Status Acknowledge bit not be set, the card status change interrupt, when enabled on a system IRQ line, will remain active until the Card Status Change Register is read.
This read operation will reset all the bits in the Card Status Change Register.

In the case where two or more card status change interrupts are pending and a card status change interrupt occurs while serving one source of card status change, the PCMCIA will not generate a second interrupt pulse.

Therefore, in explicit write back acknowledge mode, the Host System interrupt service routine must acknowledge each Card Status Change Interrupt source by writing 1’s to the respective bits in the Card Status Change Register. In the standard acknowledge mode, the software interrupt service routine must first read the Card Status Register to store all the card status change sources and then service them in turn.

In both of the above modes the Interrupt Service Routine needs the Card Status Change Register in order to be sure that all interrupt requests are serviced before issuing any service routines.

A RESTDRV clears all the bits in this register, unless RESETDRV is the result of a Host System resume (PWRGOOD = ’1’), and Disable RESETDRV = ’1’ (Power and RESETDRV control register).

Note: In the following sections bit descriptions in ( )’s indicate valid signals after the interface is configured for I/O Cards.

Bits 7-5 *Reserved.* These bits always read zero.

Bit 4 *Reserved***.** This bit is hardwired to 1.

Bit 3 **Card Detect Change.** Bit 3 is set to a one when a change has been detected on either CD#1 or CD#2.

Bit 2 **Ready Change.** Bit 2 is set to a one when a low to high transition has been detected on Ready/NBUSY, this indicates that the memory PC Card is ready to accept a new data transfer. For I/O PC Cards this bit reads zero.

Bit 1 **Battery Warning.** When a battery warning condition has been detected this bit is set to a one. For I/O PC Cards this bit reads zero.

Bit 0 **Battery Dead (STSCHG#).** Bit 0 is set to a one when a battery dead condition has been detected on Memory PC Cards.

For I/O PC Cards this bit is set to a one if the ring indicate enable bit is set to zero (Interrupt and General Control Register) and the (STSCHG#/RI#) signal from the I/O PC Card has been pulled low. The system software then has to read the status change register in the PC Card to determine the cause of the status change signal (STSSCHG#). This bit also reads zero for I/O PC Cards if the ring enable bit is set to a one.

16.6.2.5 Address Window Enable Register Index 0x06h

This register controls the enabling of the memory and I/O mapping windows to the PC Card memory or I/O space. RESETDRV clears all bits in this register, unless the RESETDRV is a result of a Host System resume (PWRGOOD=’1’) and disable resume RESETDRV is set to a one (Power and RESETDRV control register).

In order for the CS# controlled power-down mode to function properly, all the memory and I/O window enable bits in this register need to be set to zero before the PCMCIA enters the power-down mode.

Bit 7 **I/O Window 1 Enable.** In the condition where bit 7 = ’0’, an I/O access within the I/O address 1 window will inhibit card enable signals to the PC Card. Conversely if bit 7 = ’1’, an I/O access within the I/O address 1 window will generate card enables to the PC Card. (The start and stop register pairs must all be set to the desired window values before setting this bit to one).

Bit 6 **I/O Window 0 Enable.** In the condition where bit 6 = ’0’, an I/O access within the I/O address 0 window will inhibit card enable signals to the PC Card. Conversely if bit 6 = ’1’, an I/O access within the I/O address 1 window will generate card enables to the PC Card. (The start and stop register pairs must all be set to the desired window values before setting this bit to one).

Bit 5 **MRMCS16# Decode.** When bit 5 is set to zero, MEMCS16# is generated from a decode of the system address lines SA[23:17] only. Therefore at a minimum a 128K block of system (ISA) memory address space is set aside as 16 bit memory only. If bit 5 = ’1’ MEMCS16# is generated from a decode of the system address lines SA[23:12] (4K block).
Full line address decode should be used when decoding in the first 128K block of address space.

Bit 4 **Memory Window 4 Enable.** When bit 4 = ’0’, a memory access within the system memory address 4 window will inhibit the card enable signals to the PC Card. If however bit 4 = ’1’, a memory access within the system memory address 4 window will generate the card enables to the PC Card. The start, stop and offset register pairs must all be initialised before setting bit 4 to a one. When bit 4 = ’1’ the PC Card address is generated for valid window addresses.

Bit 3 **Memory Window 3 Enable.** When bit 3 = ’0’, a memory access within the system memory address 3 window will inhibit the card enable signals to the PC Card. If however bit 3 = ’1’, a memory access within the system memory address 3 window will generate the card enables to the PC Card. The start, stop and offset register pairs must all be initialised before setting bit 3 to a one. When bit 3 = ’1’ the PC Card address is generated for valid window addresses.

Bit 2 **Memory Window 2 Enable.** When bit 2 = ’0’, a memory access within the system memory address 2 window will inhibit the card enable signals to the PC Card. If however bit 2 = ’1’, a memory access within the system memory address 2 window will generate the card enables to the PC Card. The start, stop and offset register pairs must all be initialised before setting bit 2 to a one. When bit 2 = ’1’ the PC Card address is generated for valid window addresses.

Bit 1 **Memory Window 1 Enable.** When bit 1 = ’0’, a memory access within the system memory address 1 window will inhibit the card enable signals to the PC Card. If however bit 1 = ’1’, a memory access within the system memory address 1 window will generate the card enables to the PC Card. The start, stop and offset register pairs must all be initialised before setting bit 1 to a one. When bit 1 = ’1’ the PC Card address is generated for valid window addresses.

Bit 0 **Memory Window 0 Enable.** When bit 0 = ’0’, a memory access within the system memory address 0 window will inhibit the card enable signals to the PC Card. If however bit 0 = ’1’, a memory access within the system memory address 0 window will generate the card enables to the PC Card. The start, stop and offset register pairs must all be initialised before setting bit 0 to a one. When bit 0 = ’1’ the PC Card address is generated for valid window addresses.

16.6.2.6 Card Detect/ General Control Register Index 0x16h

A RESETDRV clears all bits in this register, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

Bits 7-6 *Reserved.* These bits should not be used.

Bit 5 **Software Card Detect Interrupt.** In the condition where the Card Detect Enable Bit = ’1’ (Card Status Change Interrupt Configuration Register), then writing a ’1’ to bit 5 will cause a card detect card status change interrupt for the associated slot. The functionality and acknowledgement of this software interrupt will work in the same way as the hardware generated interrupt.

Note 1: The functionality of the hardware card detect card status change interrupt will not be affected. The previous state of the CD#1 and CD#2 inputs will be latched such that while a S/W card detect card status change interrupt occurs and is being serviced, any new changes of state of CD#1 and CD#2 will cause an H/W card detect card status change interrupt to be generated.

If the Card Detect Enable Bit = 0 (Card Status Change Interrupt Configuration Register), then writing a 1 to the S/W Card Detect Interrupt bit has no effect.

Note 2: The S/W Card Detect Interrupt bit will always read back as a ’0’.

Bit 4 **Card Detect Resume Enable.** The default state of this bit = ’0’. In the condition where bit 4 = ’1’, then oCE# a card detect change has been detected on the CD#1 and CD#2 inputs, the RI_OUT# output will change state from high to low and the Card Detect Change bit (Card Status

Change Register) will be set to a ’1’. The RI_OUT# output will remain low until either a read or a write of 1 to the Card Detect Change bit (Card Status Change Register), (acknowledge cycle), causes the Card Detect Change bit to be cleared and RI_OUT# output to change state from low to high. The Card Detect Enable bit must be set in the Card Status Change Interrupt Configuration Register in order to generate RI_OUT#.

In the condition where the card status change is routed to either INTR# or any IRQ, the setting of Card Detect Resume Enable bit to a one will prevent INTR# and IRQ from going active as a result of card status change. OCE# the resume software has detected a card detect change interrupt from RI_OUT# (Card Status Change Register read), the software should initiate a software card detect change so that the card detect change condition will generate an active interrupt on IRQ or INTR# (which one being dependent on the active configuration).
When bit 4 is set to ’0’, then the card detect resume functionality is disabled. Therefore RI_OUT will not go low due to a card detect change.
The RI_OUT# output will be the logical AND of all the active low sources for ring indicate output including the RI# inputs from slot A and slot B, together with the card detect changes on CD#1, CD#2 from both slots.

Bit 3 **GPI Transition Control.** The default state of bit 3 = ’0’. The setting of the General Purpose Input (GPI) Enable bit to a ’1’ will enable a card status change interrupt when the GPI# input changes state from a high to a low, providing bit 3 (GPI Transition Control) = ’0’. If the GPI Enable bit is set to a one and bit 3 is set to a one, then oCE# the GPI# input changes state from low to high a card status change interrupt will be generated. When GPI# is used for card ejection/insertion pending events, transition control allows for either of these events, (not both), to cause an interrupt.

Bit 2 *Reserved.* This bit is hardwired to 1.

Bit 1 **Configuration Reset Enable.** The default state of bit 1 = ’0’. There is one bit for each slot. If bit 1 is set to a ’0’, the configuration register reset function based on card detects is disabled. When bit 1 is set to a ’1’, in the condition where both the CD#1 and CD#2 inputs for a particular slot go high, a reset pulse will be generated. This pulse will reset the configuration registers for that particular slot to their default states (’0’s).

NOTE: Bit 1 is required to be set by the card detect change interrupt service routine only when a PC Card is inserted, and reset when the card is removed. It is also a requirement to enable the card detect card status change interrupt if the card detect configuration reset function is to be used in the CS# power down mode.

Table 16-11. shows a list of registers that will be reset to zero when the Configuration Reset Enable is set to a 1 and both the CD#1 and CD2 inputs go high.

Table 16-11. Slot Registers set to Zero

Slot	Description
03h	Interrupt and General Control **(Except ’INTR# Enable’ bit)**
06h	Address Window Enable **(Except ’MEMCS#16 Decode A23:12 bit)**
07h	I/O Control
08h	I/O Address 0 Start Low Byte
09h	I/O Address 0 Start High Byte
0Ah	I/0 Address 0 Stop Low Byte
0Bh	I/O Address 0 Stop High Byte
0Ch	I/O Address 1 Start Low Byte
0Dh	I/O Address 1Start High Byte
0Eh	I/O Address 1 Stop Low Byte
0Fh	I/O Address 1 Stop High Byte
10h	System Memory Address 0 Mapping Start Low Byte
11h	System Memory Address 0 Mapping Start High Byte
12h	System Memory Address 0 Mapping Stop Low Byte
13h	System Memory Address 0 Mapping Stop High Byte
14h	Card Memory Index Address 0 Low Byte
15h	Card Memory Index Address 0 High Byte
18h	System Memory Address 1 Mapping Start Low Byte
19h	System Memory Address 1 Mapping Start High Byte
1Ah	System Memory Address 1 Mapping Stop Low Byte
1Bh	System Memory Address 1 Mapping Stop High Byte
1Ch	Card Memory Offset Address 1 Low Byte
1Dh	Card Memory Offset Address 1 High Byte
20h	System Memory Address 2 Mapping Start Low Byte
21h	System Memory Address 2 Mapping Start High Byte
22h	System Memory Address 2 Mapping Stop Low Byte

Table 16-11. Slot Registers set to Zero

23h	System Memory Address 2 Mapping Stop High Byte
24h	Card Memory Offset Address 2 Low Byte
25h	Card Memory Offset Address 2 High Byte
28h	System Memory Address 3 Mapping Start Low Byte
29h	System Memory Address 3 Mapping Start High Byte
2Ah	System Memory Address 3 Mapping Stop Low Byte
2Bh	System Memory Address 3 Mapping Stop High Byte
2Ch	Card Memory Offset Address 3 Low Byte
2Dh	Card Memory Offset Address 3 High Byte
30h	System Memory Address 4 Mapping Start Low Byte
31h	System Memory Address 4 Mapping Start High Byte
32h	System Memory Address 4 Mapping Stop Low Byte
33h	System Memory Address 4 Mapping Stop High Byte
34h	Card Memory Offset Address 4 Low Byte
35h	Card Memory Offset Address 4 High Byte

Bit 0 **16 Bit Memory Delay Inhibit.** The default state of bit 0 = ’0’. This is not programmable on a per window basis.
When bit 0 is set to ’0’ and a system memory window is set up to be 16 bit, (Data Size Bit = ’1’ in the System Memory Address Mapping Start High Byte Register), the falling edges of the control strobes EW# and OE# for the corresponding slot will be delayed synchronously by SYSCLK. These falling edges will be generated from the first falling edge of SYSCLK occurring after the falling edge of MEMW# or MEMR#, and will be also be gated by a valid system memory window decode. The rising edge of the control strobes will be generated from the rising edge of MEMW# or MEMR#.

In the condition where bit 0 is set to a ’1’ and the system memory window is set up to be 16 bit, the control strobes WE# and OE# for the corresponding slot will not be synchronously delayed by SYSCLK.

16.6.2.7 Global Control Register Index 0x1Eh

A RESETDRV clears all bits in this register, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = 1’ in the slot Power and RESETDRV Control Register.

Bit 7-4 *Reserved.* These bits should not be used.

Bit 3 **IRQ14 Pulse Mode Enable.** When bit 3 = ’1’ and bit 1 = ’0’ (level mode interrupt enable), the PCMCIA is enabled to use IRQ14 to support the PC Card with a pulse mode interrupt on IREQ#. In this condition other IRQ’s will still support edge-trigger interrupts from either card status change interrupts, or PC Card IREQ#. However if bit ’1’ = 1 (level mode interrupt enable), then bit 3 has no effect on IRQ14.

Bit 2 **Explicit Write Back CSC Ack.** Setting this bit = ’1’, will require an explicit write of a one to the Card Status Change Register bit which indicates an interrupting condition. When this bit = ’0’ (default state), the card status change interrupt is acknowledged by reading the Card Status Change Register, and the register bits are cleared upon a read.

Bit 1 **Level Mode Interrupt Enable.** When Bit 1 = ’1’, then all the IRQ outputs are configured to be active low level mode interrupts. In this mode, an IRQ will remain at the output logic state until there is either a card status change interrupt or an active I/O card interrupt is steered to that particular IRQ, at which time the IRQ output will go low. In the case of the interrupt being caused by an IREQ# active (low) from a PC Card, IRQ will remain low until IREQ# becomes inactive, then IRQX will be de-asserted. For an interrupt caused by the card status change, IRQ will remain low until the interrupt is acknowledged (service). OCE# serviced the IRQ output will go from a low to the output logic state.
When Bit 1 = ’0’ (Default state), the IRQ outputs will be configured to be low to high edge triggered interrupts. In this mode the IRQs will remain at the output logic state until a particular IRQ is enabled which will then cause the IRQ output to go low.
The output will remain low until a card status change interrupt or I/O card interrupt occurs to force the IRQ output high. For the interrupt caused by the IREQ# active (low) from a PC Card IRQX will remain high until IREQ# becomes inactive, then IRQ will be low again. In the state where the interrupt is caused by the card status change, it will remain high until the interrupt is acknowledged (serviced) when it will be forced low again. In ei-

ther of these cases, IRQs will remain low until they are disabled (interrupt and general control register). When disabled, IRQ will be at the output logic level state.

Bit 0 **Power Down.** The PCMCIA enters the CS# controlled power down when this bit = ’1’. In this case all the I/O memory windows are disabled and CS# is driven to inactive high. During CS# controlled power down, all the PCMCIA’s internal registers are inaccessible, outputs are disabled and minimum power consumption level is enabled. IRQs and RI_OUT# will however still be active to monitor the card detect and RI# status for any sign of a resume indication.

16.6.3 Interrupt Registers

16.6.3.1 Interrupt and General Control Register Index 0x03h

This register controls the interrupt steering for the PC Card I/O interrupt as well as providing general control of the PCMCIA. A RESETDRV clears all bits in this register, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

Bit 7 **Ring Indicate Enable.** When this bit = ’1’ and the PC Card Type bit (5) is set to a one (I/O PC Card), (STSCHG#/RI) from the I/O PC Card is used as a ring indicator signal and is passed through to the RI_OUT# output in of the PCMCIA.

When this bit = ’0’ and the PC Card Type bit (5) is set to a one (I/O PC Card), (STSCHG#/RI) from the I/O PC Card is used as the status change signal (STSCHG#). The current status of the signal is then available to be read from the Interface Status Register (01H), and this signal can be configured as a source for the card status change interrupt.

This bit has no function when the PC Card type bit is set to zero (memory PC Card).

Bit 7	Bit 5	Function
0	0	No function
0	1	STSCHG#
1	0	No function
1	1	RI# → RI_OUT#

Bit 6 **PC Card RESET#.** This is a software reset to the PC Card. Setting bit 6 = ’0’ activates the RESET signal to the PC Card. The RESET signal will remain active until bit 6 is set to a one.

Bit 5 **PC Card Type.** A I/O PC Card is selected by setting bit 5 = ’1’, thus enabling the PC Card interface multiplexer for routing of PC Card I/O signals. A Memory PC Card is selected by setting bit 5 = ’0’.

Bit 4 **INTR# Enable.** When bit 4 is set to a one it enables the card status change interrupt on the INTR# signal. However if bit 4 = ’0’, INTR# will always be inactive, and the card status change interrupt is steered to one of the IRQ lines in accordaCE# with bits 7-4 in the card status change interrupt configuration register. In CS# power down mode, INTR# will be inactive even if the bit is set and card status changes occur.

Bits 3-0 **IRQ Level Selection.** This selection is only applicable to I/O Cards. The following Table shows the functionality of bits 3-0 to control the redirection of the PC Card interrupt.

Table 16-12. PC Card IREQ# Interrupt Steering

IRQ Bit 3	IRQ 2	IRQ 1	IRQ 0	Function
0	0	0	0	IRQ Not Selected
0	0	0	1	*Reserved*
0	0	1	0	*Reserved*
0	0	1	1	IRQ3 Enabled
0	0	0	0	IRQ4 Enabled
0	1	0	1	IRQ5 Enabled
0	1	1	0	*Reserved*
0	1	1	1	IRQ7 Enabled
1	0	0	0	*Reserved*
1	0	0	1	IRQ10 Enabled
1	0	1	1	IRQ11 Enabled
1	1	0	0	IRQ12 Enabled
1	1	0	1	*Reserved*
1	1	1	0	IRQ14 Enabled
1	1	1	1	IRQ15 Enabled

16.6.3.2 CSC Interrupt Config. Register Index 0x05h

This register controls the interrupt steering of the card status change interrupt and the card status change interrupt enables. A RESETDRV clears all bits in this register, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

Bits 7-4 **CSC Interrupt Steering.** These bits select the re-direction of the card status change interrupt if the interrupt is not selected to the output.

Table 16-13. shows the routing of card status change interrupts..

Table 16-13. CSC Interrupt Steering

INTR# Enable Bit *	IRQ Bit 3	IRQ 2	IRQ 1	IRQ 0	Interrupt Request Level
0	0	0	0	0	IRQ Not Selected
0	0	0	0	1	*Reserved*
0	0	0	1	0	*Reserved*
0	0	0	1	1	IRQ3 Enabled
0	0	0	0	0	IRQ4 Enabled
0	0	1	0	1	IRQ5 Enabled
0	0	1	1	0	*Reserved*
0	0	1	1	1	IRQ7 Enabled
0	1	0	0	0	*Reserved*
0	1	0	0	1	IRQ10 Enabled
0	1	0	1	1	IRQ11 Enabled
0	1	1	0	0	IRQ12 Enabled
0	1	1	0	1	*Reserved*
0	1	1	1	0	IRQ14 Enabled
0	1	1	1	1	IRQ15 Enabled
1	X	X	X	X	Card Status Change Interrupt on INTR# pin

* Within Interrupt and General Control Register

Bit 3 **Card Detect Enable.**

When bit 3 is set to a one a card status change interrupt is enabled when a change has been detected on CD#1 or CD#2. Conversely if bit 3 is set to zero this disables the generation of a card status change interrupt when CD#1, CD#2 change state.

Bit 2 **Ready Enable.** When bit 2 = ’1’ a card status change interrupt is enabled when a low to high transition has been detected on Ready/NBusy. Conversely if bit 2 = ’0’, a card status change interrupt generation is disabled when a low to high transition has been detected on Ready/Busy#.

This bit is ignored when the interface is configured for I/O PC Cards.

Bit 1 **Battery Warning Enable.** When bit 1 is set to a one, a card status change interrupt is enabled when a battery warning condition is detected. Conversely if bit 1 is set to zero this disables the generation of a card status change interrupt when a battery warning condition has been detected.

This bit is ignored when the interface is configured for I/O PC Cards.

Bit 0 **Battery Dead Enable (STSCHG#).** For Memory PC Cards, the setting of Bit 0 to a one enables a card status change interrupt when a battery dead condition has been detected.

For I/O PC Cards, the setting of Bit 0 to a one enables the PCMCIA to generate a card status change interrupt providing (STSCHG#/RI#) has been pulled low by the I/O PC Card and also assuming that the Ring Indicate Enable bit = ‘0’ (Interrupt and General Control Register).
Setting Bit 0 = ’0’ disables the generation of a card status interrupt.
This bit is ignored when the interface is configured for I/O PC Cards and the Ring Enable bit = ‘1’ (Interrupt and General Control Register).

16.6.4 I/O Registers

16.6.4.1 I/O Control Register Index0x07h

These registers indicate the I/O configuration for I/O window 0 and 1. This information is read from the PC Card’s card information structure. Dynamic bus sizing on a cycle by cycle basis is implemented to the PC Card interface if the source of IOCS16# is the PC Card (determined by the IOCS16# source bit). In order to be compatible with a variety of software/hardware implementations, (IDE interface for example), it is necessary for the PC Card to decode two consecutive I/O addresses to determine the cycle data width.

Also in order to meet the system bus timings, this type of PC Card must decode the address lines SA[9:0]

prior to the card enable signal becoming active at the interface. The card decodes the address and responds to a 16 bit cycle by enabling IOS16#. The PCMCIA qualifies IOS16# with the card enable signals in order to generate IOCS16# to the system bus.

The individual bits in these registers set the data path size and select zero wait states for the appropriate bus access. Additionally they determine the system bus signal IOCS16# as described above.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = 1), and the Disable Resume RESETDRV bit = 1 in the slot Power and RESETDRV Control Register.

Bit 7 **I/O Window 1 Wait State.** When Bit 7 is set to a one, 16 bit system accesses occur with 1 wait state (4 System Clocks). A standard 16 bit I/O cycle completes in 3 System Clocks with IOCHRDY high.

Bit 6 **I/O Window 1 Zero Wait State.** If bit 6 is set to a one, 8 bit system I/O accesses complete in 3 System Clocks, with ZEROWS# asserted to the system bus. However if bit 6 is set to a zero, the 8 bit system I/O access will complete in 6 System Clocks when the PC Card asserts no WAIT#, or more system clock cycles when WAIT# is asserted (cause IOCHRDY de-asserted). WAIT# will override bit 6. This bit has no meaning for a 16 bit I/O access.
16 bit I/O accesses will always occur with either a 3 System Clock standard cycle when WAIT# is not active, or more cycles when WAIT# is active, or Bit 7 is set.

Bit 5 **I/O Window 1 IOCS16# Source.** When bit 5 = ’0’ PCMCIA generates IOCS16# based on the value of the data size bit. However if bit 5 = ’1’, the PCMCIA generates IOCS16# based on IOS16# from the PC Card and the data size bit is ignored.

Bits 3-0 **I/O Window 0**. The above descriptions for bits 7-4 describing Window 1, are applicable to Window 0 as follows:

Bit 3 - **I/O Window 0 Wait State**

Bit 2 - **I/O Window 0 Zero Wait State**

Bit 1 - **I/O Window 0 IOCS16# Source**

Bit 0 - **I/O Window 0 Data Size**

16.6.4.2 I/O Address 0 Start Low Byte Register Index 0x08h

This register containss the low order address bits used to determine the start address of I/O address window 0. This provides a minimum 1 byte window for I/O address window 0.

Bits 7-0 **SA7-0.** These bits are to contain the low byte start address SA[7:0] for Window 0

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot A Powerand RESETDRV Control Register.

16.6.4.3 I/O Address 0 Start High Byte Register Index 0x09h

This register contains the high order address bits used to determine the start address of I/O address window 0.

Bits 7-0 **SA15-8.** These bits are to contain the high byte start address SA15-8 for Window 0.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

16.6.4.4 I/O Address 0 Stop Low Byte Register Index 0x0Ah

This register contains the low order address bits used to determine the stop address of I/O address window 0. This provides a minimum 1 byte window for I/O address window 0.
Note: No attempt should be made to overlay the I/O window over the top of the PCMCIA’s registers. This will cause the PCMCIA access type to change.

Bits 7-0 **SA7-0.** These bits are to contain the low byte low byte stop address SA7-0 for Window 0.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

16.6.4.5 I/O Address 0 Stop High Byte Register Index 0x0Bh

This register contains the high order address bits used to determine the stop address of I/O address window 0.

Bits 7-0 **SA15-8.** These bits are to contain the high byte stop address SA15-8 for Window 0.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

16.6.4.6 I/O Address 1 Start Low Byte Register Index 0x0Ch

This register contains the low order address bits used to determine the start address of I/O address window 1. This provides a minimum 1 byte window for I/O address window 1.

Bits 7-0 **SA7-0.** These bits are to contain the low byte start address SA7-0 for Window 1.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = '1'), and the Disable Resume RESETDRV bit = '1' in the slot Power and RESETDRV Control Register.

16.6.4.7 I/O Address 1 Start High Byte Register Index 0x0Dh

These registers contain the high order address bits used to determine the start address of I/O address window 1.

Bits 7-0 **SA15-8.** These bits are to contain the high byte start address SA[15:8] for Window 1.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = '1'), and the Disable Resume RESETDRV bit = '1' in the slot Power and RESETDRV Control Register.

16.6.4.8 I/O Address 1 Stop Low Byte Register Index 0x0Eh

This register contains the low order address bits used to determine the stop address of I/O address window 1. This provides a minimum 1 byte window for I/O address window 1.
Note: No attempt should be made to overlay the I/O window over the top of the PCMCIA's registers. This will cause the PCMCIA access type to change.

Bits 7-0 **SA7-0.** These bits are to contain the low byte stop address SA7-0 for Window 1.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = '1'), and the Disable Resume RESETDRV bit = '1' in the slot Power and RESETDRV Control Register.

16.6.4.9 I/O Address 1 Stop High Byte Register Index 0x0Fh

This register contains the high order address bits used to determine the stop address of I/O address window 1.

Bits 7-0 SA15-8. These bits are to contain the high byte stop address SA15-8 for Window 1.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = '1'), and the Disable Resume RESETDRV bit = '1' in the slot Power and RESETDRV Control Register.

16.6.5 Memory Registers

16.6.5.1 Syst.Mem. Address 0 Start Low Byte Register Index 0x10h

This register contains the low order address mapping bits used to determine the start address of the corresponding system memory address mapping window. This provides a minimum memory mapping window of 4K Bytes.

A memory PC Card is selected when the following conditions are satisfied:

1. The system memory address mapping window is enabled.

2. The (ISA) system memory address is greater than or equal to the system memory address mapping start register (high and low byte).

3. The (ISA) system memory address is less than or equal to the system memory address mapping stop register (high and low byte).

The system memory address mapping windows can all be configured by software, either independently or used together to perform mapping for special memory mapping requirements. Examples of such special requirements are LIM/EMS (Lotus-Intel-Microsoft/Extended Memory Specification), or XIP (Execute in Place).

Note: A memory window cannot be set up below the first 64K of address space.

Bits 7-0 **SA7-0.** These bits are to contain the system memory low byte start address A[19:12] for Window 0.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = '1'), and the Disable Resume RESETDRV bit = '1' in the slot Power and RESETDRV Control Register.

16.6.5.2 Syst.Mem. Address 0 Start High Byte Register Index 0x11h

This register contains the high order address mapping bits used to determine the start address of the corresponding system memory address mapping window. Each system memory window has an associated data path size which is controlled by a bit in this register. Accesses to each system memory window have the potential to occur with zero additional wait states also controlled by a bit in this register.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

Bit 7 **Data Size.** When bit 7 = ’0’ an 8 bit memory data path to the PC Card is selected. If bit 7 = ’1’ then a 16 bit memory data path to the PC Card is selected.

Bit 6 **Zero Wait State.** If bit 6 = ’1’, 8 or 16 bit system memory accesses complete in 3 or 2 System Clocks respectively, providing ZEROWS# is asserted to the system bus. However if bit 6 = ’0’, the 8 or 16 bit memory access will complete in 6 or 3 System Clocks respectively with IOCHRDY asserted high.
In the event of IOCHRDY becoming de-asserted by internal wait state generation, or WAIT#, then setting bit 6 = ’0’ will cause 16 bit memory cycles to complete in more than 3 System Clocks.
In the condition where IOCHRDY becomes de-asserted by WAIT# while bit 6 = ’0’, the 8 bit memory cycles will complete in more than 6 System Clocks. Only WAIT# can override bit 6.
When bit 6 = ’1’ in this register, then the ZEROWS# output will be held high for accesses to an 8 bit system memory window with both A0 and SBHE# = ’0’.
Note: A logic low on IOCHRDY, either caused by an internal wait state generator or by WAIT# will force the ZEROWS# output high.

Bits 5-4 **Scratch Bits.** Bits 5 and 4 are available to the designer for general purpose register storage and retrieval.

Bits 3-0 **Syst. Mem. Window Start Address.** Bits 3 to 0 are high order address bits A[23:20] used to determine the start address of the corresponding system memory address mapping window.

16.6.5.3 Syst.Mem. Address 0 Stop Low Byte Register Index 0x12h

This register contains the low order address bits used to determine the stop address of the corresponding system memory address mapping window. This provides a minimum memory mapping window of 4 K bytes.

Bits 7-0 **SA19-12.** These bits are to contain the describe the required System Memory Window low byte stop address A19-12 for Window 0.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

16.6.5.4 Syst.Mem. Address 0 Stop High Byte Register Index 0x13h

This register contains the high order address mapping bits used to determine the stop address of the corresponding system memory address mapping window. All system memory windows have the ability to extend a 16 bit system bus cycle by inserting wait states. Two bits in each of these registers select the number of wait states for a 16 bit access to the system memory window.

A RESETDRV clears all bits in these registers, unless RESETDRV is a result of a Host System resume (PWRGOOD = ’1’), and the Disable Resume RESETDRV bit = ’1’ in the slot Power and RESETDRV Control Register.

Bits 7-6 **Wait State(s) Select.** Bits 7 and 6 determine the number of additional wait states for a 16 bit access to the system memory window. The internal wait state generator will not cause additional wait states to be inserted for an 8-bit system access, even if both bits = ’1’, because IOCHRDY will be pulled high by the PCMCIA before the system samples IOCHRDY. If the PC Card supports WAIT#, wait states will be generated by the PC Card asserting WAIT#. Bits 7 and 6 should be set to zero to disable the internal wait state generator. Table 16-14. shows the wait states that can be selected.

Table 16-14. Wait State Selection

Bit 7	Bit 6	Number of Additional Wait States	System Clocks per Access
0	0	Standard 16 Bit Cycle	3
0	1	1	4
1	0	2	5
1	1	3	6

Bits 5-4 *Reserved*. Bits 5 and 4 are reserved for internal use.

Bits 3-0 **SA23-SA20. Syst. Mem. Window Stop Address.** Bits 3 to 0 provide the high order address bits used to determine the stop address of the corresponding system memory address mapping window.

16.6.5.5 Card Mem. Offset Address 0 Low Byte Reg. Index 0x14h

This register contains the low order address bits which are added to the system address bits

A[19:12] to generate the memory address for the PC Card.

Bits 7-0 **OA19-OA12** These bits describe the required Card Memory Offset Address OA[19:12] for Window 0.

16.6.5.6CardMem.OffsetAddress0HighByteReg. Index 0x15h

This register contains the high order address bits which are added to the system address bits A23-20 to generate the memory address for the Pc Card. The software write protect of the PC Card memory for the corresponding system memory window is controlled by this register. Additionally the register also controls if the corresponding system memory window is mapped to attribute or common memory on the PC Card.

Bit 7 **Write Protect.** If bit 7 = ’1’, write operations to the PC Card through the corresponding system memory window are inhibited. Conversely if bit 7 = ’0’ then these operations are enabled. It should be noted that the WP switch on the memory card by itself will not block the memory write cycle as it only sets the Memory Write Protect bit in the Interface Status Register.

Bit 6 **Reg. Active.** When bit 6 = ’1’, accesses to the system memory window will result in attribute memory on the PC Card being accessed by asserting REG# to Low. However if bit 6 = ’0’, accesses to the system memory will result in common memory on the PC Card being accessed by the driving of REG# to High.

Bits 5-0 **OA25-20 Card Mem.Offset.** Bits 5 and 4 will be added to the system address bits A23-20 to generate the memory address for the PC Card.

16.6.5.7 Syst. Mem. Addresses 1-4 Registers

The mapping registers for System Memory address 1 to 4 are identical to those described for address 0.

Refer to Table 16-6. for the appropriate register addresses.

16.7 CARD BUS REGISTER DESCRIPTION

This chapter details the register in the Cardbus module.

The following registers are accessed by setting a base index in the CardBus Index Register of the PCCard Bridge (Index 010h).

16.7.1 Socket Event Register Index 0x00h

This is a Read/Write register.

Bit 31-4 *Reserved*, These bits are set to 0.

Bit 3 **EV_PWCYCLE.** Power cycle completed (Power up and power down)). Bit cleared by writing to one.

Bit 2 **EV_CCD2.** Set when the CardBus Card Detect CCD2# bit in the Socket Present State change state. Bit cleared by writing to one.

Bit 1 **EV_CCD1.** Set when the CardBus Card Detect CCD1# bit in the Socket Present State change state. Bit cleared by writing to one.

Bit 0 **EV_CSTSCHG.** Set when the CSTSCHG bit in the Socket Present State change state. Bit cleared by writing to one.

Each change on these bits will generate an interrupt if not masked.

16.7.2 Socket Event Mask Register Index 0x04h

This is a Read/Write register

Bit 31-4 *Reserved***,** This bit is hardwired to 0.

Bit 3 **MASK_EV_PWCYCLE.** Power cycle completed interrupt enabled if set to one.

Bit 2-1 **MASK_EV_CCD.** The bridge is enabled to generate an interrupt when CCD1# or CCD2# pins change state, if set to one.

Bit 0 **MASK_EV_CSTSCHG.** The bridge is enabled to generate an interrupt when CSTSCHG is set in the socket event register, if set to one.

These different interrupts are disabled, if these bits are reset to zero.

16.7.3 Socket Force Event Register Index 0xCh

This a Write only register

Bit 31-15 *Reserved***.** These bits are hardwired to 0.

Bit 14 **CVStest.** Socked interrogation (using CVS/CCD pins) performed by the card type detection mechanism, by writing to one. If the nVcard bits have previously been forced, the bridge is re-enabled to apply Vcc to the socket.

Bit 13 **YVCARD set** in Socket Present State Register by writing to one. yVcard cleared in Socket Present State Register by writing to zero. When set the bridge can not apply Vcc to the socket until the CVStest bit is set.

Bit 12 **XVCARD set** in Socket Present State Register by writing to one. xVcard cleared in Socket Present State Register by writing to zero. When set the bridge can not apply Vcc to the socket until the CVStest bit is set.

Bit 11 **3V CARD set** in Socket Present State Register by writing to one. 3Vcard cleared in Socket Present State Register by writing to zero.When set the bridge can not apply Vcc to the socket until the CVStest bit is set.

Bit 10 **5V CARD set** in Socket Present State Register by writing to one. 5Vcard cleared in Socket Present State Register by writing to zero. When set the bridge can not apply Vcc to the socket until the CVStest bit is set.

Bit 9 **BADVCCREQ set** in Socket Present State Register by writing to one. BadVccReq cleared in Socket Present State Register by writing to zero.

Bit 8 **DataLost** set in Socket Present State Register by writing to one.DataLost cleared in Socket Present State Register by writing to zero.

Bit 7 **NotAcard** set in Socket Present State Register by writing to one. NotAcard cleared in Socket Present State Register by writing to zero.

Bit 6 *Reserved* This bit is hardwired to 0.

Bit 5 **CardBus** card set in Socket Present State Register by writing to one. CardBus card cleared in Socket Present State Register by writing to zero.

Bit 4 **16-bit card** set in Socket Present State Register by writing to one. 16-bit card cleared in Socket Present State Register by writing to zero.

Bit 3 **Power cycle** set in Socket Event Register by writing to one.

Bit 2 **CCD2#** set in Socket Event Register by writing to one.

Bit 1 **CCD1#** set in Socket Event Register by writing to one.

Bit 0 **CSTSCHG** set in Socket Event Register by writing to one.

16.7.4 Socket Present State Register Index 0x08h

This is a Read only register.

Bit 31 **PRES_ST_YVSOCK.** Bridge can supply Vcc = y.y v.

Bit 30 **PRES_ST_XVSOCK.** Bridge can supply Vcc = x.x v.

Bit 29 **PRES_ST_3VSOCK.** Bridge can supply Vcc = 3.3 v.

Bit 28 **PRES_ST_5VSOCK.** Bridge can supply Vcc = 5 v.

Bits 27-14 *Reserved***.** These bits are hardwired to 0.

Bit 13 **PRES_ST_YVCARD.** Card can accept initial Vcc = y.y v.

Bit 12 **PRES_ST_XVCARD.** Card can accept initial Vcc = x.x v.

Bit 11 **PRES_ST_3VCARD.** Card can accept initial Vcc = 3.3 v.

Bit 10 **PRES_ST_5VCARD.** Card can accept initial Vcc = 5 v.

Bit 9 **PRES_ST_BADVCCREQ.** Illegal initial Vcc value requested by SW (detected by the bridge).

Bit 8 **PRES_ST_DATALOST.** Loss of data caused by abrupt card removal (detected by the bridge).

Bit 7 **PRES_ST_NOTACARD.** Cannot determine the Card type.

Bit 6 *Reserved***.** This bit is hardwired to 0.

Bit 5 **PRES_ST_CBCARD.** CardBus card detected. (switch the mux)

Bit 4 **PRES_ST_16CARD.** 16-Bit card detected. (switch the mux)

Bit 3 **PRES_ST_PWCYCLE.** Card Power up cycle completed.

Bit 2 **PRES_ST_CCD2_.** Card is not detected (CardBus Card Detect 2 signal =1).

Bit 1 **PRES_ST_CCD1_.** Card is not detected (CardBus Card Detect 1 signal =1).

Bit 0 **PRES_ST_CSTSCHG.** CardBus Status Change signal asserted.

16.7.5 Socket Control Register Index 0x10h

This is a Read/Write register

Bits 31-7 *Reserved* These registers are hardwired to 0.

Bit 6-4 **CTRL_VCC (Vcc control).** Vcc requested to be applied to the socket.

Bit 6	Bit 5	Bit 4	Description
0	0	0	power off
0	0	1	*Reserved*
0	1	0	5V
0	1	1	3.3V
1	0	0	*Reserved*
1	0	1	*Reserved*
1	1	0	*Reserved*
1	1	1	*Reserved*

Bit 3 *Reserved* This bit is hardwired to 0.

Bits 2-0 **CTRL_VPP (Vpp control).** Vpp requested to be applied to the socket

Bit 2	Bit 1	Bit 0	Description
0	0	0	power off
0	0	1	12V
0	1	0	5V
0	1	1	3.3V
1	0	0	*Reserved*
1	0	1	*Reserved*
1	1	0	*Reserved*
1	1	1	*Reserved*

The PCMCIA provides the socket with 5 power control pins. These are physically used to control the power supplies (Vcc, Vpp1 and Vpp2) to the PC Cards. The Power and RESETDRV Control Register (R/W) provides the control of PC Card Power.

Control Bit	**Name**	**Description**
0	Vpp1_EN0	Vpp1 control, bit 0
1	Vpp1_EN1	Vpp1 control, bit 1
2	Vpp2_EN0	Vpp2 control, bit 0
3	Vpp2_EN1	Vpp2 control, bit 1
4	NVcc_EN	Master Enable

Control of the power supplies Vcc and Vpp only (Vpp=Vpp1=Vpp2, as the Cardbus socket management)

-The Power and RESETDRV Control Register (R/W) provides the control of PC Card Power.

Control Bit	Name	Description
0	Vpp_vcc	Vpp control, bit 0
1	Vpp_pgm	Vpp control, bit 1
2	Vcc5en	Vcc control, bit 0
3	Vcc3en	Vcc control, bit 1
4	------	Master Enable (not connected)

-The Voltage Control Pin Interpretations using PCMCIA is:

Vpp2 _En0	Vpp2_ EN1	Card Vcc	Vpp1_E N0	Vpp1_ EN1	Card Vpp
0	0	Disabled	0	0	Disabled
			0	1	Disabled
			1	0	Disabled
			1	1	Ground
0	1	3.3V	0	0	Disabled
			0	1	3.3V
			1	0	12V
			1	1	Ground
1	0	5.0V	0	0	Disabled
			0	1	5V
			1	0	12V
			1	1	Ground
1	1	3.3V Reseved	0	0	Disabled
			0	1	3.3V
			1	0	5V
			1	1	Ground

16.7.6 DMA DREQ routing Configuration Index 0x92h

This register controls the selection of the DMA channel for PCMCIA and Parallel Port DMA request. For example see Figure 16-15.

Bit 6 *Reserved* This bit is hardwired to 0

Bit 5 **CardBus Card Set** in Socket Present State Register by writing to 1. CardBus Card cleared in Socket Present State Register by writing to zero.

Bit 4 **16-Bit Card set** in Socket Present Registerby writing to one. 16-Bit Card cleared in Socket Present State Register by writing to zero.

Bit 3 **Power Cycle** set in Socket Event Register by writing to one

Bit 2 **CCD2#** set in Socket Event Register by writing to 1.

Bit 1 **CCD1#** set in Socket Event Register by writing to 1.

Bit 0 **CSTSCHG** set in Socket Event Registerby writing to 1.

Table 16-15. PCMCIA Parallel Port Request Example

Re-served	*Re-served*	*Re-served*	PCMCIA DMA2	PCMCIA DMA 1	PCMCIA DMA0	PP DMA1	PP DMA0	PCMCIA DMA Chan-nel	P Port DMA Channel
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	2
0	0	0	0	0	0	1	1	0	3
0	0	0	0	0	1	0	0	1	0
0	0	0	0	1	0	0	0	2	0
0	0	0	0	1	1	0	0	3	0
0	0	0	1	0	1	0	0	5	0
0	0	0	1	1	0	0	0	6	0
0	0	0	1	1	1	0	0	7	0
Other								1	0

17 LOCAL BUS INTERFACE

17.1 INTRODUCTION

The Local Bus interface of the PC Industrial provides a 66MHz, low latency bus to external peripheral cards. The Local Bus may operate in asychronous or synchronous modes through the 22 bit address and 16 bit data bus.

The Local Bus interface supports up to two memory devices and four I/O devices. It can support up to 4 MBytes of memory for each of the memory chip selects and from 4 Byte to 1 KByte of I/O space for each of the I/O devices. All the chip select timings are individually programmable. This interface can be accessed only by the CPU.

The memory addresses are predefined for the memory chip selects. The first bank of the memory is intended to be used as a boot device.

The starting address for each IO chip select is programmable at 4 Byte boundary. The access range for each of the chip select is also programmable. The size varies from 4 Bytes to 1 KByte.

17.2 LOCAL BUS REGISTERS

The Local Bus configuration registers can be categorized into 3 groups

1. Address Decode Registers
2. Timing Registers
3. Control Register

All the registers,except the Control Register, are 16 bit wide. These registers are accessible only to the CPU. All the registers are accessed through I/O Port 22 and Port 23. Port 22 is used as the index to the register bank and Port 23 is used as the data port. This way CPU can access only 8 bits of register data at a time, so two accesses are required to completely read or write the 16 bit registers. The lower and upper halves of the 16 bit registers have the same index values. First access after reset with the given index will map to the lower Byte of the register. The next access with the same index will access the upper Byte. For the 8 bit registers (Control Register Index 1Ch and IOWIDTH Register Index 1Eh) data must be written twice as they are seen as 16 bit registers.

There are two other ways in which these registers can be accessed. These approaches are described in Control Register Section 17.5.

Table 17-1. Local Bus Register Indeces

Name	Index	Function
IOAREG0	10h	Address Decode
IOAREG1	11h	
IOAREG2	12h	
IOAREG3	13h	
IOMREG0	14h	
IOMREG1	15h	
TIMEBANK0	16h	Timing
TIMEBANK1	17h	
TIMEIO0	18h	
TIMEIO1	19h	
TIMEIO2	1Ah	
TIMEIO3	1Bh	
CONTROL	1Ch	Control
IOWIDTH	1Eh	

17.3 LOCAL BUS ADDRESS DECODE REGISTERS

17.3.1 IOAREG, IO Slot Base Address Regs.

This 16 bit register defines the starting address (4 Bytes or 32 bit WORD) and I/O address spaces mapped on the local bus. The base addresses for I/O slots 0 to 3 are specified in register IOAREG0, IOAREG1, IOAREG2 and IOAREG3 respectively. Any accesses that hit onto the address range defined by this register and its mask in the corresponding IOMREG register will prevent the cycle being forwarded onto the PCI bus.

Bits 15-2 Starting Address aligned to 4 I/O locations.

Bits 1-0 *Reserved.*

The 2 memory address spaces (for FLASH memory devices) have a fixed range (FFC00000h-FFFFFFFFh for the Boot device and FEC00000h-FEFFFFFFh for the second device).

17.3.2 IOMREG, IO Slot Mask Registers

The address mask register IOMREG0 and IOMEREG1 define the size of the I/O slots. The 8 bit address mask for each slot will mask the address that we do not want to compare during the address decoding process. The 8 bit mask will filter bit 9:2 of the starting address specified in the corresponding IOAREG allowing a selection of 4, 8, 16, 32, 64, 128, 256, 512 or 1K continuous or non continuous locations.

IOMREG0

Bits 15-8 **Address mask for I/O Slot 1.**

Bits 7-0 **Address mask for I/O Slot 0.**

IOMREG1

Bits 15-8 **Address mask for I/O Slot 3.**

Bits 7-0 **Address mask for I/O Slot 2.**

17.4 LOCAL BUS TIMING REGISTERS

17.4.1 TIMEBANK, Memory Timing Templates

TIMEBANK0 and TIMEBANK1 define the timing template for accessing Flash memory bank 0 and 1 respectively. The timings are programmed with reference to the host clock period as a time unit.

Timing for FLASH devices should take into account access times of 60ns and 150ns for the Boot Memory.

Bit 15 **Discard the Setup and Hold parameters for read operations**

Bit 14 **Use asyncronous ready signal for command termination.** The asynchronous ready enable bit sets the local bus logic to wait for an external ready signal before closing the current cycle.

Bits 13-11 **Command Hold time**

Bits 10-3 **Command Active time**

Bits 2-0 **Command Setup time**

17.4.2 TIMEIO, I/O Timing Templates

TIMEIO0, TIMEIO1, TIMEIO2 and TIMEIO3 registers define the timing template for accessing devices in I/O Slots 0,1,2 and 3 respectively. The timings are programmed with reference to the host clock period as a time unit.

Bit 15 **Discard the Setup and Hold parameters for read operations**

Bit 14 **Use asyncronous ready signal for command termination**

Bits 13-11 **Command Hold time**

Bits 10-3 **Command Active time**

Bits 2-0 **Command Setup time**

17.5 LOCAL BUS CONTROL REGISTER

17.5.1 Control Register

The control register is 8-bit wide.

Bits 7-6 **Register Access Map.**

Bits 7-6	Register Access
00	Access with same index will alternately map to lower and upper and lower Bytes of the 16 bit register as decribed in Section 17.2
01	Register accesses will always map to the lower Byte
10	Register accesses will always map to the upper Byte
11	Access with same index will alternately map to lower and upper and lower Bytes of the 16 bit register as decribed in Section 17.2

Bit 5 **32bit access to Flash memory Enable.** Setting this bit enables 32-bit access to both bank1 and bank0

Bit 4 **Cache Enable for Bank1**

Bit 3 **Cache Enable for Bank0**

Bit 2 **Write Enable for Bank1**

Bit 1 **Write Enable for Bank0**

Bit 0 **Real Mode Boot Access Enable.**

When set this bit enables boot access in Real Mode by mapping 000A0000h to FFFA0000h and 000FFFFFh to FFFFFFFFh

This register defaults to 00h on reset.

17.5.2 IO Width Register

This is an 8 bit register whose 4 less significant bits are used to tell the localbus if an 8 bit peripheral is attached to one of the 4 I/O slots

Bits 7-4, *Reserved.*

Bit 3, If set to 1, the I/O 3 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

Bit 2, If set to 1, the I/O 2 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

Bit 1, If set to 1, the I/O 1 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

Bit 0, If set to 1, the I/O 0 is a 16 bit peripheral, if set to 0 it is an 8 bit peripheral.

This register defaults to 00h on reset.

17.6 CHIP SELECT MEMORY MAP

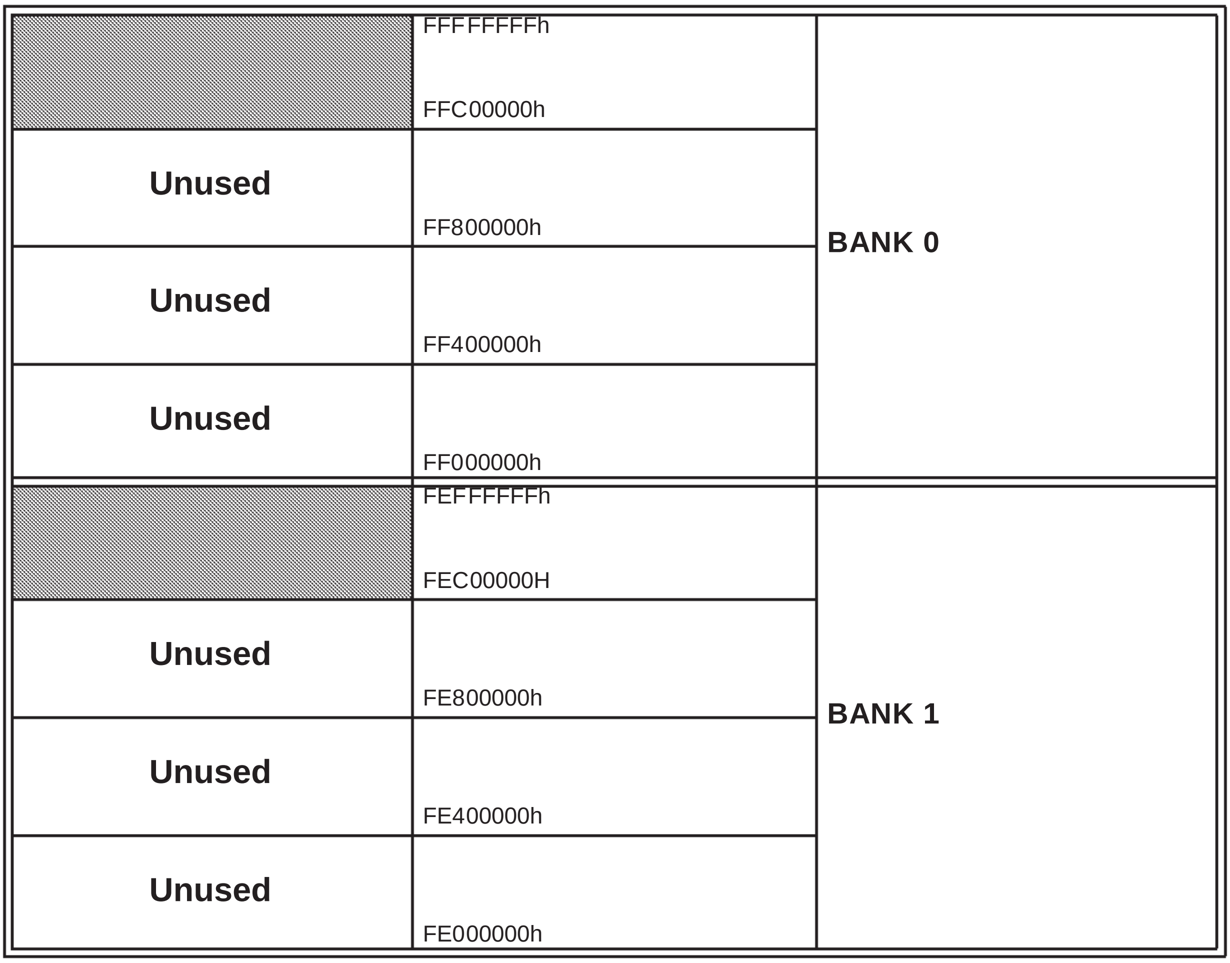
The address mapping supported by the local Bus interface is summarized by Figure 17-2. Memory address ranges are mapped at fixed addresses while the I/O devices can be mapped from 1 DW (double word) to 256 DW inside a 16 MByte segment.

Table 17-2. Local Bus Address Mapping

Flash Memory Chip Select	**Bank 0**		**Bank 1**	
Address Range	FFC00000h-FFFFFFFFh		FEC00000h-FEFFFFFFh	
Address Space	4 MBytes		4 MBytes	
Boot Address Space (Real Mode)	000A0000h-000FFFFFH		-	
I/O Control Chip Select	IOCS#0	IOCS#2	IOCS#3	IOCS#4
Address Range	0000h-FFFFh	0000h-FFFFh	0000h-FFFFh	0000h-FFFFh
Address Space	4 Bytes - 1KByte	4 Bytes - 1KByte	4 Bytes - 1KByte	4 Bytes - 1KByte
Address Mask	8-bit, Selected from 4, 8, 16, 32, 64, 128, 256, 512, 1K			

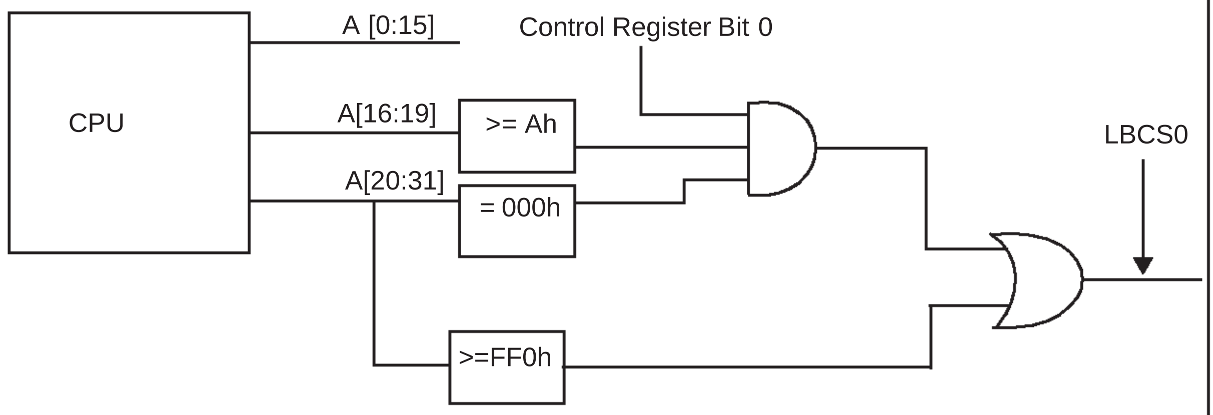
The memory mapping described in Table 17-2. above is illustrated in Figure 17-1

Figure 17-1. Local Bus Address Mapping Diagram



Memory access to the local bus is based on the following logic scheme see Figure 17-2.;

Figure 17-2. Memory Bank 0 Access Logic



LBCS0 = 0, the CPU accesses to memory are performed on either the DRAM or PCI interfaces.

LBCS0 = 1, the CPU accesses to memory are performed on the Local Bus memory Bank 0.

17.7 UPDATE HISTORY FOR LOCAL BUS CHAPTER

The following changes have been made to the Local Bus Chapter on the 17/08/99.

Section	Change	Text
Tab 17-2.	Replaced	Figure 16-1 with Table 16.2
17.1	Removed	“~~Other than the chip selects, the devices share a set of common read/write control lines. All the memory devices use MEMRD# and MEMWR# to control read and write cycles and I/O devices use IORD# and IOWR#.~~”
17.2	Removed	“~~This approach is faster compared to assigning seperate index to upper and lower Bytes of the register because it avoids one extra write to the Port 23 index register. Moreover this technique requires a lesser number of index values.~~ “
17.3.1	Removed	“~~The filtered addresses will be compared and a local bus start signal will be issued along with CS0.~~”
17.4.1	Removed	“~~This 8 bit register defines the correct timings of the I/O cycle that corresponds with the address space defined by CS0.~~”
Fig 17-1	Added	Figure 17.1 added
Fig 17-2	Added	Figure 17.2 added

UPDATE HISTORY FOR LOCAL BUS CHAPTER

18 KEYBOARD / MOUSE CONTROLLER

18.1 INTRODUCTION

Keyboard/Mouse Interface provides two PS/2 compatible serial interfaces with the industry standard 8042 compatible programming interface.

Both the keyboard and mouse are supported, with the non-used function disabled via control of the KBCLK or MCLK outputs.

Full standard operation is provided, including input and output buffering, hardware parity check, time-out check, A20# control and complete scan code conversion for the AT keyboard.

18.2 IO PINS

KBCLK, *Keyboard Clock line.* Keyboard data is latched by the controller on each negative clock edge produced on this pin. The keyboard can be disabled by pulling this pin low by software control.

KBDATA, *Keyboard Data Line.* 11-bits of data are shifted serially through this line when data is being transferred. Data is synchronised to KBCLK.

MCLK, *Mouse Clock line.* Mouse data is latched by the controller on each negative clock edge produced on this pin. The mouse can be disabled by pulling this pin low by software control.

MDATA, *Mouse Data Line.* 11-bits of data are shifted serially through this line when data is being transferred. Data is synchronised to MCLK.

The controller pins should be connected to the 14.318 MHz Series Cut Quartz Crystal oscilator to function properly.

18.3 FUNCTIONAL DESCRIPTION

18.3.1 Basic Operation During Keyboard Write

Mouse interface is disabled by pulling the Mouse clock line low
Keyboard clock line is pulled low for more than 60us so that if any keyboard read operation is in progress it is terminated.
When keyboard clock line is low the keyboard data line is pulled low.
After 60 us the keyboard clock line is pulled high and the controller expects that the keyboard starts the clock within 20ms. Failing to do so will result in a timeout .
When the keyboard starts the clock on each falling edge of the clock the controller shifts out the data. The data should be transmitted within 2ms. Failing to do so will result in a timeout.
Once a write is over (i.e. 11 bits are transmitted), the controller waits for the keyboard to send an appropiate acknowledgement byte within 20ms. Failing to do so will result in a timeout.
The received acknowledgement is placed in the output buffer and OBF (Output Buffer Full flag) is set and the system is informed about the availability of the response through IRQ1.
During this time the mouse interface is kept disabled.
When OBF is set both the interfaces are disabled and will be enabled only when the system reads the output buffer i.e, OBF is reset.

18.3.2 Basic Operation During Keyboard Read

Initially assume that both the interfaces are enabled (i.e., both Keyboard and Mouse Clock lines are high)
When Keyboard wants to send data, it pulls the Keyboard data line low and checks the status of Keyboard clock line. If the keyboard clock line is low, it implies that the keyboard interface is disabled and the keyboard makes its data line high. If the clock line is high the keyboard starts driving the clock line.
Mouse interface is disabled by the controller by pulling the Mouse clock low.
With each rising edge of the clock keyboard transmits one data bit on the data line. The data should be transmitted within 2 ms. Failing to do so will result in a timeout.
This data is latched by the controller on each negative clock edge.

After the transfer of all the 11 bits, the controller checks for the parity of the received byte. If the parity error is found, the Output Buffer register is loaded with ‘FFh’ and OBF flag is set.

If there is no parity error and Keyboard is not locked, the data sent by the Keyboard (KSCAN code) is first converted into System Scan code and then loaded into the Output buffer and OBF flag is set. However, if Kscan code to System scan code conversion is disabled, the data is placed in the output buffer without any conversion.

The system is informed about the availability of the new byte through IRQ1.

When OBF is set both the interfaces are disabled and will be enabled only when the system reads the output buffer i.e, OBF is reset.

18.3.3 Basic Operation During Mouse Write

Mouse Write is exactly similar to Keyboard Write. In the description of Keyboard Write, interchanging Mouse Clock and Keyboard Clock, Mouse Data and Keyboard Data and replacing IRQ1 with IRQ12 will give the protocol of Mouse Write.

18.3.4 Basic Operation During Mouse Read

Mouse Read is exactly similar to Keyboard Read. In the description of Keyboard Read, interchanging Mouse Clock and Keyboard Clock, Mouse Data and Keyboard Data and replacing IRQ1 with IRQ12 will give the protocol of Mouse Read. Another difference in case of Mouse Read is that there is no conversion of data sent. In the case of Mouse AUXBUF (Auxiliary Buffer Full) bit is also set by the controller.

Summary of Keyboard Mouse differences:

Address	Clock line	Data line	Int
Keyboard	KBCLK	KBDATA	IRQ1
Mouse	MCLK	MDATA	IRQ12

18.4 KEYBOARD/MOUSE CONTROLLER REGISTERS

The keyboard/mouse controller is accessed through IO Port 60h and 64h.

Address	Function	R/W#
60h	Output Buffer	Write
60h	Input Buffer	Read
64h	Command Register	Write
64h	Status Register	Read

18.4.1 Output Buffer

This 8-bit register is written when a write operation is made to IO address 60h. The data written is the can be a command to the keyboard or to the mouse.

18.4.2 Input Buffer

This 8-bit register is read when a read operation is made to IO address 60h. The data read is the keyboard scan code value for keyboard operation, or the raw keyboard code if enabled, or the raw mouse data.

18.4.3 Command Register

This 8-bit register is written when a write operation is made to IO address 64h.

18.4.4 Status Register

This 8-bit register is read when a read operation is made to IO address 64h.

Bit 7 **PARE** *Parity Error Detected.* This bit is set if a parity error has been detected.

Bit 6 **TIM** *Timeout Detected.* This bit is set if a time-out error has been detected.

Bit 5 **BUSY** *Busy.* This bit indicates whether the keyboard/mouse controller is busy or available. This bit should be checked before any operation.

Bit 4 *Reserved.*

Bit 3 **MSE** *Mouse Enable.* This bit indicates that the mouse is being used.

Bit 2 **ERR** *Error Detected.* This bit indicates that an error condition has been detected.

Bit 1 **IBF** *Input Buffer Full.* This bit indicates whether the Input Buffer is full, or whether it is available for taking the data.

Bit 0 **OBF** *Output Buffer Full.* This bit indicates whether the Output Buffer is full, or whether it is available for entering the data or commands.

18.5 KEYBOARD/MOUSE CONTROLLER SUPPORTED COMMANDS

The following table outlines the Keyboard / Mouse Controller Commands supported by the STPC.

Table 18-1. Supported KBM Controller Commands

Command	Command Description
20h	Read the current keyboard command byte. A 20h written to port 64h is followed by a read of port 60h.
60h	Write a new command byte to the keyboard controller. A command of 60h at port 64h is followed by a write to port 60h.
A7h -A8h	Disable mouse port and Enable Mouse port respectively. A disable mouse port command will set bit 5 of the command byte to 0 and pull down the Mouse clock line, preventing any data from being received or sent to the mouse. A enable mouse port command will set bit 5 of command byte to a 1 and the mouse clock line becomes active.
ADh-AEh	Disable Keyboard and Enable Keyboard respectively.A disable keyboard port command will set bit 4 of the command byte to 0 and pull down the keyboard clock line, preventing any data from being received or sent to the keyboard. A enable keyboard port command will set bit 4 of command byte to a 1 and the keyboard clock line becomes active.
AAh	Keyboard Controller self test. In response to this command the controller will return 55h. Internally no tests will be carried out.
C0h	Read Input Port. Reads the controllers input port P1. A C0h command to port 64h followed by a read of 60h will return the contents of the input port. For bit definitions of the input port refer to Section
D0h -D1h	Read Output port and Write Output Port respectively. A read O/p port command followed by a read of port 60h will return the current status of the o/p port. A write o/p port command,D1h, followed by a write to port 60h with the appropriate byte will update the contents of the o/p port. For the bit definitions of the output port refer to section **“How the System Views the Controller”**
D4h	Write to Mouse port. Any command issued to the mouse should be preceded by this command to port 64h then any subsequent byte written to port 60h will be sent to mouse.
DDh-DFh	Disable A20 Address Line and Enable A20 Address Line respectively.
E0h	Read Test inputs
FEh	Generate System Reset. Issues hardware reset by setting the system reset line low for approximately 6 microseconds.

The following table outlines the Keyboard / Mouse Controller Commands sent from the Controller to the Keyboard.

Table 18-2. Support KBM Controller to Keyboard

Command	Description
EDh	LED Write (KBD sends FAh back to the Controller)
EEh	Diagnostic Echo (KBD sends EEh back to the Controller)
F0h	Set/Get Alternate Scan Codes (KBD sends FAh back to the Controller)
F2h	Read Keyboard ID (KBD sends FAh followed by 2-bytes back to the Controller)
F4h	Keyboard Enable (KBD sends FAh back to the Controller)
F7h	Set all keys to Typematic (KBD sends FAh back to the Controller)
F8h	Set all keys to Make/Release (KBD sends FAh back to the Controller)
F9h	Set all keys to Make (KBD sends FAh back to the Controller)
FAh	Set all keys to Typematic/Make/Release (KBD sends FAh back to the Controller)
FBh	Set a key to Typematic (KBD sends FAh, Controller sends a key to the KBD)
FCh	Set a key to Make/Release (KBD sends FAh,Controller sends a key to the KBD)
FDh	Set a key to Make only (KBD sends FAh,Controller sends a key to the KBD)
FEh	Resend (KBD sends FAh back to the Controller)

18.6 UPDATE HISTORY FOR KEYBOARD / MOUSE CONTROLLER CHAPTER

The following changes have been made to the Keyboard / Mouse Controller Chapter on 23/09/99.

Section	Change	Text
18.5	Added	“The following table outlines the Keyboard / Mouse Controller Commands sent from the Controller to the Keyboard.”
18.5	Added	Table 18-2.

Update History for Keyboard / mouse controller chapter

19 SERIAL PORT

19.1 INTRODUCTION

The Serial Port of the STPC is a universal asynchronous receiver/transmitter (UART) which is fully programmable by an 8-bit CPU interface. It supports word lengths from five to eight bits, an optional parity bit and one or two stop bits. If enabled the parity can be odd, even or forced to a defined state. A 16-bit programmable baud rate generator and an 8-bit scratch register are included. Eight modem control lines and a diagnostic loop-back mode are provided. Two 16-Byte FIFOs are included, one for transmit and one for receive. Two DMA handshake lines are provided to indicate when the FIFOs are ready to transfer data to the CPU. An interrupt can be generated from any one of 10 sources.

Up to two Serial Ports are available depending on the Strap Options set (see Section 7). The following bit descriptions relate to one Serial Port, the same bits are also applicable to both Serial Ports, only the register address is different.

19.2 FUNCTIONAL DESCRIPTION

19.2.1 Transmit Operation

Transmission is initiated by writing the data to be sent to the Transmitter Holding Register. The data will then be transfered to the Transmit Shift Register together with a start bit and parity and stop bits as determined by the Line Control Register. The bits to be transmitted are then clocked out of the transmit shift register by the transmit clock (BAUD#) which comes from the baud rate generator.
If enabled, an interrupt will be generated when the Transmitter Holding Register becomes empty.

19.2.2 Receive Operation

Data is clocked into the receiver by the receive clock (RCLK). The receive clock should be 16 times the baud rate of the received data. A filter is used to remove spurious inputs which last for less than two periods of RCLK. When the complete word has been clocked into the receiver the data bits are transfered to the Receiver Buffer Register to be read by the CPU. The receiver also checks for a stop bit and for correct parity as determined by the Line Control Register.

If enabled, an interrupt will be generated when the data has been transfered to the Receiver Buffer Register. Interrupts can also be generated for incorrect parity or a missing stop bit (frame error).

19.2.3 Modem Control Lines

The output modem control lines; RTS# and DTR#, can be set or cleared by writing to the Modem Control Register. The current status of the input modem control line; DCD#, RI#, DSR# and CTS# can be read from the Modem Status Register. Bit 2 of this register will be set if the RI# modem status line has changed from low to high since the register was last read.
If enabled, an interrupt will be generated when DSR#, CTS#, RI# or CD# are asserted.

19.3 SERIAL INTERFACE SIGNALS

SIN1, SIN2, Input Serial input, data is clocked in using RCLK/16.,

SOUT1, SOUT2 Output Serial output, data is clocked out using TCLK/16 (TCLK=BAUD#).

DCD1#, DCD2#, Input Data carrier detect, Active low.

RI1#, RI2#, Input Ring indicator, Active low.

DSR1#, DSR2#, Input Data set ready, Active low.

CTS1#, CTS2#, Input Clear to send, Active low.

RTS1#, RTS2#, Output Request to send, Active low.

DTR1#, DTR2#, Output Data terminal ready, Active low.

BAUD# is an internal output transmit timing clock, derived from CLK divided by the value in the divisor latch DLL & DLM.

19.4 REGISTER DESCRIPTION

19.4.1 Addressing

A0-2 and DLAB (LCR bit 7) define which register appears on DA0 - 7. When CE# and WR# are true, A0-2 and DLAB define which register is to be written with data

.

Address	DLAB	Register Name	Comment
000	0	RBR Receiver buffer	Read Only
000	0	THR Transmitter Holding	Write Only
001	0	IER Interrupt Enable	
010	X	IIR Interrupt Ident	Read Only

010	X	FCR FIFO Control	Write Only
011	X	LCR Line Control	
100	X	MCR Modem Control	
101	X	LSR Line Status	Read Only
110	X	MSR Modem Status	Read Only
111	X	SCR Scratch	
000	1	DLL Divisor Latch (LS)	
001	1	DLM Divisor Latch (MS)	

Note: X = don't care, either 0 or 1.

The first Serial Port is addressed as COM1 at IO address 3F8h, the second Serial Port is addressed as COM2 at IO address 2F8h.

19.4.2 RBR - Receiver Buffer Register (Address 0 DLAB = '0' read only)

This register is updated from the receive shift register at the end of a receive sequence.

If the FIFOs are disabled this register is undefined after reset.
If the FIFOs are enabled this register will return zero after a reset if the receive FIFO is empty.

19.4.3 THR - Transmitter Holding Register (Address 0 DLAB = '0' write only)

Data is held in this register until transferred to the transmitter shift register.

19.4.4 IER - Interrupt Enable Register (Address 1 DLAB = '0')

Bits 4-7 *Reserved.* Read as '0'.

Bit 3 **EDSSI, Enable Modem status interrupt.** When set ("1"), an interrupt is generated if D0, D1, D2 or D3 of the Modem Status Register become set.

Bit 2 **ELSI, Enable Rx Status Interrupt** When set ("1"), an interrupt is generated if D1, D2, D3 or D4 of the Line Status Register become set.

Bit 1 **ETBEI, Enable Tx Holding Register Empty Interrupt** When set ("1"), an interrupt is generated if THRE=1 or the Transmitting Holding Register is empty.

Bit 0 **ERBFI, Enable Receiver Buffer Register** When set ("1"), an interrupt is generated if the Receive Buffer contains data.

After reset Bits 7-0 = '0'

19.4.5 IIR - Interrupt Identification Register (Address 2 read only)

Bits 6-7 **FIFOE,** Returns '1' if FIFOs enabled, otherwise '0'

Bits 4-5 *Reserved.* Always returns 0

Bit 3 **ID2, Interrupt ID Bit 2,** if FIFOs disabled returns 0

Bit 2 **ID1, Interrupt ID Bit 1**

Bit 1 **ID0, Interrupt ID Bit 0**

Bit 0 **INT#, Not interrupt pending**

ID2	ID1	ID0	INT#	Priority	Comment
0	0	0	1		No interrupt pending
0	1	1	0	1	Receiver Line Status
0	1	0	0	2	Receive Data Available or RX FIFO trigger
1	1	0	0	2	Character Timeout Indication
0	0	1	0	3	Transmitter Holding Register Empty
0	0	0	0	4	Modem Status

I

Pending Interrupts are cleared by the following actions:

Priority 1) Reading line status register

Priority 2) Reading receive buffer register

Priority 3) Reading this register if priority 3 interrupt OR writing to the transmitter holding register

Priority 4) Reading the MODEM status register

When multiple interrupts are pending the interrupt line pulses low after each service.

After reset D0 = "1", D1 - D7 = "0".

19.4.6 Receive Timeout Interrupt

A RX FIFO character timeout can be identified when ID2 is '1'.

A RX FIFO character timeout occurs if all the following apply:

1. There is at least one character in the FIFO.
2. The most recent character was received longer than four character periods ago (inclusive of all start, parity, and stop bits).
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The character timeout is dependent on the RX clock.

A timeout interrupt is cleared by a CPU read from the RX FIFO.

The timeout timer is restarted on receipt of a new Byte from the input shift register, or on a CPU read from the RX FIFO.

19.4.7 TX FIFO Interrupt

The Transmitter Holding Register interrupt occurs when the TX FIFO is empty. It is cleared by writing to the Transmitter Holding Register, or by reading from IIR.

The TX FIFO empty interrupt will be delayed one character period minus the last stop bit period whenever; THRE = 1 and there have not been at least two Bytes in the TX FIFO at the same time since the last time THRE = 1. If the TX interrupt is enabled, setting bit 0 of the FCR will generate an immediate interrupt.

19.4.8 FIFO Polled Operation

If the FIFOs are enabled and at least one of the active bits in IER is disabled, then the Serial Port will operate in the FIFO polled mode. Since the TX and RX paths are controlled seperately either one or both can be in the polled mode. The application software should check TX and RX status using the LSR.

19.4.9 FCR - FIFO Control Register (Address 2 write only)

Bit 7-6 **RFTL1-0, RX FIFO trigger level bit 1**.

RFTL1	RFTL0	RX FIFO Trigger Level (Bytes)
0	0	01
0	1	04
1	0	08
1	1	14

Bits 4-5 *Reserved.*

Bit 3 **DMA1, Set DMA mode 1.** This bit determines the DMA mode which the TXRDY and RXRDY pins support. On reset, or when this bit is cleared, the device operates in DMA mode 0. When this bit is set the device operates in DMA mode 1. This bit has no effect unless the FIFOE bit is set as well.

TXRDY - Mode 0: Goes active (low) when TX FIFO, or TX holding register, is empty. Becomes inactive when a Byte is written to the TX channel.

TXRDY - Mode 1: Goes active (low) when there is at least one unfilled position in the FIFO, becomes inactive when the FIFO is full.

RXRDY - Mode 0: Becomes active (low) when there is at least one character in the RX FIFO or the holding register is full. It becomes inactive when there are no more characters in the FIFO or holding register.

RXRDY - Mode 1: Becomes active (low) when the RX FIFO trigger level or timeout occurs, goes inactive when the RX FIFO is empty.

Bit 2 **CLRT, Clear TX FIFO.** Writing a 1 to this bit clears all Bytes in the TX FIFO and resets its counter logic. The output shift register is not affected. This bit is self-clearing.

Bit 1 **CLRR, Clear RX FIFO.** Writing a 1 to this bit clears all Bytes in the RX FIFO and resets its counter logic. The input shift register is not affected. This bit is self-clearing.

Bit 0 **FIFOE, Enable FIFOs.** Writing a 1 to this bit enables both the RX and TX FIFOs. When the FIFOs are either enabled or disabled, both the RX and the TX FIFOs are reset. This bit must be a 1 for any of the other bits in the register to have any effect.

19.4.10 LCR - Line Control Register (Address 3)

Bit 7 **DLAB, Divisor Latch Access Bit.** When clear ‘0’, Receive and Transmitter Registers are read/written address 0 and IER register at address 1. When set ‘1’, Divisor Latch LS is read/written at address 0 and Divisor Latch MS read/written at address 1.

Bit 6 **SB, Set Break.** When set ‘1’, SOUT signal is forced into the ‘0’ state.

Bit 5 **SP, Stick Parity.** When set ‘1’, Parity bit is forced into a defined state, dependent upon state of EPS, PEN:

If EPS = ‘1’ & PEN = ‘1’
Parity bit is set and checked = ‘0’.
If EPS = ‘0’ & PEN = ‘1’
Parity bit is set and checked = ‘1’.

Bit 4 **EPS, Even Parity Select.** When set ‘1’ and PEN = ”1” an even number of ones is sent and checked. When clear ”0” and PEN = ”1” an odd number of ones is sent and checked.

Bit 3 **PEN, Parity Enabled.** When set ”1” parity is transmitted and checked. Parity bit is added after the data field and before the STOP bits. When clear ”0” parity is neither transmitted or checked.

Bit 2 **STB, Number of Stop bits.** When set ”1” two STOP bits are added after each character is sent, except if character length is 5 then 11/2STOP bits are added. When clear ”0” one STOP bit is always added. Only the transmit STOP bits are programmable, the receive stage only expects one STOP bit irrespective of the state of STB.

Bits 1-0 **WLS1-0, Word Length Select.** Transmitted and Received character size is defined as follows:

WLS1	WLS0	Character Size
0	0	5 bits
0	1	6 bit
1	0	7 bit
1	1	8 bit

After reset Bits 7-0 = ‘0’

19.4.11 MCR - Modem Control Register (Address 4)

Bits 7-5 *Reserved.* Read as ’0’

Bit 4 **Loop, Loop back mode.** When set ‘1’ the following conditions are implemented:

i) SOUT is forced to ‘1’.

ii) SIN is disconnected from the Receive input shift register.

iii) Receive shift register input is connected to Transmitter shift register output.

iv) The Modem status signals are disconnected (CTS#, DSR#, DCD#, RI#).

v) The Modem control signals are connected to modem status inputs (RTS to CTS and DTR to DSR).

When clear ‘0’, Modem and control/status signals SIN/SOUT are as normal.

Bits 3-2 *Reserved.*

Bit 1 **RTS, Control Signal.** This signal controls the state of the RTS# output even in loop mode.

When RTS = ‘0’ RTS# = ‘1’.
When RTS = ‘1’ RTS# = ‘0’.

Bit 0 **DTR, Control Signal.** This signal controls the state of the DTR# output even in loop mode.

When DTR = ‘0’ DTR# = ‘1’.
When DTR = ‘1’ DTR# = ‘0’

After reset Bits 7-0 = ‘0’

19.4.12 LSR - Line Status Register (Address 5 read only)

Bit 7 **FIFOERR, RX Data Error in FIFO.** This bit is set to ‘1’ when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register, if there are no subsequent errors in the FIFO.

Bit 6 **TEMT, Transmitter Empty.** If the FIFOs are disabled, this bit is set to ‘1’ whenever the transmitter holding register and the transmitter shift register are empty.
If the FIFOs are enabled, this bit is set whenever the TX FIFO and the transmitter shift register are empty.
In both cases this bit is cleared when a Byte is written to the TX data channel.

Bit 5 **THRE, Transmitter Holding Register Empty.** If the FIFOs are disabled, this bit is set to ‘1’ whenever the transmitter holding register is empty and ready to accept new data, this bit is cleared when the data is transferred to the transmitter shift register.

If the FIFOs are enabled, this bit is set to ‘1’ whenever the TX FIFO is empty. It is cleared when at least one Byte is written to the TX FIFO.

Bit 4 **BI, Break Interrupt.** If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than a transmission time (START bit + DATA bits + PARITY + STOP bits). BI is reset by the CPU reading this register.
If the FIFOs are enabled, this error is associated with the coresponding character in the FIFO. The error is flagged when this Byte is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO, the next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.

Bit 3 **FE, Framing Error.** If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit, FE is reset by the CPU reading this register.
If the FIFOs are enabled, the state of this bit is revealed when the Byte it refers to is at the top of the FIFO.

Bit 2 **PE, Parity Error.** If the FIFOs are disabled, this bit is set if the received data does not have a valid parity bit, PE is reset by the CPU reading this register.
If the FIFOS are enabled, the state of this bit is revealed when the Byte it refers to is at the top of the FIFO.

Bit 1 **OE, Overrun Error.** If the FIFOs are disabled, this bit is set if the receive buffer was not read by the CPU before new data from the receive shift register over wrote previous contents. OE is cleared when the CPU reads this register.
If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX shift register becomes full. OE is set as soon as this happens. The character in the shift register is then overwritten, but is not transferred to the FIFO.

Bit 0 **DR, Data Ready.** This bit is set whenever the receive buffer is full., or by a Byte being transferred into the FIFO. DR is cleared by the CPU reading the receive buffer, or by reading all of the FIFO Bytes.

This bit is also cleared whenever the FIFO enable bit is changed.

After reset bit 7 = ‘0’, bits 6-5= ‘1’, bits 4-0 = ‘0’.

19.4.13 MSR - Modem Status Register (Address 6) Read Only

Bit 7 **DCD, Data Carry Detect**. When Loop = ‘0’ this is the complement of input signal DCD#.
When Loop = ‘1’ this is equal to OUT2.

Bit 6 **RI, Ring Indicator.** When Loop = ‘0’ this is the complement of input signal RI#
When Loop = ‘1’ this is equal to OUT1.

Bit 5 **DSR, Data Set Ready.**

When Loop = ‘0’ this is the complement of input signal DSR#.

When Loop = ‘1’ this is equal to DTR.

Bit 4 **CTS, Clear To Send.** When Loop = ‘0’ this is the complement of input signal CTS#.
When Loop = ‘1’ this is equal to RTS.

Bit 3 **DDCD, Delta Data Carry Detect.** This bit is set (‘1’) if the state of DSR has changed since this register was last read.

Bit 2 **TERI, Trailing Edge Ring Indicator.** This bit is set if the RI# input changes from ‘0’ to ‘1’ since this register was last read.

Bit 1 **DDSR, Delta Data Set Ready.** This bit is set (‘1’) if the state of DSR has changed since this register was last read.

Bit 0 **DCTS, Delta Clear to Send.** This bit is set (‘1’) if the state of CTS has changed since this register was last read.

After reset:

Bits 7-4 are inputs.
Bits 3-0 = ‘0’ and can be written to.

A modem status interrupt can be cleared by writing ’0’ or set by writing ’1’ to this register.

19.4.14 SCR - Scratch Register (Address 7)

This is a general purpose read/write register.

After reset this register is undefined.

19.4.15 DLL & DLM - Divisor Latch

The incoming clock is divided by the value held in DLL & DLM (1-65535) to produce the BAUD# signal.
Note: DLL & DLM can only be updated if DLAB bit is set '1'.

The table below shows required decimal divisor to generate a given baud rate from CLK input of 1.8432, 3.072, or 8.00 MHz:

BAUD# generated is 16 x the required baud rate.

BAUD	8.00MHz
110	9090
300	3333
1200	833
2400	417
4800	208
9600	104
19200	52
38400	26
57600	17

Note: ** No suitable divisor.

If DLL & DLM = 1 then BAUD# is same as CLK
If DLL & DLM = 2 then BAUD# is CLK/2 (square wave)
If DLL & DLM = 3 - 65535 (N) then BAUD# is low for two CLK cycles and high for N-2 CLK cycles.
If DLL & DLM = 0 then BAUD# is low for two CLK cycles and high for 65534 CLK cycles.

After reset DLL & DLM are undefined.

DLL & DLM are not affected by reset.

Writing either DLL or DLM causes count to be loaded immediately and the divisor count to be reset.

19.5 6. SPECIAL FEATURES

19.5.1 Transmit Machine Timing

The TXM (Transmit Machine) starts after 2-3 baud clocks from the time the Transmitter Holding Register is written. The SOUT goes low 7-8 baud clocks from the Transmitter Holding Register being written.

19.5.2 THR Empty Interrupt Timing

A Transmitter Holding Register Empty interrupt will be generated 17-18 clocks after data has been written to the Transmitter Holding Register, providing that the Transmit Machine was idle when the data was written.
If the Transmitter Holding Register is empty when the Transmitter Holding Register Empty interrupt is enabled an interrupt will be generated immediately.

19.5.3 FIFO Reset Timing

When using bits 0-3 of the FIFO Control Register to reset the FIFOs the following timing restrictions apply:

FCR0 - Both FIFOs are reset by the master reset (MR), and are held reset unless FCR0 is set to 1.

FCR1 - The RXFIFO clear requires at least one RCLK period to complete the reset and clear itself.

FCR2 - When set to 1, the TXFIFO clear holds the transmit FIFO reset until the leading edge of the next write strobe to the transmit FIFO, or the next read strobe to the IIR.

20 PARALLEL PORT

20.1 INTRODUCTION

The Parallel Port is a multi-function parallel port that transfers information between the host and a peripheral device (eg. a printer). The parallel port interface contains nine control/status lines and an 8-bit data bus.

The standard PC/AT compatible logical address assignments for LPT1, LPT2 and LPT3 are supported.

The parallel port can be configured for any of the following three modes and supports the IEEE Standard 1284 parallel interface protocol as follows:

– Compatibility Mode (Forward channel, industry-standard parallel port interface ISA)

– Nibble Mode (Reverse channel, compatible with all existing PC hosts)

– Byte Mode (Reverse channel, compatible with IBM PS/2 hosts)

20.2 FUNCTIONAL DESCRIPTION

20.2.1 Communication Modes

The interface is initialised in Compatibility Mode. The other modes provide additional features and/or improved performance. Compliant devices can determine and switch to the most effective mode supported by both devices.

The different communication modes are outlined below.

Note: This document covers the operations and functions of the parallel port hardware. The reader should refer to the IEEE standard 1284 for detailed descriptions of the Compatibility and Nibble, Byte protocols. It should also be noted that the hardware operation for Compatibility and Nibble Mode is identical.

20.2.2 Compatibility Mode

Compatibility Mode provides an asynchronous, byte wide, forward channel (host-to-peripheral), with the data and status lines used according to original definitions, as per the original Centronics port.

20.2.3 Nibble Mode

Nibble Mode provides an asynchronous, reverse channel (peripheral-to-host) under the control of the host. Data bytes are transmitted as two sequential, four-bit nibbles using four peripheral-to-host status lines. When the host and/or peripheral do not support bi-directional use of the data lines, Nibble Mode may be used with Compatibility Mode to implement a bi-directional channel. The two modes cannot be active simultaneously.

20.2.4 PS/2 or Byte Mode

Byte Mode provides an asynchronous, byte-wide, reverse channel (peripheral-to-host) using the eight data lines of the interface for data and the control/status lines for handshaking. Byte Mode may be used to implement a bi-directional channel, with the transfer direction controlled by the host, when both host and peripheral support bi-directional use of the data lines.

20.2.4.1 Matrix of Protocol Signal Names

Table 0-1 below shows the signal names for each protocol as a matrix.

Table 0-1. Parallel Port Protocol Signal Names.

Parallel Port Signal Names	Compatible Signal Names	Nibble Signal Names	Byte Signal Names
PD[7-0]	PD[7-0]		PD[7-0]
SLCT	Select	Xflag	Xflag
ACK#	Ack#	PtrClk	PtrClk
BUSY	Busy	PtrBusy	PtrBusy
PE	pe	AckDataReq	AckDataReq
ERR#	Fault#	DataAvail#	DataAvail#
SLCTIN#	Selectin#	1284 Active	1284 Active
INIT#	INIT#		
STROBE#	Strobe#	HostClk	HostClk
AUTOFD#	AutoFd#	HostBusy	HostBusy

20.2.5 Parallel Port Configuration

The Parallel Port is configured by a set of three programmable registers accessed through a Configuration Select Register (CSR). These registers are in the default state after power-up and are unaffected by RESET.

20.2.5.1 Configuration Procedure

The following sequence in Table 0-2 is required to program the configuration registers

:

Table 0-2. Configuration Register Programming Procedure

Step	Process	Method
1	Enter Configuration.	This requires 55h to be written to port 3F0h (CSR) twice in Mode succession. Note: It is recommended that interrupts be disabled for the duration of the two writes. If a write to another address or port occurs between the two writes, the Parallel Port will not enter Configuration Mode.
2	Configure Registers.	The Parallel Port contains three configuration registers CR1, CR4 and CRA. These registers are accessed by first writing the number of the desired register to port 3F0h (CSR), then writing or reading the selected register through port 3F1h.
3	Exit Configuration Mode.	Configuration Mode is exited by writing AAh to port 3F0h (CSR).

20.2.5.2 Configuration Select Register (CSR)

This register can only be accessed when the Parallel Port is in Configuration Mode. The CSR is located at port 3F0h and must be initialised upon entering Configuration Mode before the three configuration registers can be accessed, after which it can be used to select which of the configuration registers is to be accessed at port 3F1h.

20.2.5.3 Configuration Register (CR1)

This register can only be accessed when the Parallel Port is in the Configuration Mode and after CSR has been initialised to 01h.

Bits 7-4 Reserved.

Bit 3 Parallel Port Mode.

If ’1’, sets the Parallel Port for Printer Mode (Default). If ’0’, enables the Extended Parallel Port mode. (See CR4)

Bit 2 Reserved.

Bits 1-0 Parallel Port Address.

These bits are used to select the Parallel Port Address.

Bit 1	Bit 0	Description
0	0	Disabled
0	1	3BCh
1	0	378h
1	1	278h (Default)

The default value of this register after power-up is 9Fh.

20.2.5.4 Configuration Register (CR4)

This register can only be accessed when the Parallel Port is in Configuration Mode and after CSR has been initialised to 04h.

Bit 7 Reserved. Set/read as ’0’

Bits 6-2 Reserved.

Bits 0,1 Parallel Port Extended Modes

Bit 1	Bit 0	If CR1 (3) = 0 then:
0	0	Standard and Bi-directional Modes(SPP) (Default)

The default value of this register after power-up is 00h.

20.2.6 Parallel Port Registers

This section describes Compatibility and Byte Modes.

Each register set description contains the I/O address assignments and a description of the registers and register bits. STAT and CTRL registers are common to all modes.

The base address for the parallel port is determined at power-up. This can be changed by software as described in Section 20.2.5. All registers are accessed as byte quantities.

Some of the registers described contain reserved bits. They have an associated value defined in the register description which is a hard value: it will not change even if these bits are written to. A read from a register that contains reserved bits will return the hard values associated with those bits.

Data is latched into all of the registers on the rising edge of the internal IOW# signal.

20.2.6.1 Compatibility and Byte Modes

The port consists of three registers and can be programmed to operate at three different base addresses: 278h, 378h and 3BCh.

The write locations are :

(I) Write data to output port (DATA)
- Base Address + 0h

(ii) Write command to output port (CTRL)
- Base Address + 2h

The read locations are :

(I) Read peripheral data (DATA)
- Base Address + 0h

(ii) Read peripheral status data (STAT)
- Base Address + 1h

(iii) Read back control register (CTRL) - Base Address + 2h

Reads should not be performed from the DATA register in Compatibility mode.

20.2.6.2 Status Register (P_STAT)
Base Address + 01h.

Bit 7 **BUSY.** If asserted, indicates that the peripheral is busy.

Bit 6 **ACK#.** If asserted, indicates that the peripheral has received a data byte and is ready for another.

Bit 5 **PE.** If asserted, indicates that the peripheral is out of paper.

Bit 4 **SLCT.** If asserted, indicates that the peripheral is selected.

Bit 3 **ERR#.** If asserted, indicates that the peripheral has a fault.

Bit 2 **PINTR1.** This indicates a CPU interrupt by the parallel port, ie. the printer has accepted the previous character and is ready for another.

Bit 1 *Reserved.*

Bit 0 *Reserved*

The content of the STAT register is stored on the falling edge of the internal IOR# signal.

The ’STAT’ register is read only, writes to it have no effect.

20.2.6.3 Control Register (P_CTRL)
Base Address + 02h.

Bits 7-6 *Reserved.* Always returns ’0’.

Bit 5 **PDIR. Direction bit.** In PS/2 (Byte) Mode, this bit is used to control the direction of the data transfer on the parallel port data bus.
Also in Byte Mode, when PDIR = 0 (forward direction), PDOUT[7:0] is enabled; when PDIR = 1 (reverse direction), PDIN[7:0] is enabled.

Bit 4 **INTEN.** If asserted, allows the peripheral to interrupt the CPU.

Bit 3 **SLCTIN#.** If asserted, it means the host has selected the peripheral.

Bit 2 **INIT#.** If asserted, the peripheral is initialised.

Bit 1 **AUTOFD#.** This tells the printer to advance the paper by one line each time a carriage return is received.

Bit 0 **STROBE#.** If asserted, this instructs the peripheral to accept the data on the data bus.

At reset, CTRL is set to 00h.

21. POWER MANAGEMENT

21.1 INTRODUCTION

For full information on the action of the SMM, please refer to the STMicroelectronics manual titled “ST486DX SMM programming manual”. This chapter presents the control registers for SMM of the STPC.

The STPC provides the following hardware structures to assist the software in managing the power consumption by the system.

– System Activity detection

– Three power down timers:

– Doze timer for detecting lack system activity for short durations

– Standby timer for detecting lack of system activity for medium durations

– Suspend timer for detecting lack of system activity for long durations.

– House-keeping activity detection

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

– Peripheral Activity detection

– Peripheral timer for detecting lack of peripheral activity

– STPCLK# modulation to adjust the system performance in various power down states of the system including full power on state

Lack of system activity for progressively longer period of times is detected by the three power down timers. These timers can generate SMI interrupts to CPU so that the SMM software can put the system in decreasing states of power consumption. System activity in a power down state can generate an SMI interrupt to allow the software to bring the system back up to the full power on state. The chip-set supports up to 3 power down states: Doze state, Standby state and Suspend state. These correspond to increasing levels of power savings.

The chip-set can detect presence/absence of the following System activities.

– DMA Request (DRQ) activity

– Interrupt Request (INTR) activity

– Parallel IO (PIO) activity

– Serial IO (SIO) activity

– Keyboard (KBD) activity

– Floppy Disk Controller (FDC) activity

– Hard Disk Controller (HDC) activity

– PCI master device activity

– A programmable address range

Each of these can be individually enabled. The presence of an enabled system activity resets the power down timers. The chip-set generates the SMI interrupt when no system activity is detected for the delay period programmed in the power-down timers. The software can then put appropriate sub-systems in power down mode, request STPCLK# assertion and stop CPU and other system clocks, program the current power-down state in the chip set and set up the next timer.

Presence of an enabled system activity, when the STPC is in a power down state will first enable any stopped clocks, wait for a a programmable delay to allow any internal PLLs to stabilize and then deassert STPCLK# to enable CPU execution. The device can optionally generate SMI interrupt to allow the SMM to bring the system back to power-on state.

The current revision of the STPC does not implement support for stopping CPU and other system clocks.

In Doze or Standby state, a house-keeping activity can bring the system back to full speed for a short period of time before returning back to Doze or Standby state. The chip-set can detect following house-keeping activities.

– DMA Request (DRQ) activity
– Interrupt Request (INTR) activity
– Keyboard (KBD) activity
– PCI master device activity

The house-keeping timer determines the length of time the system will be on before returning to the original power-down state. An activity can be either a system activity or a house-keeping activity but not both at the same time. Further, the Suspend state can not make use of this feature.

The absence of the following peripheral activities can be enabled to cause a SMI interrupt and thus allowing the software to put the unused peripherals in power down state while the remainder of the system is still in full power on state.

– Parallel IO (PIO) activity
– Serial IO (SIO) activity
– Keyboard (KBD) activity
– Floppy Disk Controller (FDC) activity
– Hard Disk Controller (HDC) activity
– A programmable address range

Each of these can be individually enabled for inactivity detection. The presence of a peripheral activity does not reset the peripheral timer. It always times out after the programmed delay period. An SMI interrupt is generated if any of the enabled peripheral was not active for this time period. The device provides IO access trapping to detect access to a powered-down peripheral so that the software can bring the peripheral to power on state before the access is completed.

The STPC can also do software transparent power management if so enabled. In this mode of operation, doze and standby time-outs will change the CPU clock without generating an SMI interrupt. The state transitions from fully-on to doze or standby and back to fully-on will take place automatically. Also note that the suspend state can never be entered automatically and always requires software assist.

The STPC decodes the following to detect activities of various kind, see Table 21-1:

Table 21-1. Activity detection

Activity	Detected via
ISA DMA masters	Low to high transition of hold request of 206
PCI masters	High to low transition of any of PCIRQ2-0#
Parallel port	IO read/write at 378h-37Fh, 278h-27Fh and 3BCh-3BFh
Serial port	IO read/write at 3F8h-3FFh, 2F8h-2FFh, 3E8h-3EFh and 2E8h-2EFh
Keyboard	IO read/write at 60h, 62h, 64h and 66h
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.

21.2 POWER MANAGEMENT REGISTERS

21.2.1 TIMER REGISTER 0 INDEX 60H (TIMER0)

This register controls the timer selection for the length of timeout for doze, standby, and suspend modes.

Bits 7-5 **Suspend Timeout Timer**, when set to any value other than the disable value (000), this timer will generate SMI interrupt on time out.

Once enabled this timer counts down from the programmed valued. If any of the enabled system activities are detected before time out, the timer will reset and start again. These bits are encoded as follows in Table 21-2:

Table 21-2. Suspend timer encoding

7	6	5	Suspend Timer reset
0	0	0	disabled
0	0	1	4 minutes
0	1	0	8 minutes
0	1	1	12 minutes
1	0	0	16 minutes
1	0	1	32 minutes
1	1	0	48 minutes
1	1	1	64 minutes

The suspend timer will count whenever it is not disabled and the suspend time-out bit in the SMI status register 0 is not set to a 1.

Bits 4-2 **Standby Timeout Timer,** when set to any value other than the disable value (000) this timer, on expiration, can either generate the SMI interrupt to the CPU or if programmed for auto-power saving (software transparent power management) mode, change the power-down state to Standby state (refer to auto-power saving mode for details of the power saving features are enabled with standby state). Similar to the Suspend timer, presence of an enabled system activity will reset the timer to start counting again. These bits are encoded as follows in Table 21-3:

Table 21-3. Standby timer encoding

4	3	2	Standby Timer reset
0	0	0	disabled
0	0	1	Reserved
0	1	0	2 minutes
0	1	1	4 minutes
1	0	0	6 minutes
1	0	1	8 minutes
1	1	0	12 minutes
1	1	1	16 minutes

The standby timer will count whenever it is not disabled and the standby time-out bit in the SMI status register 0 is not set to a 1.

Bits 1-0 *Reserved.*

21.2.2 TIMER REGISTER 1 INDEX 61H (TIMER1)

Bit 7 *Reserved.*

Bits 6-4 **House-keeping Timer.** This timer determines how long the PMU will be in Doze house-keeping state when an enabled house-keeping activity is detected while in doze or standby power-down states. It is encoded as follows in Table 21-4:

Table 21-4. Housekeeping timer encoding

6	5	4	House-keeping Timer reset
0	0	0	disabled
0	0	1	64 micro-seconds
0	1	0	128 micro-seconds
0	1	1	256 micro-seconds
1	0	0	Reserved
1	0	1	4 milli-seconds
1	1	0	16 milli-seconds
1	1	1	32 milli-seconds

The house-keeping counts only when the PMU is in one of the house-keeping states. Another house-keeping activity while the controller is in house_keeping state will reset the house-keeping timer to start counting again.

A system activity detection in the house_keeping state will have the same effect as if the controller was in Doze or Standby state. Either a SMI interrupt will be generated to allow the software to bring the system to power-on state or the controller will automatically transition to power-on state. The house-keeping timer and function can be disabled by masking out all activity detection via House-keeping Enable registers.

Bits 3-1 **Peripheral Timeout Timer.** When set to a value other than (000) this timer on expiration, will generate SMI if any of the enabled peripherals remained inactive during the entire period. Unlike the power-down timers, the peripheral timer does not reset due to an enabled peripheral activity. It always times out after the programmed delay. A SMI interrupt is generated only if any of the enabled peripherals were inactive during this period. This field is encoded as follows in Table 21-5:

Table 21-5. Peripheral timeout encoding

3	2	1	Peripheral Timer reset
0	0	0	disabled
0	0	1	8 seconds
0	1	0	16 seconds
0	1	1	32 seconds
1	0	0	64 seconds
1	0	1	128 seconds
1	1	0	256 seconds
1	1	1	512 seconds

The peripheral timer counts whenever it is enabled.

Bit 0 *Reserved.*

This register defaults to 00h after reset.

21.2.3 TIMER REGISTER 2 INDEX 8DH (TIMER2)

Bits 7-5 **Doze Timeout Timer.** When set to any value other than the disable value (00), this timer, on expiration, can either generate the SMI interrupt to the CPU or if programmed for auto-power saving (software transparent power management) mode, change the power-down state to Doze state (refer to auto-power saving mode for details of the power saving features that are enabled with Doze state). Similar to the suspend timer, presence of an enabled system activity will reset the timer to start counting again. This 3-bit field is encoded as follows in Table 21-6:

Table 21-6. Doze timeout encoding

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	Reserved
1	0	1	4 seconds
1	1	0	8 seconds
1	1	1	16 seconds

The doze timer will count whenever it is not disabled and the doze time-out bit in the SMI status register 0 is not set to a ‘1’.

Bits 4-2 *Reserved.*

This register defaults to 00h at reset.

21.2.4 SYSTEM ACTIVITY ENABLE REGISTER 0 INDEX 62H (SYS_ACTIV_EN0)

This is the first of the three registers that control which system activity to detect.

Bit 7 **DMA Request (DRQ)**

Bit 6 **PCI master device (PCIM)**

Bit 5 **Parallel IO (PIO)**

Bit 4 **Serial IO (SIO)**

Bit 3 **Keyboard (KBD)**

Bit 2 **Floppy Disk Controller (FDC)**

Bit 1 **Hard Disk Controller (HDC)**

Bit 0 *Reserved.*

This register defaults to 00h at reset, which corresponds to ignoring all of the listed system activities.

Programming notes

When detected, the power-down timers will reload with their initial time values or if enabled via SMI control register, a SMI interrupt will be generated or if programmed for auto-power down mode and in Doze or Standby power-down states, transition to power-on state will take place. Set the following bits to ‘1’ to detect the associated activity, and to ‘0’ to ignore the associated activity.

21.2.5 SYSTEM ACTIVITY ENABLE REGISTER 1 INDEX 63H (SYS_ACTIV_EN1)

This is the second of the three registers that control which system activity to detect.

Bits 7-6 *Reserved.* Must be programmed to ‘0’

Bit 5 **Address range 0**

Bits 4-0 *Reserved.* Must be programmed to ‘0’

This register defaults to 00h at reset, which corresponds to ignoring all of the listed system activities.

21.2.6 SYSTEM ACTIVITY ENABLE REGISTER 2 INDEX 64H (SYS_ACTIV_EN2)

This is the third of the three registers that control which system activity to detect.

Bit 7 **IRQ15-1** 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.

This register defaults to 00h disabling detection of all house-keeping activity detection.

21.2.7 HOUSE-KEEPING ACTIVITY ENABLE REG. 0 INDEX 65H (HK_ACTIV_EN0)

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.*

21.2.8 HOUSE-KEEPING ACTIVITY ENABLE REG. 1 INDEX 66H (HK_ACTV_EN1)

This is the second house-keeping activity detection enable register.

Bits 7-6 *Reserved.* Must be programmed to ‘0’

Bit 5 **Address range 0**

Bits 4-0 *Reserved.* Must be programmed to ‘0’

This register defaults to 00h disabling detection of all house-keeping Activity detection.

21.2.9 PERIPHERAL INACTIVITY DETECTION REG. 0 INDEX 67H (PERIF_INACTV0)

This register controls which peripheral inactivity is enabled for generating a SMI interrupt on a peripheral time-out.

Bit 7 **Parallel IO (PIO)** activity

Bit 6 **Serial IO (SIO)** activity

Bit 5 **Keyboard (KBD)** activity

Bit 4 **Floppy Disk Controller (FDC)** activity

Bit 3 **Hard Disk Controller (HDC)** activity

Bit 2 **Address range 0**

Bits 1-0 *Reserved.* Must be programmed to ‘0’

This register defaults to 00h disabling detection of all peripherals.

Programming notes

Lack of peripheral activity for an enabled peripheral for one peripheral time-out period generates SMI interrupt. A ‘1’ in a bit position enables the SMI generation for associated peripheral and a ‘0’ disables it. Software can use Peripheral Inactivity status register to determine which peripheral should be powered down.

21.2.10 PERIPHERAL ACTIVITY DETECTION REG. 0
INDEX 69H (PERIF-ACTIV0)

This register controls which peripheral accesses will cause a SMI.

Bit 7 **Parallel port (PIO)** access

Bit 6 **Serial port (SIO)** access

Bit 5 **Keyboard (KBD)** access

Bit 4 **Floppy Disk Controller (FDC)** access

Bit 3 **Hard Disk Controller (HDC)** access

Bit 2 **Address range 0**

Bits 1-0 *Reserved.* Must be programmed to ‘0’

This register defaults to 00h disabling all SMI generation.

Programming notes

Typically the power management software will detect non-usage of a peripheral device via Peripheral inactivity status registers, bring the peripheral into power down state and then enable trapping access to that peripheral via this register.

Thus when an application attempts to make use of a powered down peripheral, the access is trapped and a SMI interrupt is generated to allow software to re-power the peripheral device before allowing the access to complete. This is register is first of the two such registers.

A ‘1’ in a bit position enables SMI generation for the associate peripheral and a ‘0’ disables.

21.2.11 PERIPHERAL ACTIVITY DETECTION REG. 1
INDEX 6AH (PERIF_ACTIV1)

This is the second register that controls which peripheral accesses will cause a SMI interrupt. This register is similar in functionality to Peripheral Activity detection register 0.

Bits 7-0 *Reserved.* Must be programmed to ‘0’

This register defaults to 00h.

21.2.12 ADDRESS RANGE 0 REGISTER 0
INDEX 6BH (ADDRESS_RANGE0-0)

This register contains bits which are compared with PCI address bits 31-24 if range compare is enabled for memory cycle or compared against bits 15-8 if range compare is enabled for IO cycles.

This register defaults to 00h after reset.

21.2.13 ADDRESS RANGE 0 REGISTER 1
INDEX 6CH (ADDRESS_RANG0-1)

Bits 7-3 These bits are compared with PCI address bits 23-19 if range compare is enabled 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.

21.2.14 SMI CONTROL REGISTER 0 INDEX 71H (SMI_CONT0)

This register controls the generation of SMI interrupt as follows:

Bit 7 If ‘1’ then generate SMI on Doze time-out. Otherwise if set to a ‘0’, the hardware will transition to Doze state automatically on Doze time-out.

Bit 6 If ‘1’ then generate SMI on Standby time-out. Otherwise if set to a ‘0’, the hardware will transition to Standby state automatically on Standby time-out.

Bit 5 If ‘1’ then generate SMI on Suspend time-out. Otherwise if set to a ‘0’, SMI is not generated. The hardware never transitions into Suspend state by itself.

Bit 4 If ‘1’ then generate SMI on House-keeping time-out. Otherwise if set to a ‘0’, the hardware will automatically transition back to the doze or standby state (which ever state it was in before entering house-keeping state).

Bit 3 If ‘1’ then generate SMI on detecting a 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.

21.2.15 SMI STATUS REGISTER 0 INDEX 73H (SMI_STAT0)

This register contains the status information pertaining to the SMI interrupt.

Bit 7 **Doze time-out.** This bit is set to a ‘1’ when Doze time-out occurs. An SMI interrupt will be generated if associated SMI enable bit in SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#. If SMI generation has been disabled then the controller will automatically transition to Doze state. This bit will then be cleared on transition from Doze or Standby to Power-on state.

Bit 6 **Standby time-out.** This bit will be set to a ‘1’ when Standby time-out occurs. A SMI interrupt will be generated if the associated SMI enable bit in SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#. If SMI generation has been disabled then the hardware will automatically transition to Standby state. This bit will then be cleared on transition Standby to Power-on state.

Bit 5 **Suspend time-out.** This bit will be set to a ‘1’ when Suspend time-out occurs. A SMI interrupt will be generated if the associated SMI enable bit in the SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#.

Bit 4 **House-keeping timeout detected.** This bit will be set to a ‘1’ if the controller is in one of the house-keeping states and the house-keeping timer expires. A SMI interrupt will be generated if the associated SMI enable bit in the SMI Control register is set to a ‘1’. The software must write this bit to a ‘1’ to deassert SMI#. If the SMI generation has been disabled, the hardware will automatically transition to doze or standby state. This bit then will be cleared on transition from Doze or Standby states to any other state.

Bit 3 **House-keeping activity detected.** This is a read-only bit and represents the OR of the System activity status registers masked (ANDed) with the corresponding bits in the House-keeping Activity enable registers. A SMI interrupt will be generated when this bit is a 1’ and if the associated SMI enable bit in the SMI Control register is set to a ‘1’. The software can refer to Activity status register to determine the cause of the interrupt. The software must clear the corresponding bits of the Activity Status register to deassert SMI#. If SMI generation has been disabled and if the controller in Doze or Standby state, it will automatically transition to House-keeping state.

Bit 2 **System Activity detected**. This is a read-only bit and represents the OR of the System activity Status registers masked (ANDed) with the corresponding bits of the System Activity enable registers. A SMI interrupt will be generated if this bit is a ‘1’ and if the associated SMI enable bit in SMI Control register is set to a ‘1’. The software can refer to Activity status register to determine the cause of this interrupt. The software must clear the System Activity Status registers bits for the enabled system activities to deassert SMI#. If SMI generation has been disabled and if the controller is in Doze or Standby state, it will automatically transition to Power-on state.

Bit 1 **Peripheral Inactivity detected.** This is a read-only bit and represents the OR of Peripheral Inactivity Status register bits masked (ANDed) with the associated Peripheral inactivity detection register bit. A SMI# interrupt will be generated when this bit is a ‘1’. The software can refer to Peripheral Inactivity status registers to determine which peripheral should be powered down. The software must clear the corresponding bits of the Peripheral Inactivity detection register to deassert SMI#.

Bit 0 **Peripheral activity detected.** This is a read-only bit and represents the OR of the System activity Status register masked (ANDed) with the corresponding bits of the Peripheral Activity detection registers. A SMI interrupt will be generated when this bit is a ‘1’. The software can refer to the System Activity status register to determine which peripheral caused the interrupt. The software must clear the corresponding bits of the System activity register to deassert SMI#.

Programming notes

The SMI# output is a logical OR of all the bits (ANDed with their respective SMI generation enable bits) in this register. SMI# output will be deasserted within 3 PCI clocks after the cause of the SMI# interrupt is cleared.
This register defaults to 00h after reset deasserting SMI# output.

21.2.16 SMI STATUS REGISTER 1 INDEX 74H (SMI_STAT1)

This register is similar to SMI Status register 0 in that it reports the cause of the SMI interrupt to the software.

Bit 7 **Software SMI.** This bit is set to a ‘1’ by write writing a ‘1’ in bit-1 of the SMI Control register. The software must clear this bit to deassert SMI#.

Bits 6-0 *Reserved.*

This register defaults to 00h after reset.

21.2.17 PERIPHERAL INACTIVITY STATUS REGISTER 0 INDEX 75H (PERIF_STATUS0)

This register contains a ‘1’ in a bit position if the associated peripheral was inactive for the entire duration of the last peripheral time-out period.

Bit 7 **Parallel IO (PIO)** activity

Bit 6 **Serial IO (SIO)** activity

Bit 5 **Keyboard (KBD)** activity

Bit 4 **Floppy Disk Controller (FDC)** activity

Bit 3 **Hard Disk Controller (HDC)** activity

Bit 2 **Address range 0**

Bits 1-0 R*eserved.*

This register defaults to 00h after reset.
It can also be cleared by software by writing a ‘1’ in the bit which is set to ‘1’.

Programming notes

A bit in this register is set to a ‘1’ only at peripheral timer time-out. It is set to a ‘0’, as soon as an 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.

21.2.18 ACTIVITY STATUS REGISTER 0 INDEX 77 (ACTV_STATUS0)

This register records presence of activity.

Bit 7 **DMA Request (DRQ)** activity

Bit 6 **PCI master device (PCIM)** activity

Bit 5 **Parallel IO (PIO)** activity

Bit 4 **Serial IO (SIO)** activity

Bit 3 **Keyboard (KBD)** activity

Bit 2 **Floppy Disk Controller (FDC)** activity

Bit 1 **Hard Disk Controller (HDC)** activity

Bit 0 *Reserved.*

This register defaults to 00h after reset.

Programming notes

A ‘1’ in a bit position indicates that presence of the associated activity since the bit was last cleared. Once set, a bit of this register can only be cleared by software writing a ‘1’ to it or by reset or if auto power management is enabled then any transition to Doze or Standby state (including the ones from house-keeping states) will clear all enabled System and House-keeping activities.

The status reflected in this register is not conditioned by the settings of System Activity Enable, House-keeping Activity Enable, Peripheral Inactivity or Peripheral Activity Detection registers.

21.2.19 ACTIVITY STATUS REGISTER 1 INDEX 78H (ACTIV_STAT1)

This register is similar to Activity Status register 0. It contains the status for the following bits.

Bits 7-6 *Reserved.* Must be programmed to ‘0’

Bit 5 **Address range 0**

Bits 4-0 *Reserved.* Must be programmed to ‘0’

This register defaults to 00h after reset.

21.2.20 ACTIVITY STATUS REGISTER 2 INDEX 79H (ACTIV_STAT2)

This register is similar to Activity Status registers 0 and 1.

Bit 7 **IRQ15-1** activity

Bit 6 **IRQ0** activity

Bit 5 **NMI** activity

Bits 4-0 *Reserved.*

This register defaults to 00h after reset.

21.2.21 PMU STATE REGISTER INDEX 7AH (PMU)

This register contains the state the power management controller currently is in.

Bit 7 *Reserved.*

Bit 6 **PMU microsecond clock test mode.** This bit is for factory use only and must be set to a ‘0’ by the software. Setting this bit to a 1’ causes the microsecond clock to tick at oscillator clock frequency instead of every microsecond.

Bit 5 **PMU millisecond clock test mode.** This bit is for factory use only and must be set to a ‘0’ by the software. Setting this bit to a 1’ causes the millisecond clock to tick at oscillator clock frequency instead of every millisecond.

Bit 4 **PMU second clock test mode.** This bit is for factory use only and must be set to a ‘0’ by the software. Setting this bit to a 1’ causes the second clock to tick at oscillator clock frequency instead of every second.

Bit 3 **PMU minute clock test mode.** This bit is for factory use only and must be set to a ‘0’ by the software. Setting this bit to a 1’ causes the minute clock to tick at oscillator clock frequency instead of every minute.

Bits 2-0 **PMU state** see Table 21-7

:

Table 21-7. PMU encoding

2	**1**	**0**	**PMU state**
0	0	0	Power-on
0	0	1	Doze
0	1	0	Standby
0	1	1	Suspend
1	0	1	Doze_house_keeping
1	1	0	Standby_house_keeping
1	1	1	Reserved

This register defaults to 00h after reset.

The architecture allows for either the software to explicitly program the power-down state the controller should be in or the controller can change states automatically (auto-power down mode of operation) or a mix of the two. Some power-down states are entered and exited automatically by the hardware while the others require software assist. This is based on the SMI Control register settings as follows:

Transition from Power-on to Doze state will take place automatically on Doze time-out, if bit-7 of the SMI control register is set to a ‘0’. Otherwise if bit-7 is programmed to be a ‘1’, an SMI will be generated instead and the software can change the state to Doze.

Transition from Doze to Power-on will take place automatically in presence of an enabled system activity if bit-2 of the SMI control register is programmed to be a 0’. Otherwise if bit-2 is programmed to a ‘1’, an SMI interrupt will be generated instead and software can change the state to Power-on.

Transition from Doze to Doze_house_keeping state will take place automatically if an enabled house_keeping activity is detected and bit-4 of the SMI control register is set to a ‘0. Otherwise if bit-7 is programmed to be a ‘1’, an SMI interrupt will be generated instead.

Transition from Doze_house_keeping state to Doze will take place automatically on house-keeping time-out if bit-3 of the SMI control register is set to a ‘0’. Otherwise an SMI interrupt will be generated instead.

Transition from Doze_house_keeping state to Power-on will take place on detecting an enabled system activity automatically if bit-2 of the SMI control register is programmed to ‘0’. Otherwise an SMI interrupt will be generated instead.

Transitions from Doze or Power-on state to Standby will take place automatically on standby time-out if bit-6 of the SMI control register is set to a ‘0’. Otherwise an SMI interrupt will be generated instead.

Transition from Standby to Power-on will take place automatically in presence of an enabled system activity if bit-2 of the SMI control register is programmed to be a ‘0’. Otherwise if bit-2 is programmed to a ‘1’, SMI interrupt will be generated instead and software can change the state to Power-on.

Transition from Standby to Standby_house_keeping state will take place automatically if an enabled house_keeping activity is detected and bit-4 of the SMI control register is set to a ‘0’. Otherwise if bit-7 is programmed to be a ‘1’, a SMI interrupt will be generated instead.

Transition from Standby_house_keeping state to Standby will take place automatically on house-keeping time-out if bit-3 of the SMI control register is set to a ‘0’. Otherwise SMI interrupt will be generated instead.

Transition from Standby_house_keeping state to Power-on will take place on detecting an enabled system activity automatically if bit-2 of the SMI control register is programmed to ‘0’. Otherwise SMI interrupt will be generated instead.

The hardware never transitions to Suspend state automatically.

The power saving features associated with each power-down state are independent of how the state was entered.

21.2.22 GENERAL PURPOSE REGISTER INDEX 7BH (GP)

This is a read/write IO register that can be used by software.

Bit 7 **General Purpose Register Bit 7**

Bit 6 **General Purpose Register Bit 6**

Bit 5 **General Purpose Register Bit 5**

Bit 4 **General Purpose Register Bit 4**

Bit 3 **General Purpose Register Bit 3**

Bit 2 **General Purpose Register Bit 2**

Bit 1 **General Purpose Register Bit 1**

Bit 0 **General Purpose Register Bit 0**

This register defaults to 00h at reset.

Programming notes

Writing to this register also updates the external ‘373 latch that can be used to control external 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.

21.2.23 CLOCK CONTROL REGISTER 0 INDEX 7CH (CLOCK_CONT0)

This register allows control over power saving via stop clock modulation. The power-saving can be tuned to the power-management state the PMU is in.

Bits 7-5 **Power-on and housekeeping states** STPCLK# modulation control. These bits control the duty cycle of STPCLK# deassertion when the PMU is in Power-on or one of the house-keeping states as follows in Table 21-8:

Table 21-8. STPCL# modulation encoding

7	6	5	Ratio	Power-on STPCLK# Modulation
0	0	0	1	STPCLK# is never asserted
0	0	1	1/2	1 half period
0	1	0	1/4	1 quarter period
0	1	1	1/8	one-eighth period
1	0	0	1/16	one-sixteenth period
1	0	1	1/32	1/32 period
1	1	0	1/64	1/64 period
1	1	1		*Reserved.*

The STPCLK# is deasserted and the duty-cycle control ignored if a SMI interrupt is pending.

Bits 4-2 **Doze/Standby/Suspend states** STPCLK# modulation control. These bits control the duty cycle of the STPCLK# deassertion when PMU is in one of the power-down states as follows in Table 21-9:

Table 21-9. D/S/S State encoding

7	6	5	Ratio	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 64ms.

Bit 0 *Reserved.*

This register defaults to 00h.

21.2.24 DOZE TIMER READ BACK REGISTER INDEX 88H (DOZE)

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit doze timer.

Bits 7-0 Bits 8-1 of the current value of the doze timer.

Programming notes

This register should not be used by the software.

Note that bit 0 of the current value of the doze timer is not readable.

21.2.25 STANDBY TIMER READ BACK REGISTER INDEX 89H (STANDBY)

This read only register is provided for test purposes to read back the current value of 5-bit standby timer.

Bits 7-5 *Reserved.*

Bits 4-0 Bits 4-0 of the current value of the standby timer.

Programming notes

This register should not be used by software.

21.2.26 SUSPEND TIMER READ BACK REGISTER INDEX 8AH (SUSPEND)

This read only register is provided for test purposes to read back the current value of the 7-bit Suspend timer.

Bit 7 *Reserved.*

Bits 6-0 Bits 6-0 of the current value of the suspend timer.

Programming notes

This register should not be used by software.

21.2.27 HOUSE-KEEPING TIMER READ BACK REG. INDEX 8BH (HK_TIMER)

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit house-keeping timer.

Bits 7-0 Bits 8-1 of the house-keeping timer.

Programming notes

This register should not be used by software.

21.2.28 PERIPHERAL TIMER READ BACK REGISTER INDEX 8CH (PERIF_TIMER)

This read only register is provided for test purposes to read back the current value of the upper 8-bits of the 9-bit Peripheral timer.

Bits 7-0 Bits 8-1 of the Peripheral timer.

Programming notes

This register should not be used by software.

22. MECHANICAL DATA

22.1 388-PIN PACKAGE DIMENSION

The pin numbering for the STPC 388-pin Plastic BGA package is shown in Figure 22-1.

Dimensions are shown in Figure 22-2, Table 22-1 and Figure 22-3, Table 22-2.

Figure 22-1. 388-Pin PBGA Package - Top View

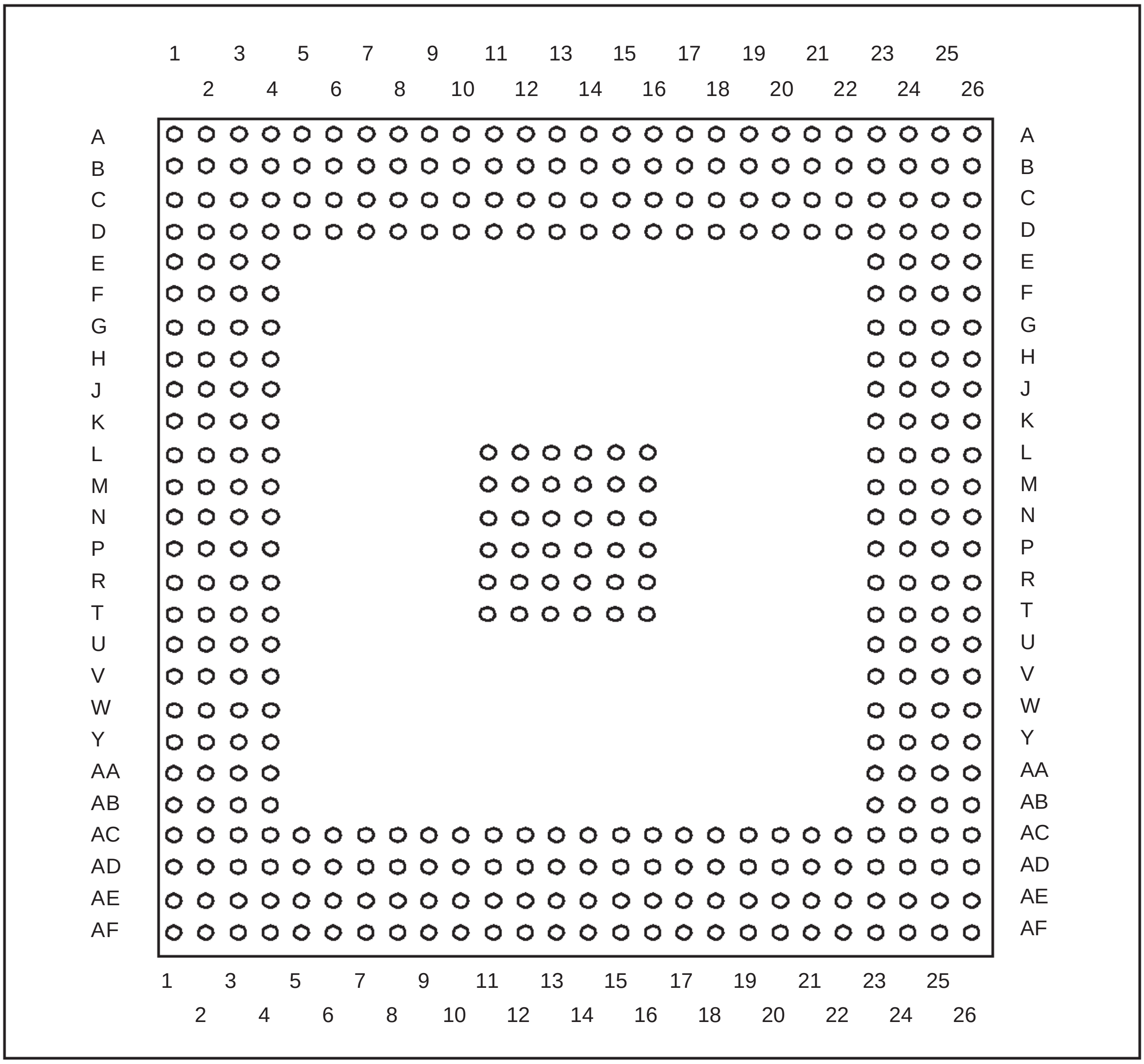


Figure 22-2. 388-pin PBGA Package - PCB Dimensions

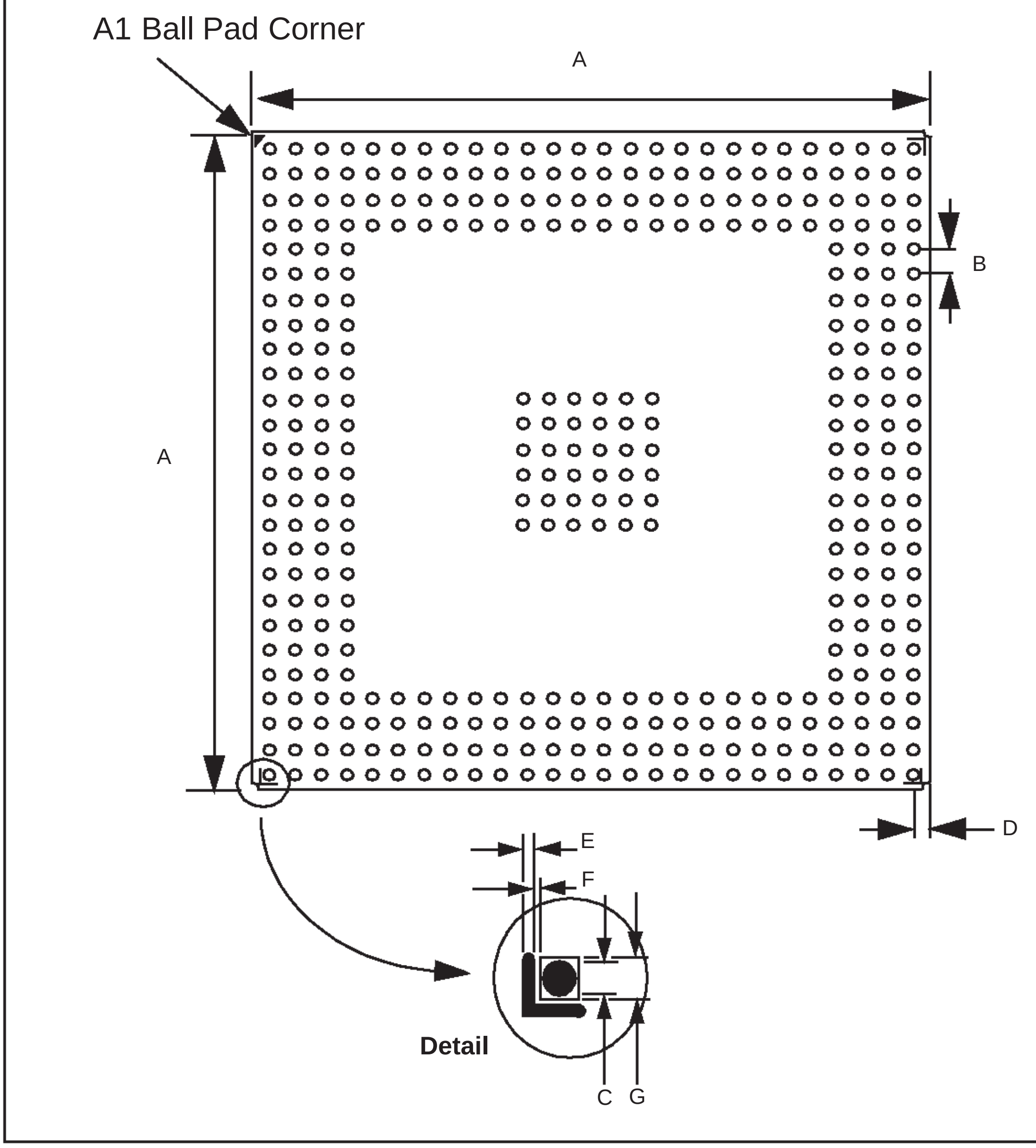


Table 22-1. 388-pin PBGA Package - PCB Dimensions

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 22-3. 388-pin PBGA Package - Dimensions

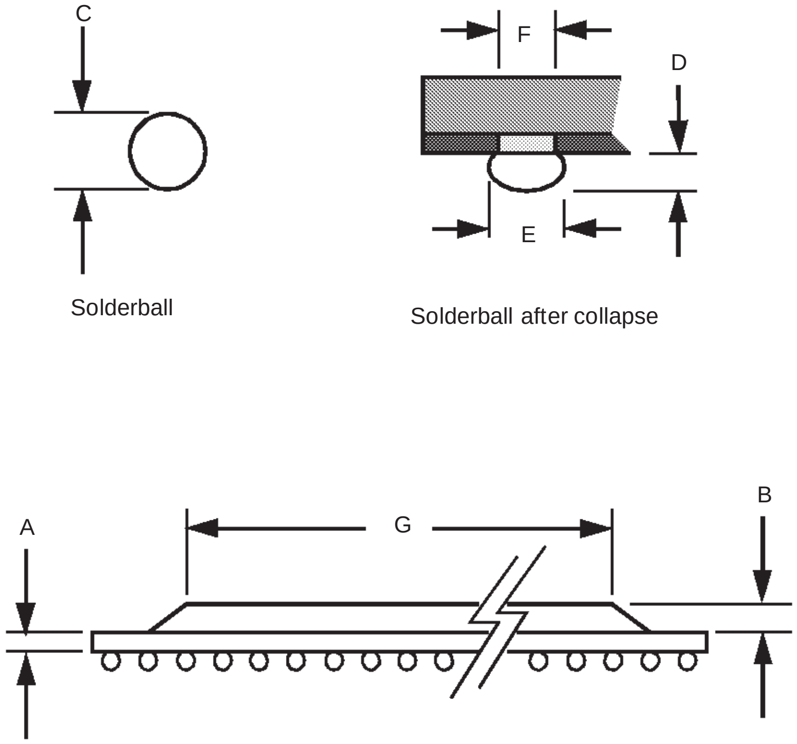


Table 22-2. 388-pin PBGA Package - Dimensions

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	

22.2 388-PIN PACKAGE THERMAL DATA

388-pin PBGA package has a Power Dissipation Capability of 4.5W which increases to 6W when used with a Heatsink.

Structure in shown in Figure 22-4.

Thermal dissipation options are illustrated in Figure 22-5 and Figure 22-6.

Figure 22-4. 388-Pin PBGA structure

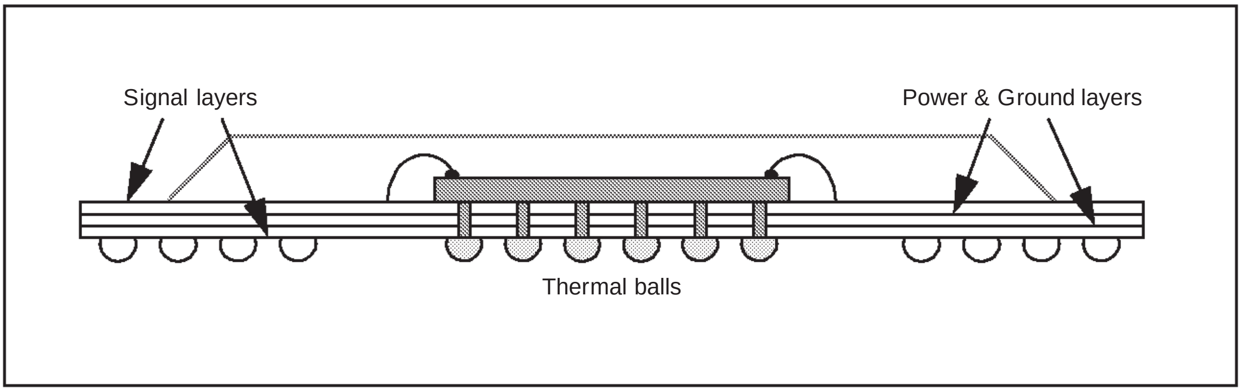
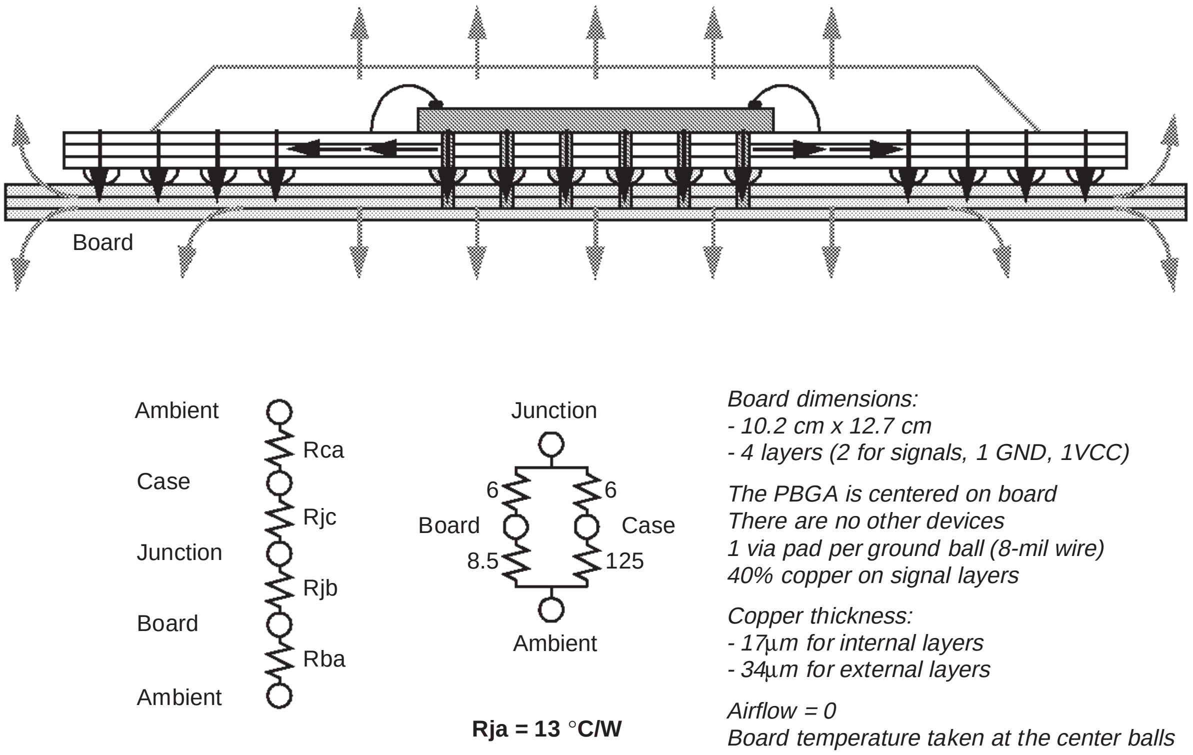


Figure 22-5. Thermal dissipation without heatsink



Rja = 13 °C/W

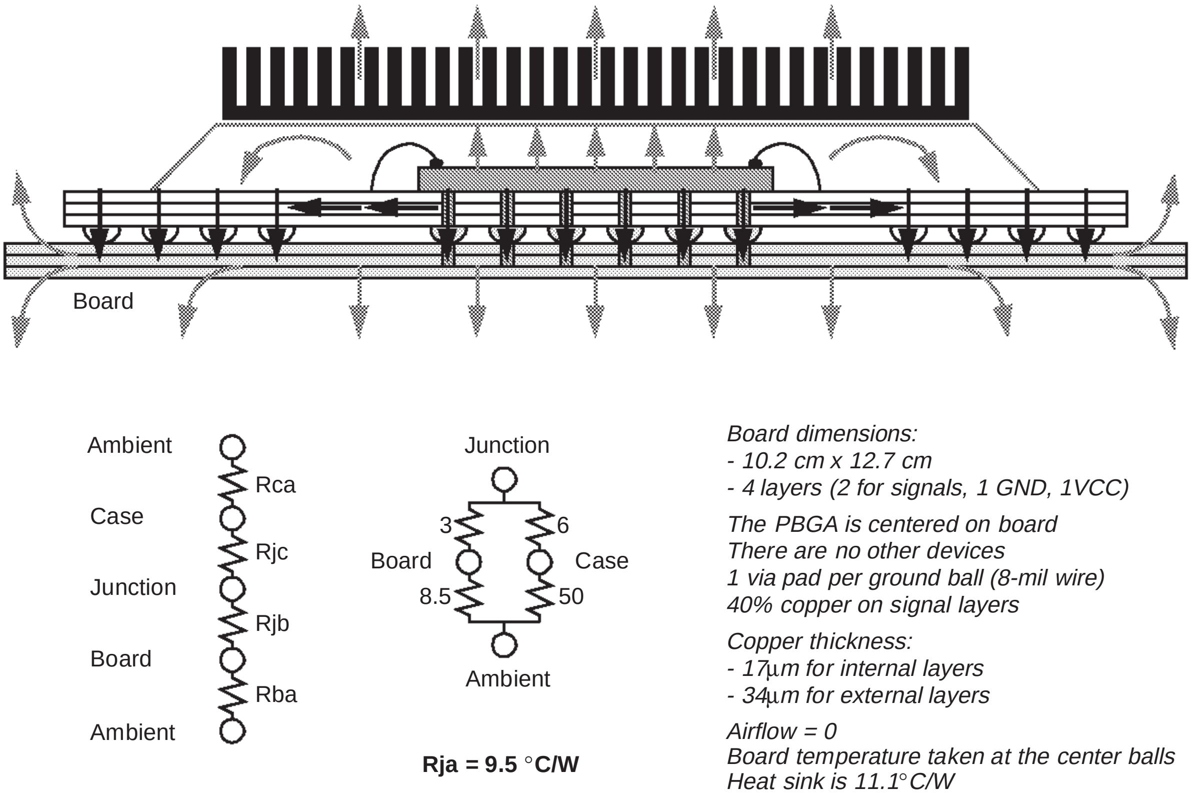
Board dimensions:
- 10.2 cm x 12.7 cm
- 4 layers (2 for signals, 1 GND, 1VCC)

The PBGA is centered on board
There are no other devices
1 via pad per ground ball (8-mil wire)
40% copper on signal layers

Copper thickness:
- 17µm for internal layers
- 34µm for external layers

Airflow = 0
Board temperature taken at the center balls

Figure 22-6. Thermal dissipation with heatsink



Board dimensions:
- 10.2 cm x 12.7 cm
- 4 layers (2 for signals, 1 GND, 1VCC)

The PBGA is centered on board
There are no other devices
1 via pad per ground ball (8-mil wire)
40% copper on signal layers

Copper thickness:
- 17µm for internal layers
- 34µm for external layers

Airflow = 0
Board temperature taken at the center balls
Heat sink is 11.1°C/W

23. BOARD LAYOUT

23.1 THERMAL DISSIPATION

Thermal dissipation of the STPC depends mainly on supply voltage. As a result, when the system does not need to work at 3.3V, it may be to reduce the voltage to 3.15V for example. This may save few 100's of mW.

The second area that can be concidered is unused interfaces and functions. Depending on the application, some input signals can be grounded, and some blocks not powered or shutdown. Clock speed dynamic adjustment is also a solution that can be used along with the integrated power management unit.

The standard way to route thermal balls to internal ground layer implements only one via pad for each ball pad, connected using a 8-mil wire.

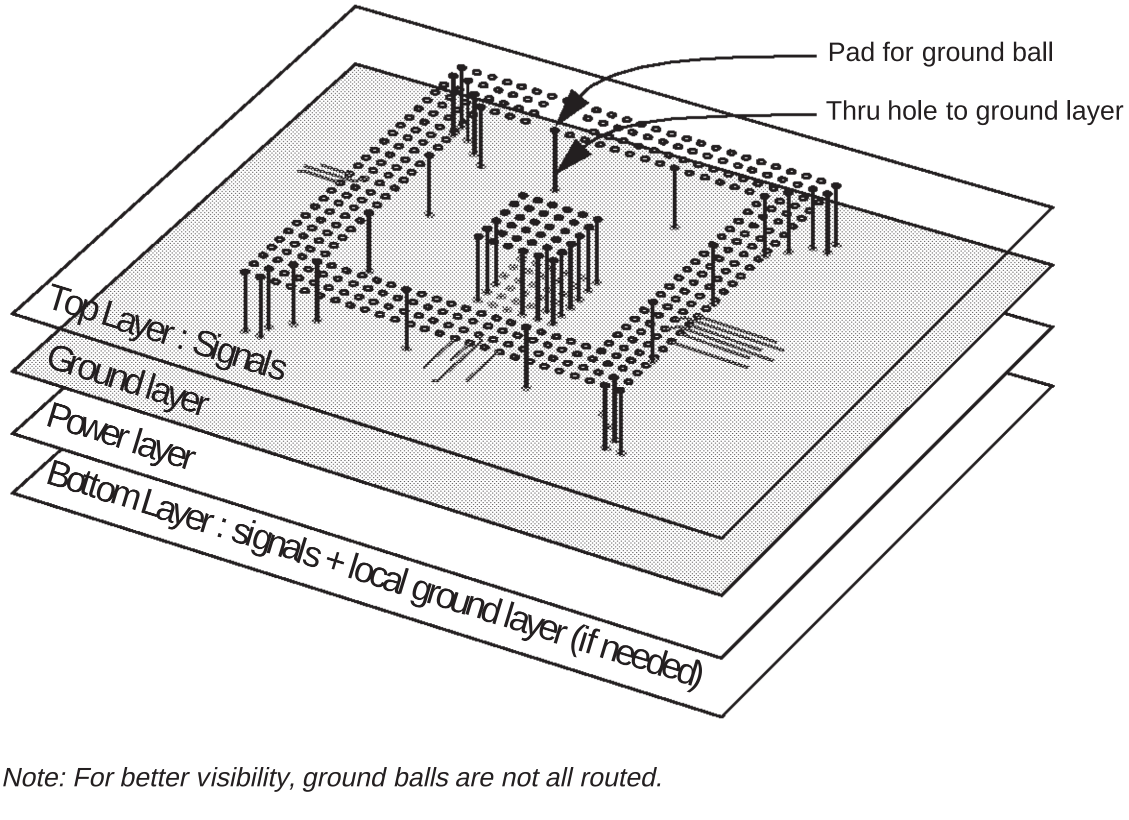
With such configuration the Plastic BGA 388 package dissipates 90% of the heat through the ground balls, and especially the central thermal balls which are directly connected to the die, the remaining 10% is dissipated through the case. Adding a heat sink reduces this value to 85%.

As a result, some basic rules have to be applied when routing the STPC in order to avoid thermal problems.

First of all, the whole ground layer acts as a heat sink and ground balls must be directly connected to it as illustrated in Figure 23-1.

If one ground layer is not enough, a second ground plane may be added on the solder side.

Figure 23-1. Ground routing


Pad for ground ball
Thru hole to ground layer
Top Layer : Signals
Ground layer
Power layer
Bottom Layer : signals + local ground layer (if needed)

Note: For better visibility, ground balls are not all routed.

When considering thermal dissipation, the most important - and not the more obvious - part of the layout is the connection between the ground balls and the ground layer.

A 1-wire connection is shown in Figure 23-2. The use of a 8-mil wire results in a thermal resistance of 105°C/W assuming copper is used (418 W/ m.°K). This high value is due to the thickness (34 µm) of the copper on the external side of the PCB.

Considering only the central matrix of 36 thermal balls and one via for each ball, the global thermal resistance is 2.9°C/W. This can be easily improved by using four 10 mil wires to connect to the four vias around the ground pad link as in Figure 23-3. This gives a total of 49 vias and a global resistance for the 36 thermal balls of 0.6°C/W.

The use of a ground plane like in Figure 23-4 is even better.

To avoid solder wicking over to the via pads during soldering, it is important to have a solder mask of 4 mil around the pad (NSMD pad), this gives a diameter of 33 mil for a 25 mil ground pad.

To obtain the optimum ground layout, place the vias directly under the ball pads. In this case no local boar d distortion is tolerated.

The thickness of the copper on PCB layers is typically 34 µm for external layers and 17 µm for internal layers. This means thermal dissipation is not good and temperature of the board is concentrated around the devices and falls quickly with increased distance.

When it is possible to place a metal layer inside the PCB, this improves dramatically the heat spreading and hence thermal dissipation of the board.

Figure 23-2. Recommended 1-wire ground pad layout

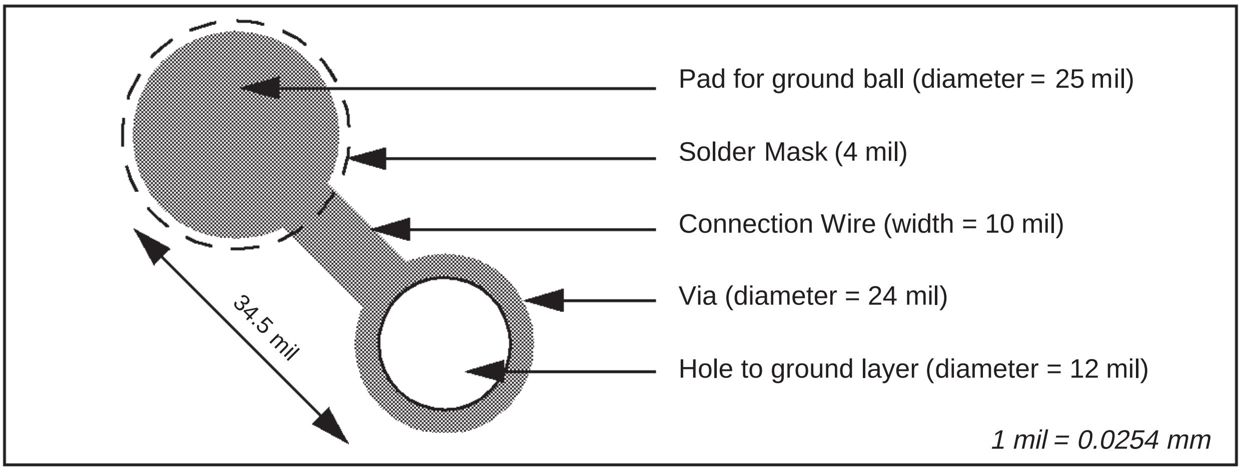

Pad for ground ball (diameter = 25 mil)
Solder Mask (4 mil)
Connection Wire (width = 10 mil)
Via (diameter = 24 mil)
Hole to ground layer (diameter = 12 mil)
34.5 mil
1 mil = 0.0254 mm

Figure 23-3. Recommended 4-wire ground pad layout

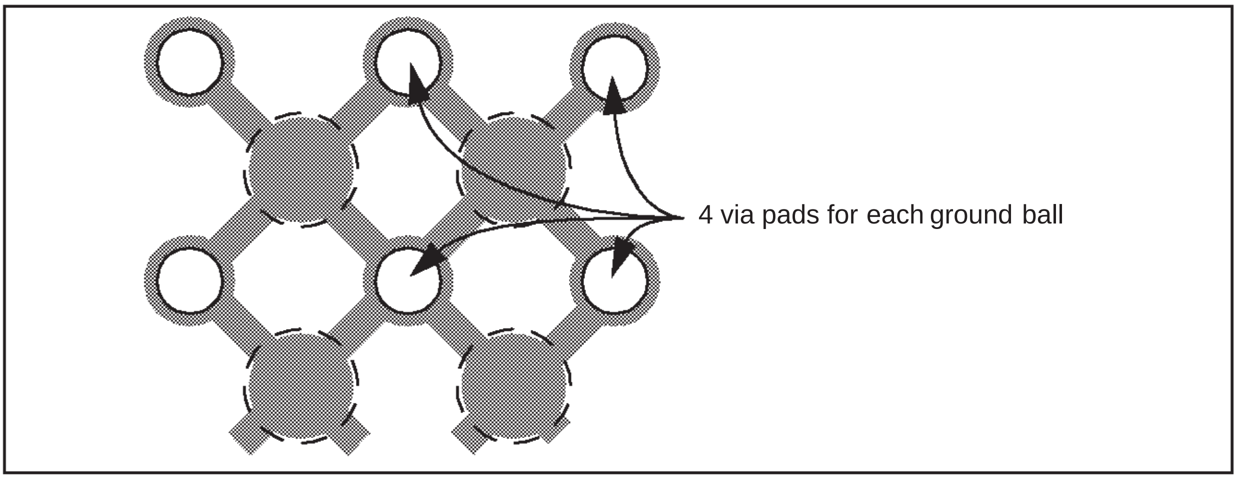
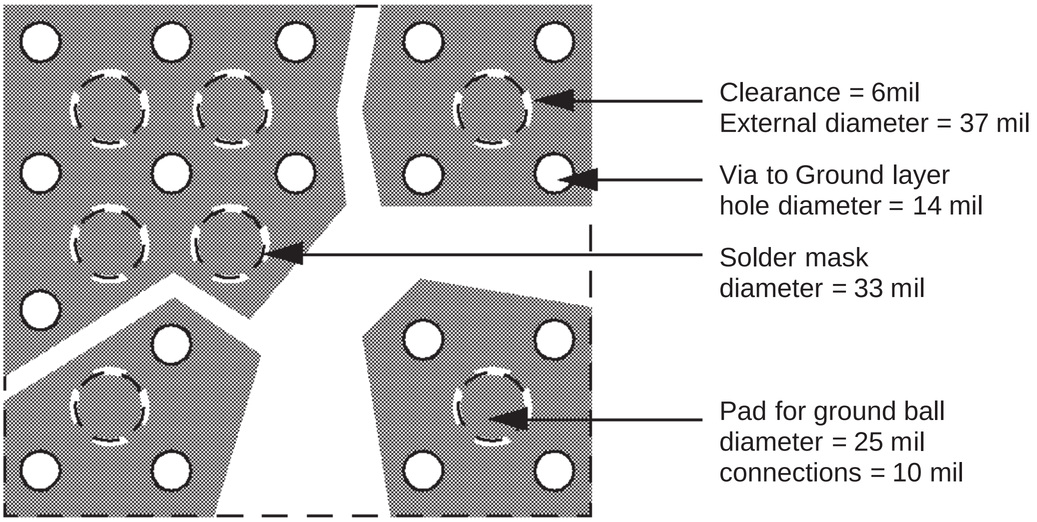


Figure 23-4. Optimum layout for central ground ball



The PBGA Package also dissipates heat through peripheral ground balls. When a heat sink is placed on the device, heat is more uniformely spread throughout the moulding increasing heat dissipation through the peripheral ground balls.

The more via pads are connected to each ground ball, the more heat is dissipated . The only limitation is the risk of lossing routing channels.

Figure 23-5 shows a routing with a good trade off between thermal dissipation and number of routing channels.

Figure 23-5. Global ground layout for good thermal dissipation

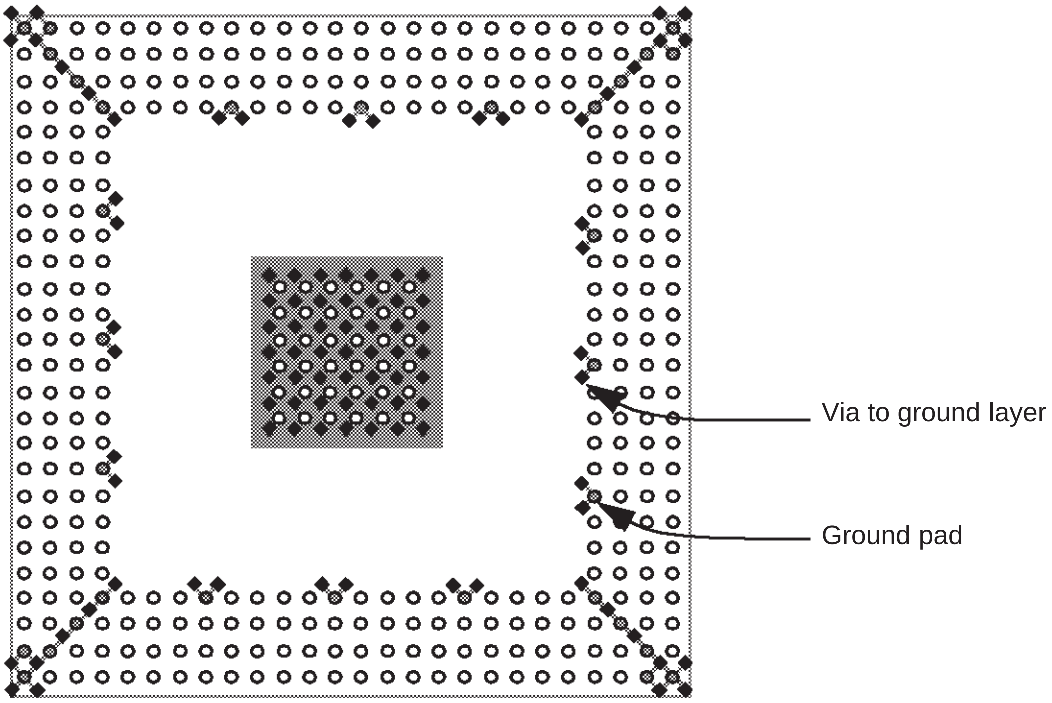
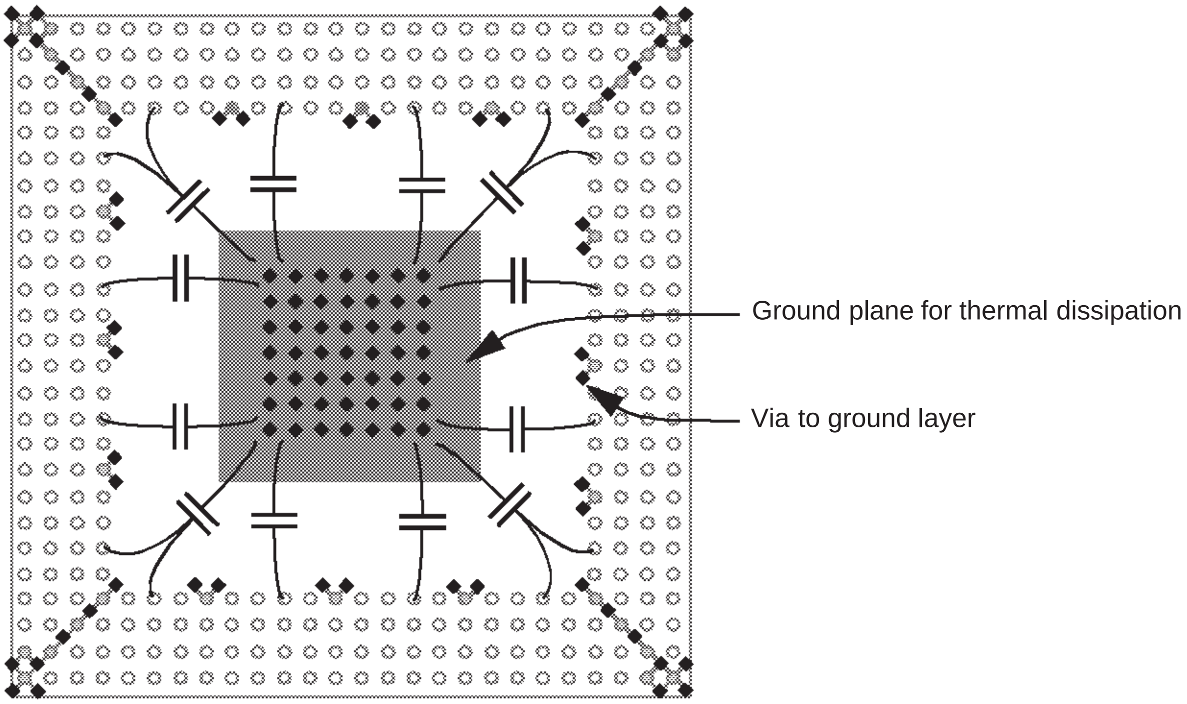


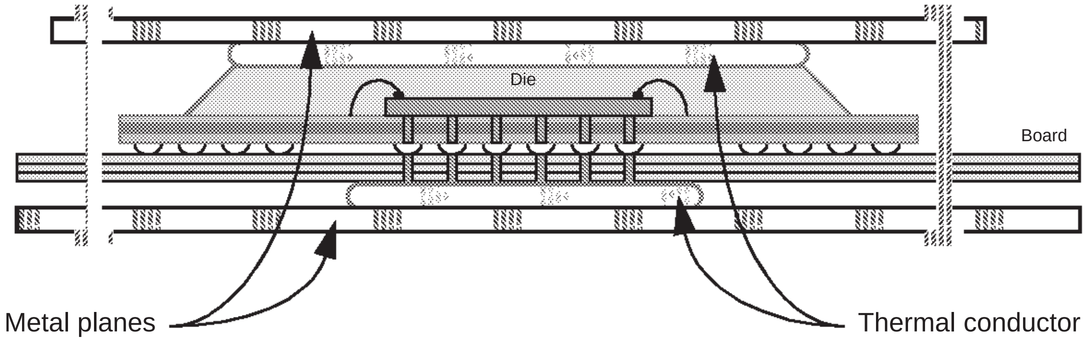
Figure 23-6. Bottom side layout and decoupling



A local ground plane on opposite side of the board as shown in Figure 23-6 improves thermal dissipation. It is used to connect decoupling capacitances but can also be used for connection to a heat sink or to the system's metal box for better dissipation.

This possibility of using the whole system's box for thermal dissipation is very usefull in case of high temperature inside the system and low temperature outside. In that case, both sides of the PBGA should be thermally connected to the metal chassis in order to propagate the heat through the metal. Figure 23-7 illustrates such an implementation.

Figure 23-7. Use of metal plate for thermal dissipation



23.2 HIGH SPEED SIGNALS

Some Interfaces of the STPC run at high speed and have to be carefully routed or even shielded.

Here is the list of these interfaces, in decreasing speed order:

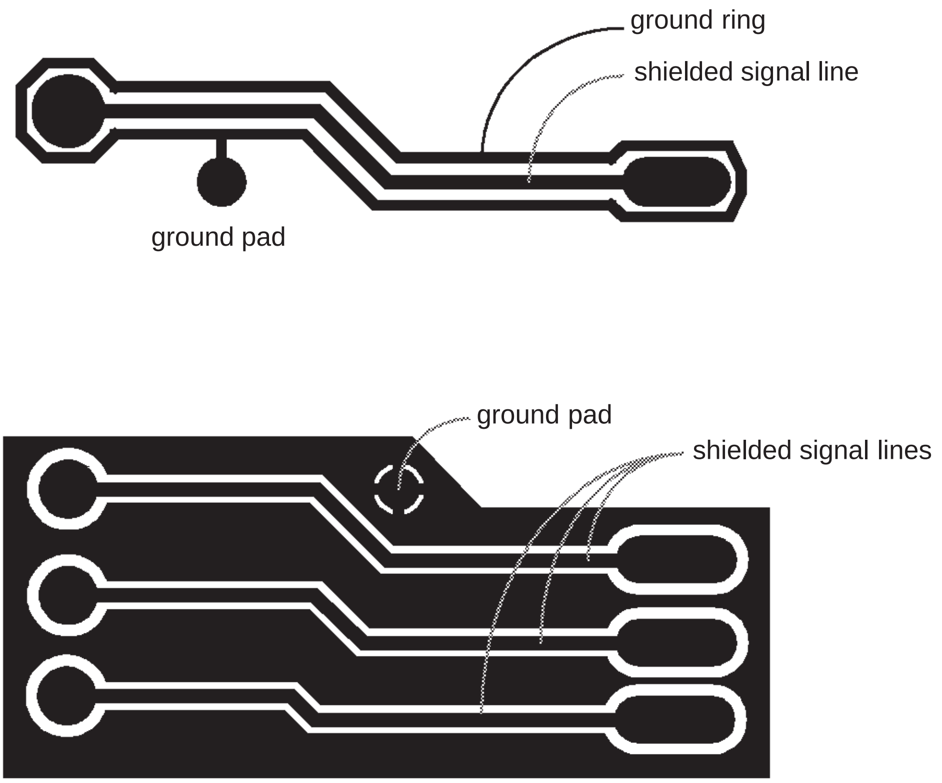
- Memory Interface.
- Graphics and video interfaces
- PCI bus
- 14MHz oscillator stage

All the clocks have to be routed first and shielded for speeds of 27MHz or more. The high speed signals have the same contrainsts as some of the memory interface control signals.

The next interfaces to be routed are Memory, Video/graphics, and PCI.

All the analog noise sensitive signals have to be routed in a separate area and hence can be routed indepedently.

Figure 23-8. Shielding signals



24 ELECTRICAL SPECIFICATIONS

24.1 INTRODUCTION

The electrical specifications in this chapter are valid for the STPC Industrial.

24.2 ELECTRICAL CONNECTIONS

24.2.1 Power/Ground Connections/Decoupling

Due to the high frequency of operation of the STPC Industrial, it is necessary to install and test this device using standard high frequency techniques. The high clock frequencies used in the STPC Industrial 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.

24.2.2 Unused Input Pins

All inputs not used by the designer and not listed in the table of pin connections in Section 6 should be connected either to VDD or to VSS. Connect active-high inputs to VDD through a 20 kΩ (±10%) pull-down resistor and active-low inputs to VSS and connect active-low inputs to VCC through a 20 kΩ (±10%) pull-up resistor to prevent spurious operation.

24.2.3 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.

24.3 ABSOLUTE MAXIMUM RATINGS

The following table lists the absolute maximum ratings for the STPC Industrial device. Stresses beyond those listed under Table 24-1 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 those outlined in Table 24-1 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 24-1) may also result in reduced useful life and reliability.

24.3.1 5V Tolerance

The STPC is capable of running with I/O systems that operate at 5V such as PCI and ISA devices. Certain pins of the STPC tolerate inputs up to 5.5V. Above this limit thecomponent is likely to sustain permanent damage.

All the pin that are V_{5T} have been denoted with a * besides the Signal Name in Table 6-1 .

Table 24-1. Absolute Maximum Ratings

Symbol	Parameter	Minimum	Maximum	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
V_{5T}	5Volt Tolerance	2.5	5.5	
T_{STG}	Storage Temperature	-40	+150	°C
T_{OPER}	Operating Temperature	0	+70	°C
P_{TOT}	Maximum Power Dissipation	-	4.8	W

24.4 DC CHARACTERISTICS

Table 24-2. 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)			80	Mhz
V_{DAC}	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_{ILD}	Input Low Voltage	Except XTALI	-0.3		0.8	V
		XTALI	-0.3		0.5	V
V_{IHD}	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 yet released.

Table 24-3. RAMDAC DC Specification

Symbol	Parameter	Min	Nom	Max
Vref	Voltage Reference	1.00V	1.12V	1.24V
INL	Integrated Non Linear Error	-	-	2 lsb
DNL	Differentiated Non Linear Error	-	-	1lsb
FS	Full Scale	-	-	20mA
FSR	Full Scale Range	14.00 mA	16.50mA	19.00 mA
LSB	Least Significant Byte Size	54uA	63uA	72uA
Zero	Zero Scale @ 7.5IRE Mode	0.95mA	1.44mA	1.90mA
Compare	DAC to DAC matching	-	-	+/- 5%

24.5 AC CHARACTERISTICS

Table 24-5 through Table 24-14 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 24-1. The rising clock edge reference level VREF , and other reference levels are shown in Table 24-4 below for the STPC Industrial. Input or output signals must cross these levels during testing.

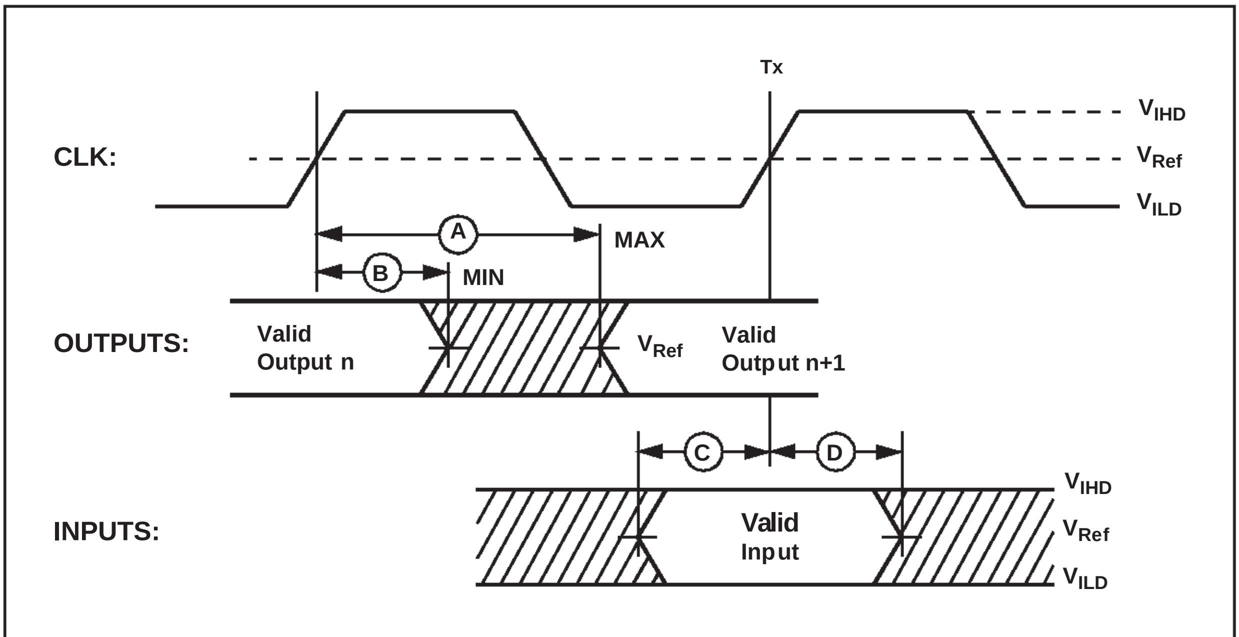
Figure 24-1 shows output delay (A and B) and input setup and hold times (C and D). Input setup and hold times (C and D) are specified minimums, defining the smallest acceptable sampling window a synchronous input signal must be stable for correct operation.

Table 24-4. 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 24-1.

Figure 24-1. 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

Table 24-5. DRAM Bus AC Timing

Name	Parameter	Min	Max	Unit
t1	HCLK to RAS#[3:0] valid		16	ns
t2	HCLK to CAS#[7:0] bus valid		16	ns
t3	HCLK to MA[11:0] bus valid		16	ns
t4	HCLK to MWE# valid		16	ns
t5	HCLK to MD[63:0] bus valid		16	ns
t6	MD[63:0] Generic setup	9		ns
t7	GCLK2X to RAS#[3:0] valid		17	ns
t8	GCLK2X to CAS#[7:0] valid		17	ns
t9	GCLK2X to MA[11:0] bus valid		17	ns
t10	GCLK2X to MWE# valid		17	ns
t11	GCLK2X to MD[63:0] bus valid		17	ns
t12	MD[63:0] Generic hold	5		ns

Table 24-6. PCI Interface AC TImings

Name	Parameter	Min	Max	Unit
t13	PCI_CLKI to FRAME# valid		15	nS
t14	PCI_CLKI to TRDY# valid		15	nS
t15	PCI_CLKI to IRDY# valid		15	nS
t16	PCI_CLKI to STOP# valid		15	nS
t17	PCI_CLKI to DEVSEL# valid		15	nS

Table 24-7. Graphics Adapter (VGA) AC Timing

Name	Parameter	Min	Max	Unit
t18	DCLK to VSYNC valid		27	ns
t19	DCLK to HSYNC valid		27	ns

Table 24-8. IPC Interface AC Timings

Name	Parameter	Min	Max	Unit
t20	XTALO to DACK_EN[2:0] valid		71	nS
t21	XTALO to TC valid		68	nS
t22	IRQ_MUX Input setup to ISACLK2X	0	-	nS
t23	DREQ_MUX[1:0] Input setup to ISACLK2X	0	-	nS

Table 24-9. PCMCIA Interface AC Timing

Name	Parameters	Min	Max	Units
t24	Input setup to ISACLK2X	24		nS
t25	Input hold from ISACLK2X	5		nS
t28	ISACLK2X to IORD	-	55	nS
t29	ISACLK2X to IORW	-	55	nS
t30	ISACLK2X to AD[25:0]	-	25	nS
t31	ISACLK2X to OE#	2	55	nS
t32	ISACLK2X to WE#	2	55	nS
t33	ISACLK2X to DATA[15:0]	0	35	nS
t34	ISACLK2X to INPACK	2	55	nS
t35	ISACLK2X to CE1#	7	65	nS
t36	ISACLK2X to CE2#	7	65	nS
t37	ISACLK2X to RESET	2	55	nS

Table 24-10. Parallel Interface AC Timing

Name	Parameters	Min	Max	Units
t37	STROBE# to BUSY setup	0	-	nS
t38	PD bus to AUTPFD# hold	0	-	nS
t39	PB bus to BUSY setup	0	-	nS

Table 24-11. Keyboard Interface AC Timing

Name	Parameters	Min	Max	Units
t40	Input setup to KBCLK	5	-	nS
t41	Input hold to KBCLK	1	-	nS
t42	KBCLK to KBDATA	-	12	nS

Table 24-12. Mouse Interface AC Timing

Name	Parameters	Min	Max	Units
t43	Input setup to MCLK	5	-	nS
t44	Input hold to MCLK	1	-	nS
t45	MCLK to MDATA	-	12	nS

Table 24-13. Local Bus Interface AC Timing

Name	Parameters	Min	Max	Units
t46	PRDY# Input hold to HCLK	2		nS
t47	PD[15:0] Input hold to HCLK	2		nS
t48	PRDY# Input setup to HCLK	1	-	nS
t49	PD[15:0] Input setup to HCLK	2	4	nS
t50	HCLK to PA bus	-	15	nS
t51	HCLK to PD bus	-	15	nS
t52	HCLK to PWR0#	-	15	nS
t53	HCLK to PWR1#	-	15	nS
t54	HCLK to PRD0#	-	15	nS
t55	HCLK to PRD1#	-	15	nS
t56	HCLK to FCS0#	-	15	nS
t57	HCLK to FCS1#	-	15	nS
t58	HCLK to IOCS#[3:0]	-	15	nS

Table 24-14. TFT Interface Timing

Name	Parameters	Min	Max	Units
t59	DCLK to FPLINE		15	nS
t60	DCLK to R[2]		15	nS
t61	DCLK to R[3]		15	nS
t62	DCLK to R[4]		15	nS
t63	DCLK to R[5]		15	nS
t64	DCLK to G[2]		15	nS
t65	DCLK to G[3]		15	nS
t66	DCLK to G[4]		15	nS
t67	DCLK to G[5]		15	nS
t68	DCLK to B[2]		15	nS
t68	DCLK to B[3]		15	nS
t69	DCLK to B[4]		15	nS
t70	DCLK to B[5]		15	nS
t71	DCLK to FPFRAME		15	nS

24.6 UPDATE HISTORY FOR ELECTRICAL CHARACTERISTICS CHAPTER

The following changes have been made to the Electrical Characteristics Chapter on 12/10/99.

Section	Change	Text
24.3.1	Added	Paragraph added to detail 5V tolerance of the chip.

The following changes have been made to the Electrical Characteristics Chapter on 12/10/99.

Section	Change	Text
Table 24-2.	Replaced	VILD XTALI value changed from 0.9V to 0.5V
Table 24-3.	Replaced	RAMDAC DC Specification vref min changed from 1.04V to 1.00V
	Replaced	RAMDAC DC Specification vref nom changed from 1.135V to 1.12V
	Replaced	RAMDAC DC Specification vref max changed from 1.16V to 1.24V
	Replaced	FSR min changed from 15.75mA to 14.00mA
	Replaced	FSR nom changed from 16.50mA to 16.50mA
	Replaced	FSR max changed from 17.40mA to 19.00mA
	Replaced	LSB min changed from 61uA to 54uA
	Replaced	LSB nom changed from 65uA to 63uA
	Replaced	LSB max changed from 69uA to 72uA
Table 24-9.	Removed	Timings t26 & t27 duplicates
	Removed	Timings t38 to t45 duplicates
	Replaced	t28, 20ns changed to 55ns
	Replaced	t29, 20ns changed to 55ns
	Replaced	t30, 23ns changed to 25ns
	Replaced	t33, min none changed to 0ns, max 25ns changed to 35ns
	Replaced	t35 & t36, min none changed to 7ns, max 20ns changed to 65ns
	Replaced	t37, min none changed to 2ns, max 20ns changed to 55ns
Table 24-11.	Replaced	t40, min none changed to 5ns, max 12ns changed to none
	Replaced	t42, min 2ns changed to none, max none changed to 12ns
Table 24-12.	Replaced	t43, min 0ns changed to 5ns
	Replaced	t34, min 0ns changed to 1ns, max 0.02ns changed to none

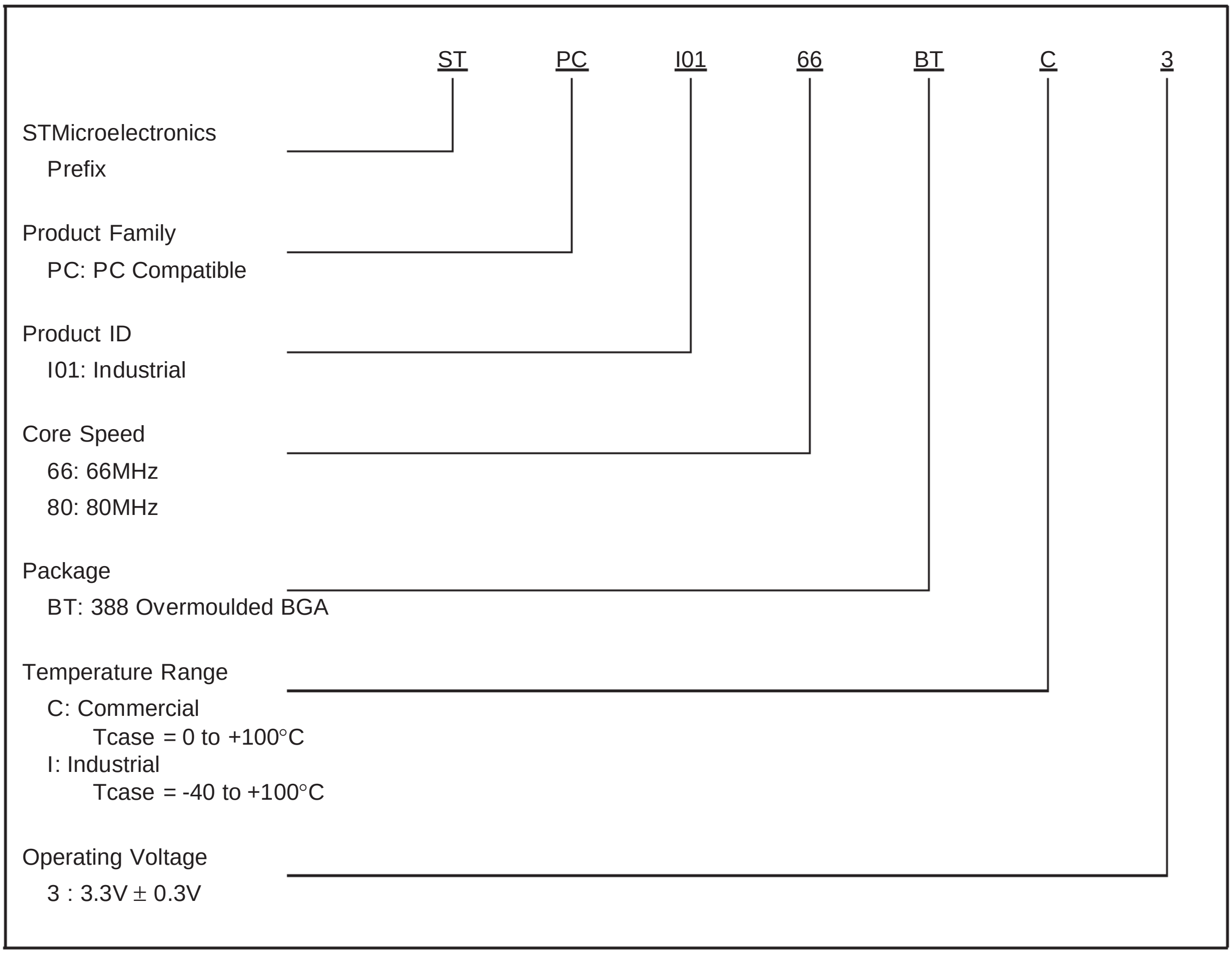
The following changes have been made to the Electrical Characteristics Chapter on 12/08/99.

Section	Change	Text
Table 23-3	Added	RAMDAC DC Specification.
Table 23-5	Updated	DRAM Interface Timings t1 to t12.
Table 23-6	Added	PCI Timing AC.
Table 23-7	Updated	Graphics Adapter (VGA) AC Timings t18 & t19.
Table 23-8	Added	IPC Interface AC Timings.
Table 23-9	Updated	PCMCIA Interface AC Timings t28 to t45.
Table 23-11	Updated	Keyboard Interface AC Timings t40 to t42
Table 23-12	Updated	Mouse Interface AC Timings t43 to t45
Table 23-13	Updated	Localbus Interface AC Timings t46 to t58.
Table 23-14	Updated	TFT Interface AC Timings t59 to t71.

Update History for Electrical Characteristics chapter

25 ORDERING DATA

25.1 ORDERING CODES



ST PC I01 66 BT C 3

STMicroelectronics
Prefix

Product Family
PC: PC Compatible

Product ID
I01: Industrial

Core Speed
66: 66MHz
80: 80MHz

Package
BT: 388 Overmoulded BGA

Temperature Range
C: Commercial
Tcase = 0 to +100°C
I: Industrial
Tcase = -40 to +100°C

Operating Voltage
3 : 3.3V ± 0.3V

25.2 AVAILABLE PART NUMBERS

Part Number	Core Frequency (MHz)	CPU Mode	Tcase Range (C)	Operating Voltage (V)
STPCI0166BTC3	66	DX	0°C to +100°C	3.3V ± 0.3V
STPCI0180BTC3	80	DX		
STPCI0166BTI3	66	DX	-40°C to +100°C	
STPCI0180BTI3	80	DX		

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.