1 Overview

About This Chapter

Read this chapter to find out

- A general overview of the AIC-7870 host adapter chip
- Features of the AIC-7870 host adapter chip
- Reference documents applicable to the AIC-7870 host adapter chip

▼▼▼▼▼ 1

Introduction

The AIC-7870 provides advanced host adapter features in a single chip with a SCSI-2 bus controller and a full featured PCI 32-bit bus master with zero wait state transfer capability including PCI enhanced data transfer commands. The AIC-7870 chip incorporates a dedicated processor, the SCSI PhaseEngine™ (RISC Sequencer), which executes a SCSI command described by a Sequencer Control Block (SCB). Sixteen SCBs may be stored in the internal SCB Array and with the addition of an external SRAM, a maximum of 256 SCBs may be stored for execution. These SCBs are executed independently of the SCSI target ID in the order that they are received. The SCB is a data structure which contains all information necessary for the execution of the command. The sequencer in the chip handles all phases of the SCSI bus, including the Disconnect/Reconnect and Command Complete message. On the PCI host side, bus master transfers are made in a 64-bit address space at up to the maximum burst rate of 133 MBytes/sec with data buffering of 256 bytes.

The AIC-7870 also provides a memory port for external access, an 8-bit ROM/EEPROM (for add-in card local BIOS support), a serial 1-bit EEPROM (for nonvolatile SCSI bus device and parameter storage, and/or adapter board assembly/serial/revision information), an 8/9-bit SRAM (for SCB expansion), and external board logic control.

Feature Summary

- Automatic data threshold selection
- Power-down modes
- Scatter/Gather operations supported
- Extremely low SCSI command overhead
- Data residue reported on underruns
- One interrupt per command completion, multiple command completions may be queued on a single interrupt
- Queued commands per Target/LUN
- Overlapped command execution
- Modify Data Pointers message handled
- Tagged Queuing supported
- SCSI Configured AutoMagically (SCAM level 1 support)
- Data path from PCI bus to SCSI bus internally byte parity protected
- Hardware address breakpoint capability for software debug
- External BIOS ROM option with in-place BIOS update (EEPROM) capability
- Serial SEEPROM read/write option for saving SCSI bus device and adapter board parameters

- External SRAM (8/9-bit/parity) option for SCB expansion
- Memory Port Arbitration option including a lock feature for sharing external BIOS, SRAM and SEEPROM
- Device ID option for exchanging internal default value with an external value
- External Board Control option for controlling host adapter logic external to the AIC-7870 from the PCI bus

SCSI Features

- Fast (10 MHz) data transfers
- Wide data transfers
- Differential controls
- Flexible configuration
  - One 8-bit Single-ended, Fast
  - One 8-bit Differential, Fast
  - One 16-bit Single-ended, Fast
  - One 16-bit Differential, Fast
- Auto SCSI bus PIO
- Wide data connector indicator
- LED indicator control (SCSI busy, diagnostics, external ID clock) or general purpose control
- Selectable SCSI output active negation
- SCSI termination power down control or general purpose control
- SCAM level 1 support
- Digital filtering for incoming REQ and ACK signals

PCI 32-bit Interface

- Direct pin out connection to PCI 32-bit bus interface
- PCI bus master with zero wait state 32-bit memory data transfers at 133 MBytes/sec, capable of leading and trailing 32-bit boundary offset bytes, with a 32-bit address range within a selected 64-bit address page
- Supports both PCI single and dual address cycles
- PCI bus master/slave timing referenced to PCI signal PCLK (33.3 MHz max)
- Buffered PCI signal PCLK output for adapter card logic usage
- PCI bus master programmable Latency Timer, Cache Size, and Interrupt Line Select registers
- PCI bus access of AIC-7870 device registers from both PCI I/O and memory address spaces
- Supports exchange of internal device ID default value with an external value

- Supports medium PCI target device-select response time
- Supports enhanced PCI system memory data Read and Write commands
- Cache line streaming capability
- Supports PCI bus address and data parity generation and checking
- Supports PCI PERR and SERR requirements
- Supports 32-bit external ROM read access
- Data transfers may be selected to be initiated by CACHESIZE or data level thresholds
- Data FIFO data flush for transfers to system memory
- IRQA# interrupt generation from hardware, firmware and software controlled sources
- Supports reduced power requirements when not performing master data transfers

Data Buffer

- Data FIFO provides 256 bytes of storage, dual-ported RAM, with parity per byte
- Programmable data and cache size threshold levels to initiate PCI bus master requests
- Early FIFO full status
- Multi byte-width data ports: 8 (PCI), 2 (SCSI), 1 (sequencer or driver) byte with byte parity
- Byte write parity generation and byte read parity checking
- Read/write capable address counters
- Partial quad-word detection and adjustment
- Starting address byte offset capable

Scratch RAM

- 64 bytes of dual-ported SRAM, accessible by sequencer and host drivers
- Byte parity protected

Sequencer (SCSI PhaseEngine)

- RISC instruction per clock design
- Clock rate selectable for 8 or 10 MHz operation
- SRAM microcode storage, 512 29-bit words plus byte parity
- Sixteen instruction group types
- Operation may be paused by maskable interrupt condition or software driver
- Diagnostic single-step and address breakpoint
- SLEEP mode for chip power reduction

SCB Array

- Internal SCB array for storage of sixteen SCBs.
- SCB array expandable to 256 SCBs with external SRAM
- SCBPTR register for SCB page (32-bytes) selection
- SCBCNT register for auto SCB address increment
- SCB write supports 32-bit burst transfers
- Queue In FIFO (selectable entry depths of 16 or 255 with stored count status)
- Queue In CNT (selectable entry depths of 16 or 255 with stored count status)
- SCB, Queue In and Queue Out are byte parity protected

Memory Port

- Read only, write only, or read-modify-write for sequencer instruction access of SRAM/SEEPROM/external logic
- Read only or write only for software driver PCI access of SRAM/ROM/ EEPROM/SEEPROM/external logic
- Single ported 8 or 9-bit SRAM (20 nsec access) with direct connection for 8 KBytes
- Single ported 8-bit ROM/EEPROM (150 nsec access) with direct connection for 64 KBytes
- External SRAM presence buffered PCLK pin
- Serial SEEPROM with register based access with hardware timer support
- Multidevice arbitration (two wire) for sharing memory port connected devices
- SRAM, ROM, EEPROM three line interface for read/write control
- SEEPROM four line interface
- External logic six line interface

AIC-7870 Block Diagram

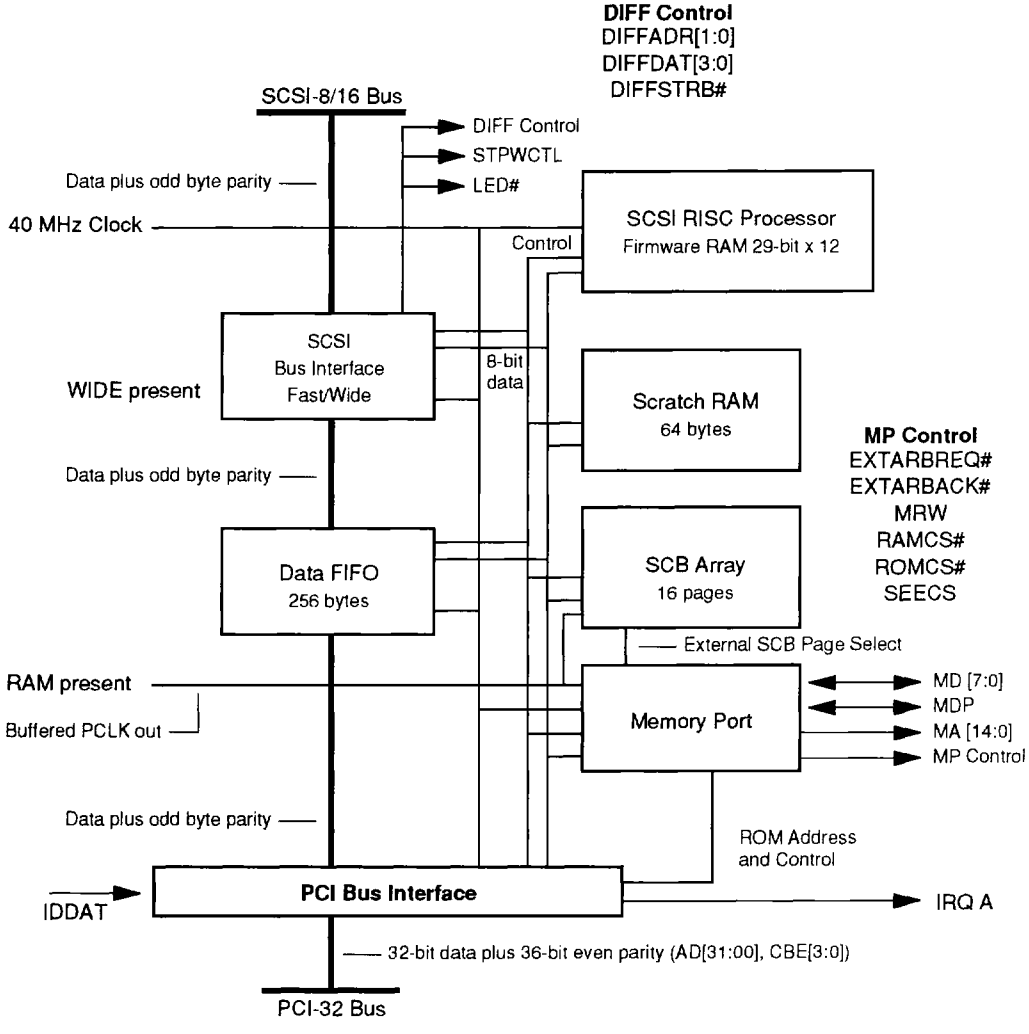


Figure 1-1. AIC-7870 Block Diagram

Reference Documents

Reference is made in this specification to the following additional documents:

- Peripheral Component Interconnect (PCI) Local Bus Interface Specification, Rev. 2.0
- AIC-7770 Specification 707001, Adaptec Inc.
- SCSI-2 Specification - ANSI Standard # X3T9.2 Rev 10h

10 System Cycles

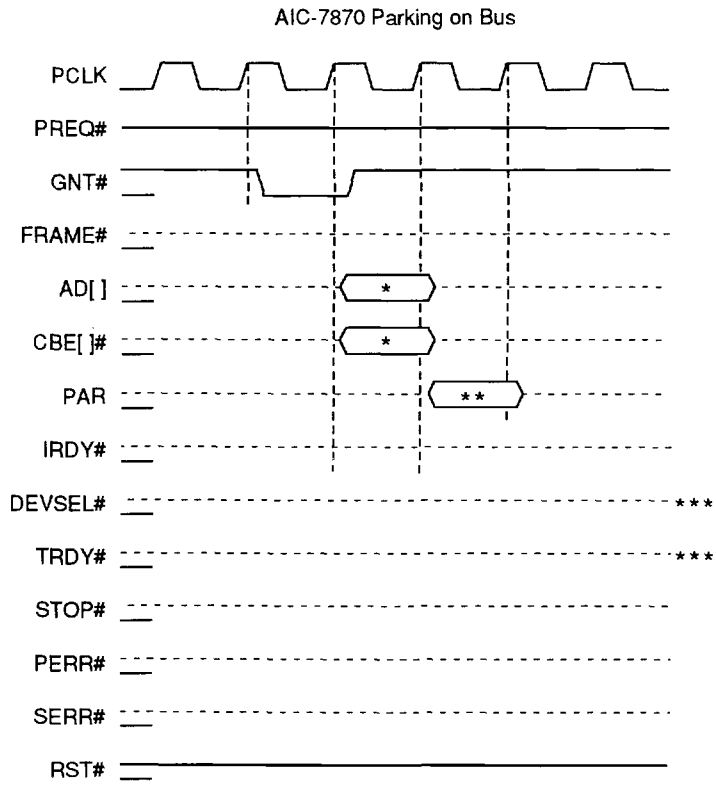
About This Chapter

Read this chapter to find out

- The PCI Master Bus Cycles
- The PCI Slave Bus Cycles

▼▼▼▼▼ 10

PCI Master Bus Cycles



- - - - - - - - Float Condition (signals other than AD, CBE, and PAR require pull-up resistors on the system board)

* Current internal values

** Correct parity for AD, CBE

*** Target response

Figure 10-1. Master Parking on Bus

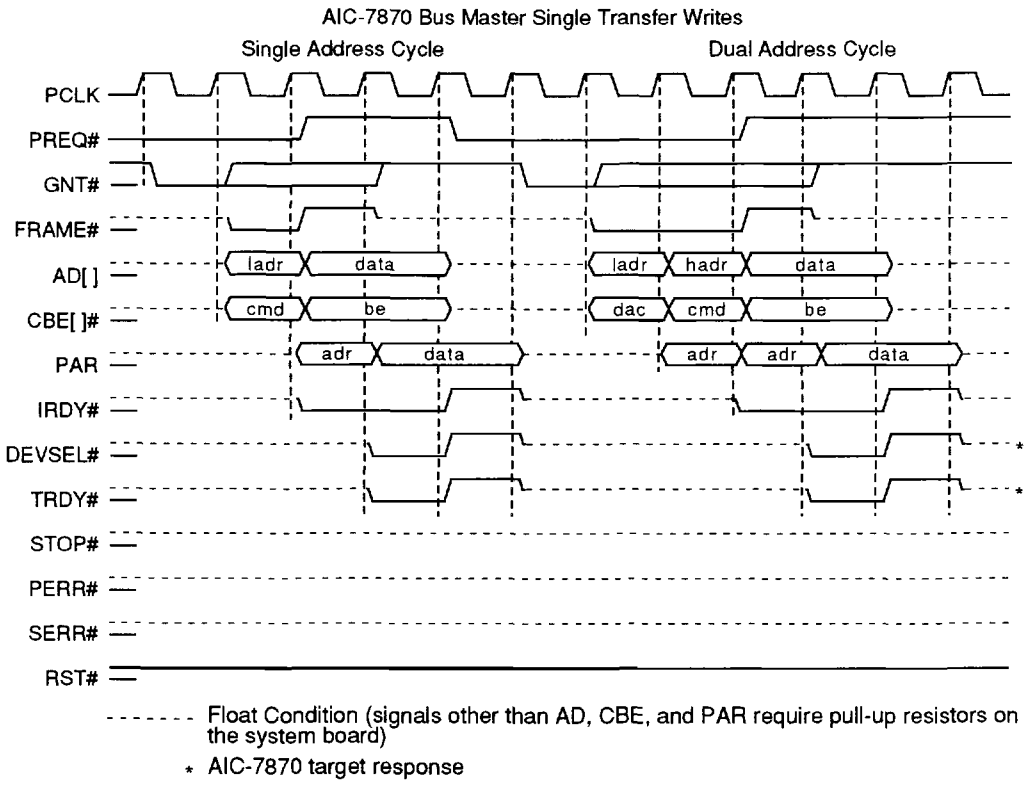


Figure 10-2. Master Single Transfer Write

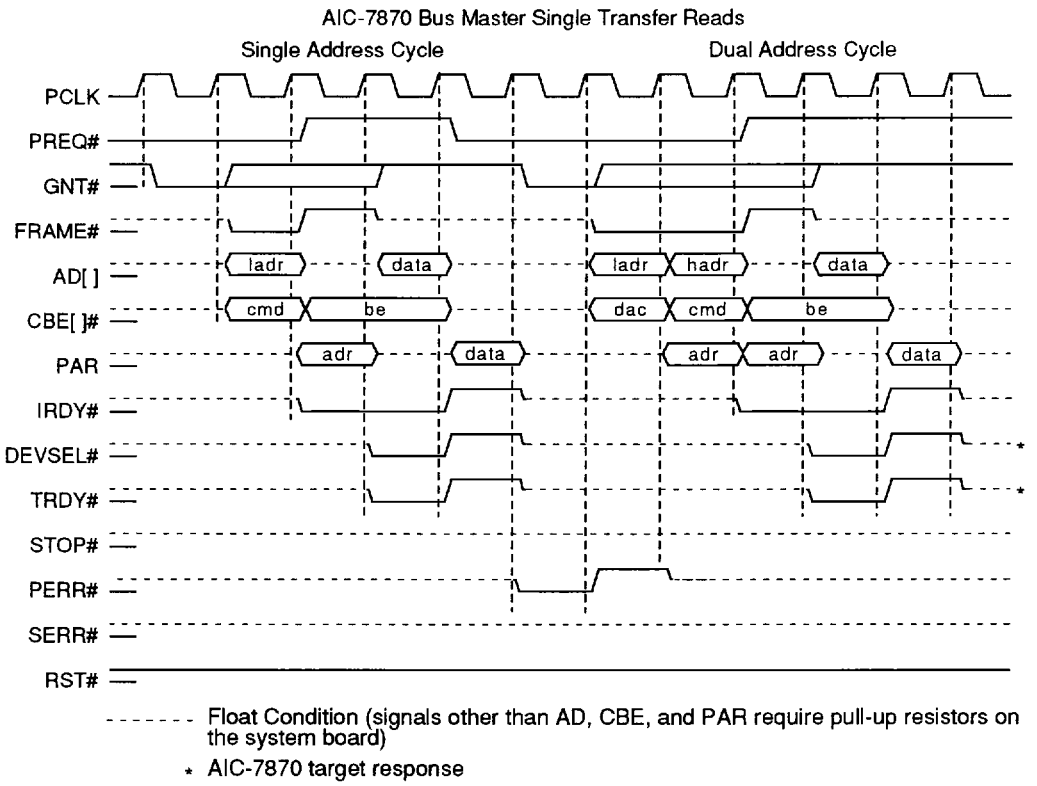
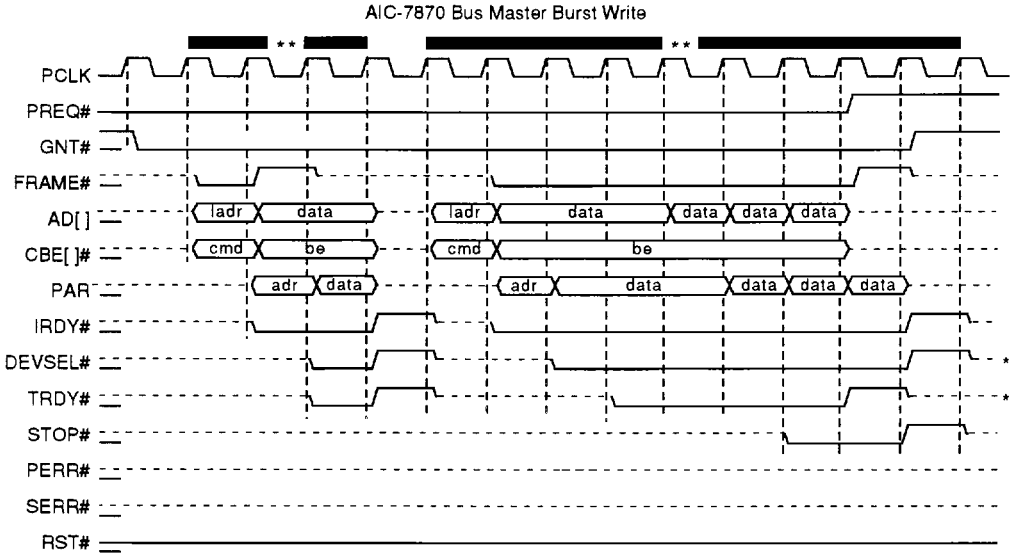


Figure 10-3. Master Single Transfer Read



------ Float Condition (signals other than AD, CBE, and PAR require pull-up resistors on the system board)

* AIC-7870 target response

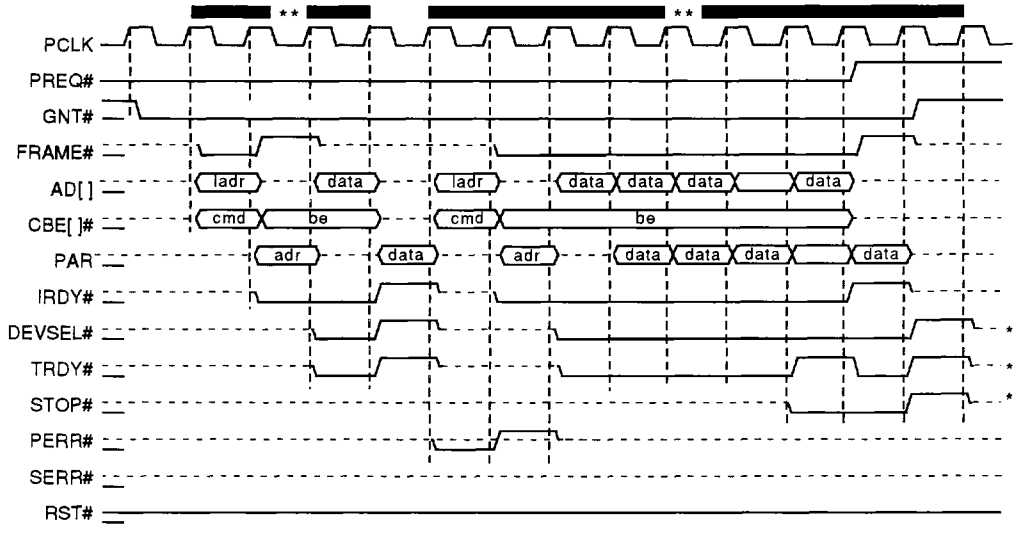
** AIC-7870 MRDC burst to cache line boundary

*** AIC-7870 MRDC burst for 16-byte cache line

The AIC-7870 is capable of cache line streaming, with no wait states, however the target disconnects at end of cache line

Figure 10-4. Master Burst Transfer Write

AIC-7870 Bus Master Burst Read



- - - - - - Float Condition (signals other than AD, CBE, and PAR require pull-up resistors on the system board)

* AIC-7870 target response

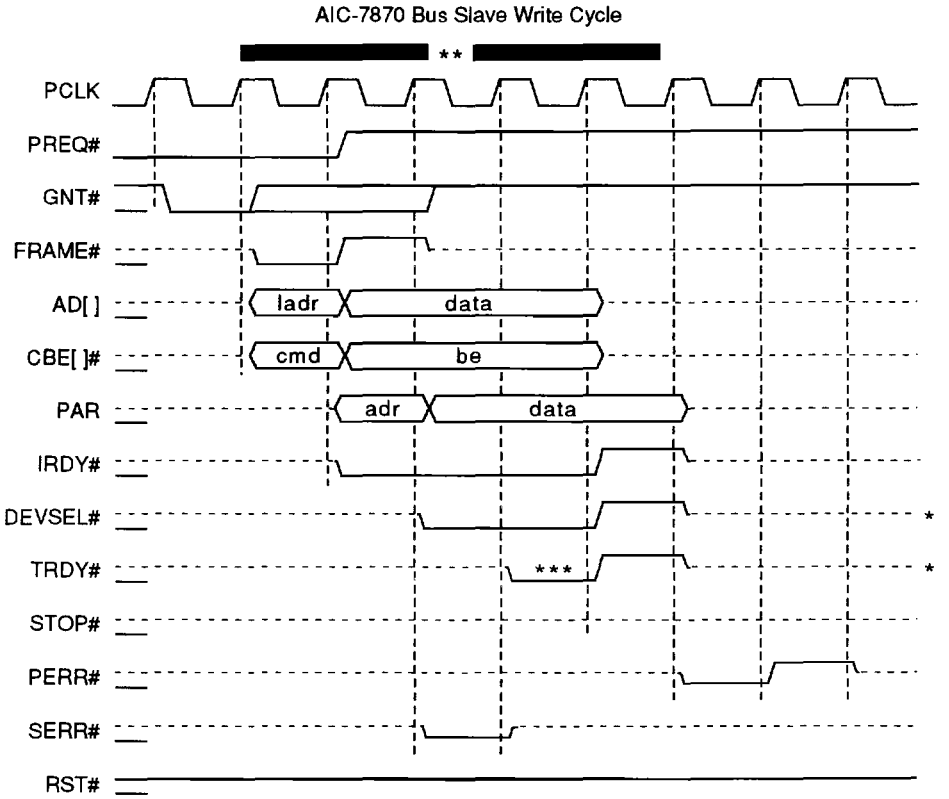
** AIC-7870 MRDC burst to cache line boundary

*** AIC-7870 MRDC burst for 16-byte cache line

The AIC-7870 is capable of cache line streaming, with no wait states, however the target disconnects at end of cache line

Figure 10-5. Master Burst Transfer Read

PCI Slave Bus Cycles



- - - - - - - - Float Condition (signals other than AD, CBE, and PAR require pull-up resistors on the system board)

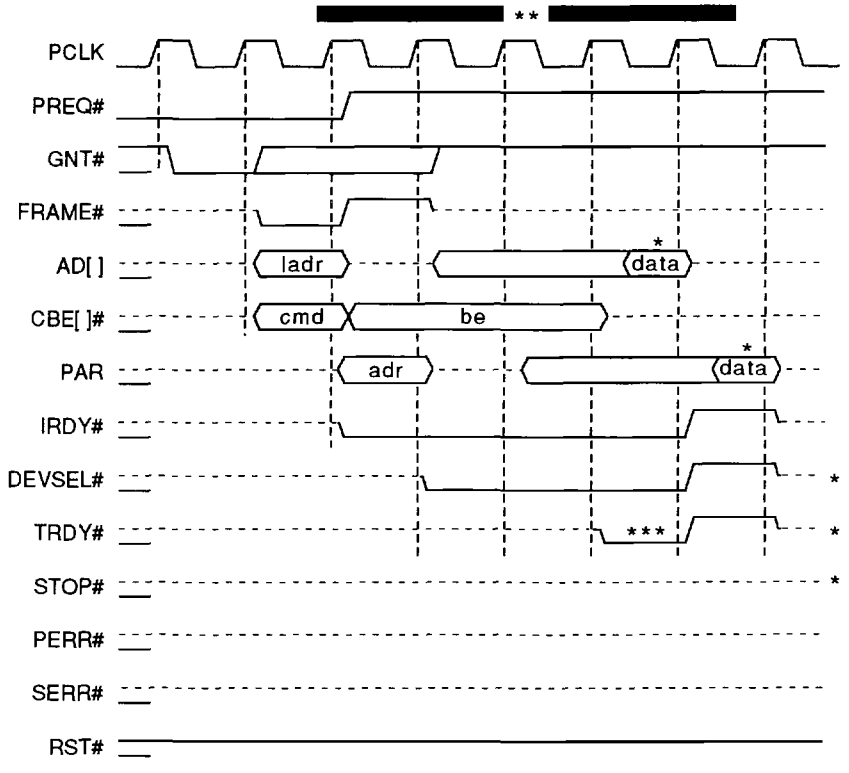
* AIC-7870 target response

** AIC-7870 register access cycle

*** May be delayed one or more clocks for some addresses

Figure 10-6. Slave Single Transfer Write

AIC-7870 Bus Slave Read Cycle



- - - - - - - - Float Condition (signals other than AD, CBE, and PAR require pull-up resistors on the system board)

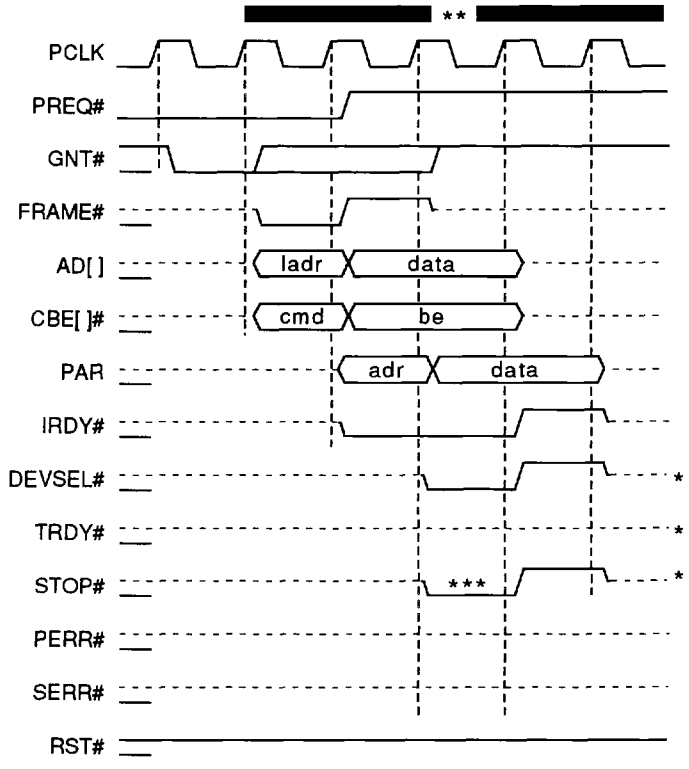
* AIC-7870 target response

** AIC-7870 register access cycle

*** May be delayed one or more clocks for some addresses

Figure 10-7. Slave Single Transfer Read

AIC-7870 Bus Slave Retry Cycle



- - - - - - - - Float Condition (signals other than AD, CBE, and PAR require pull-up resistors on the system board)

* AIC-7870 target response

** AIC-7870 register access cycle

*** May be delayed one or more clocks for some addresses

Figure 10-8. Slave Retry

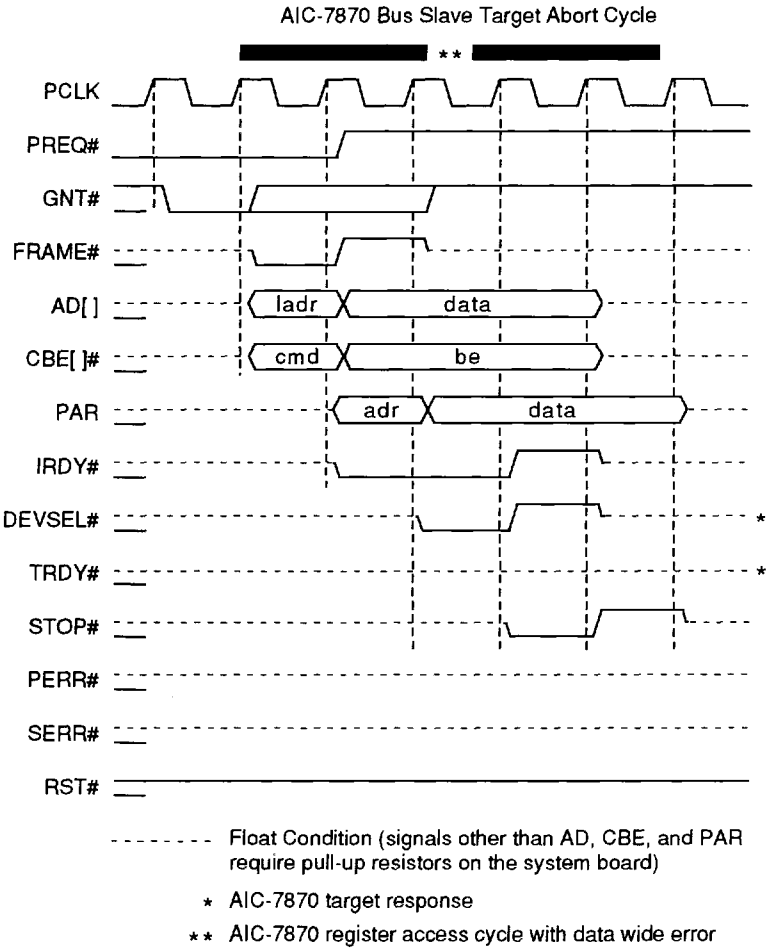
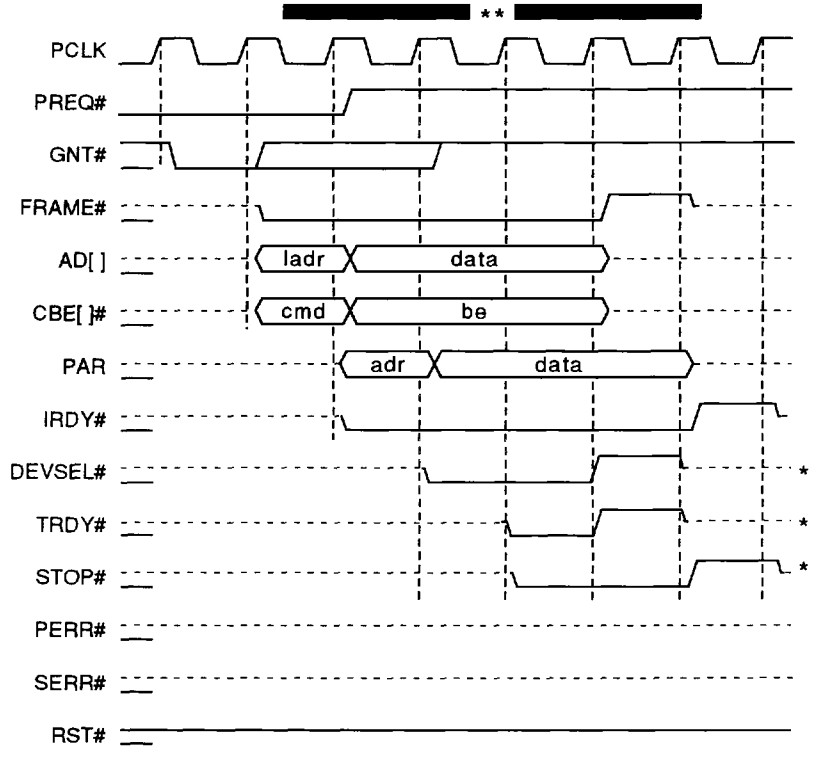


Figure 10-9. Slave Target Abort (width error)

AIC-7870 Bus Slave Burst Disconnect Cycle



- - - - - - - - Float Condition (signals other than AD, CBE, and PAR require pullup resistors on the system board)

* AIC-7870 target response

** AIC-7870 register access cycle with unsupported burst attempt

Figure 10-10. Slave Burst Disconnect

11 Differential Timing

About This Chapter

Read this chapter to find out

- The timing of the differential cycles

▼▼▼▼▼ 11

Differential Cycles

SCSI Initiator Arbitration Selection

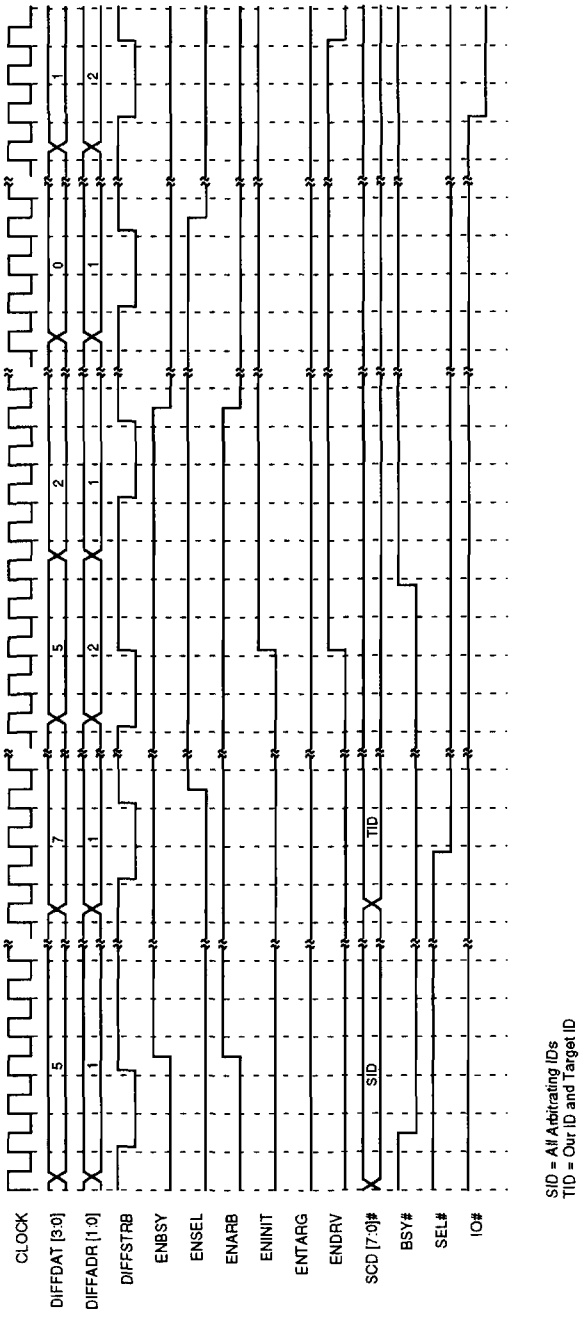


Figure 11-1. SCSI Initiator Arbitration Selection

SCSI Target Arbitration Reselection

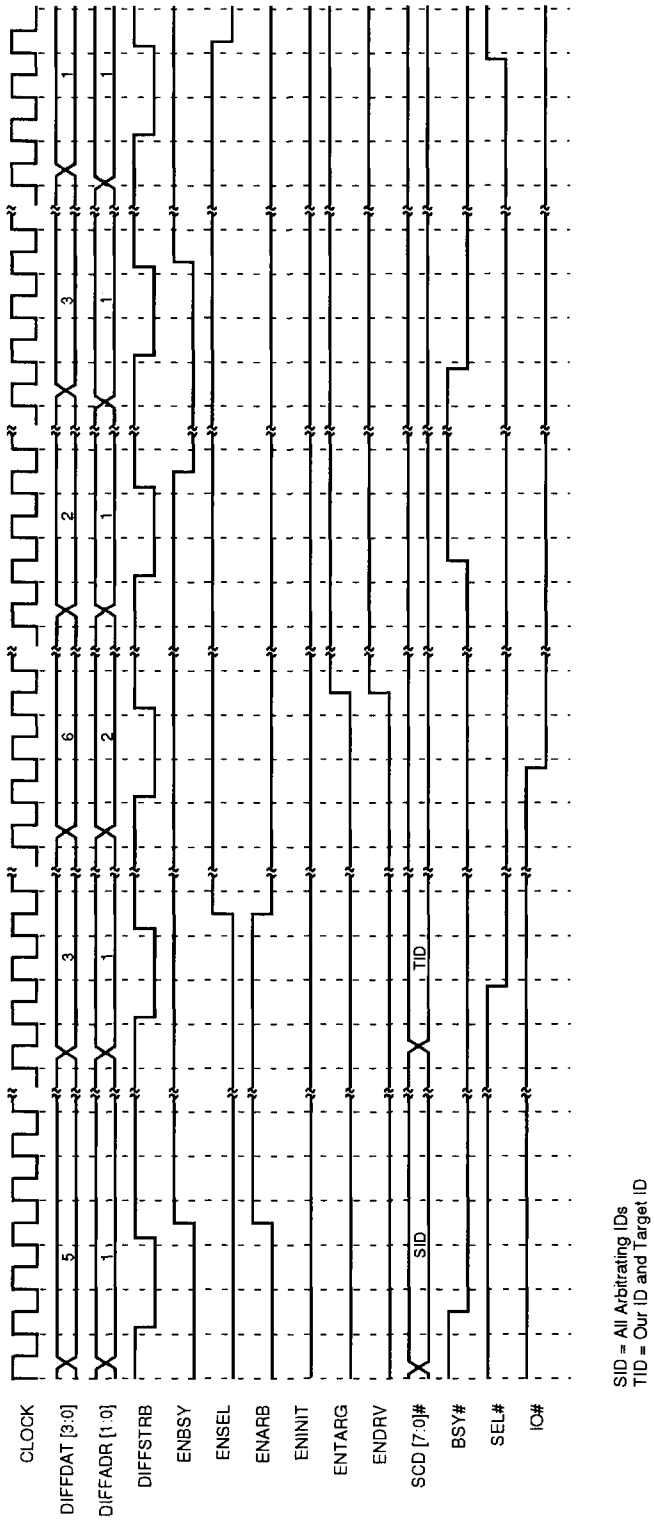


Figure 11-2. SCSI Target Arbitration Reselection

SCSI Initiator Reselection

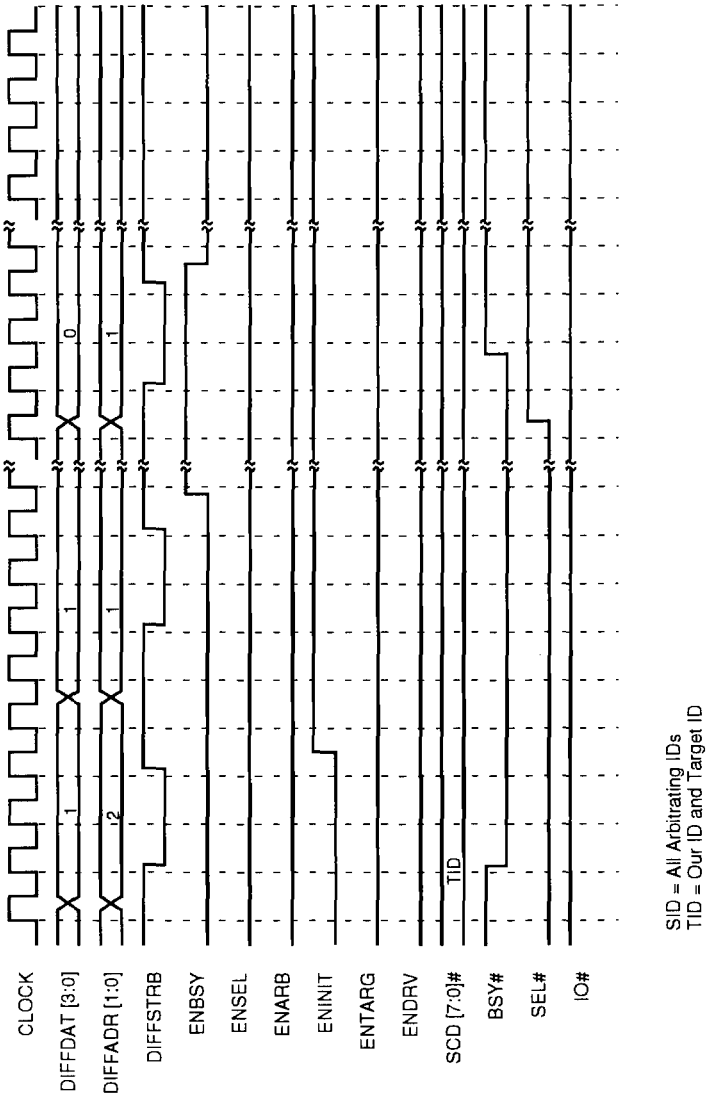


Figure 11-3. SCSI Initiator Reselection

SCSI Target Selection

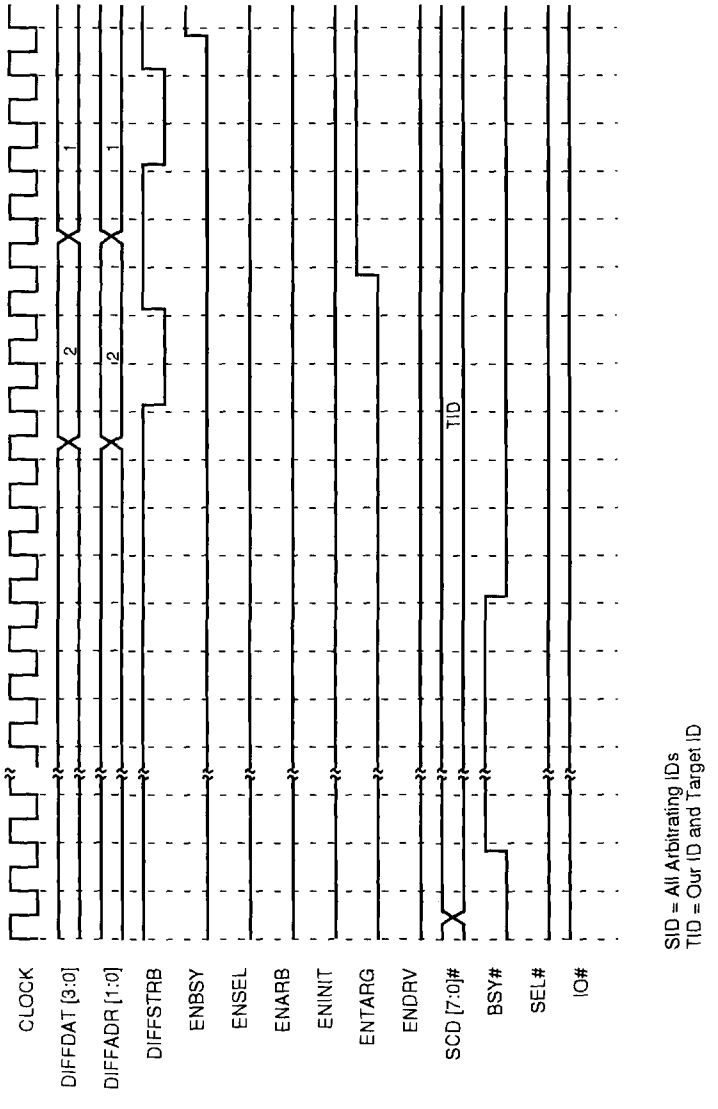


Figure 11-4. SCSI Target Selection

12 Package Outline

About This Chapter

Read this chapter to find out

- A package outline of the AIC-7870

▼▼▼▼▼ 12

AIC-7870 Package Outline

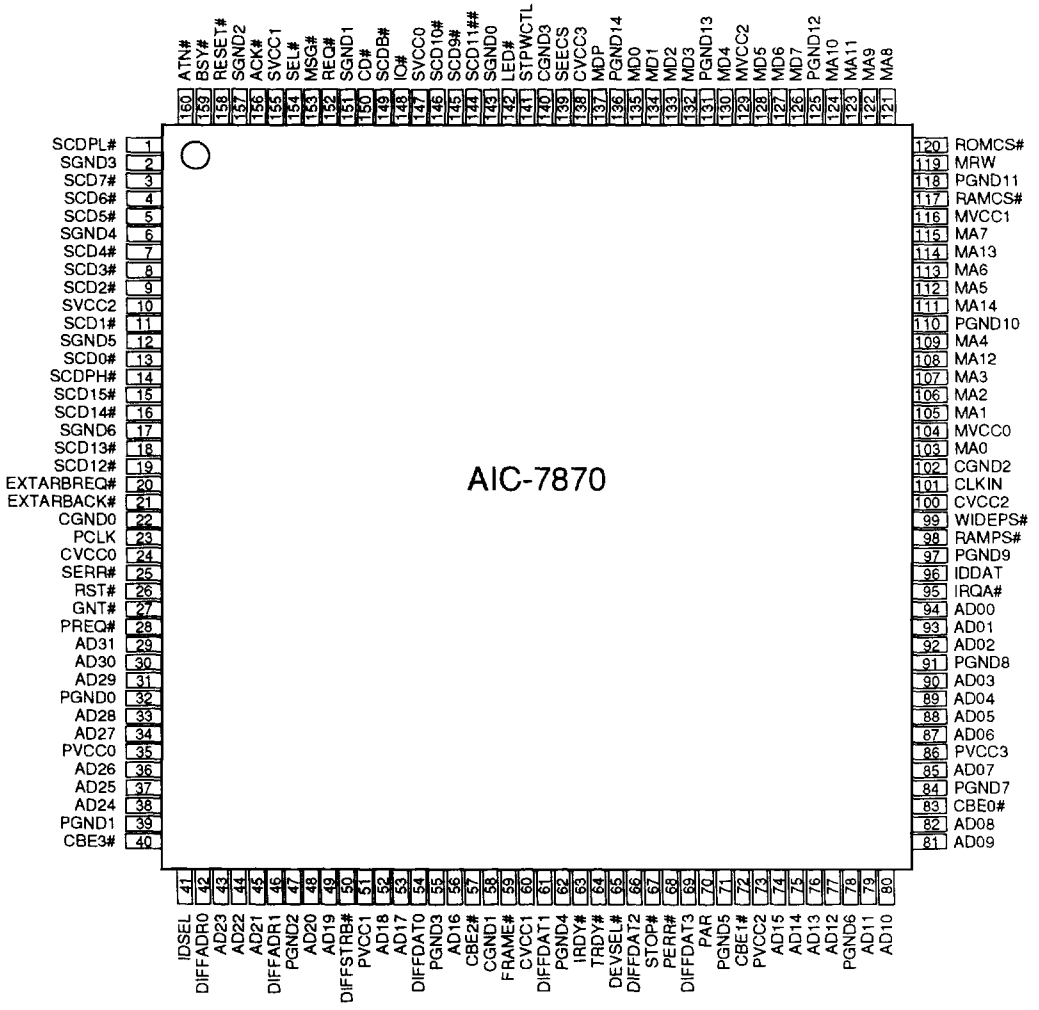


Figure 12-1. AIC-7870 Package Outline

2 Pin Description

About This Chapter

Read this chapter to find out

- A summary of the pin pad type and signal assignment
- A detailed description of the supported PCI-32 pin and Memory Port pin signals

▼▼▼▼▼ 2

Pin Pad Type and Signal Assignment Summary

The pins are summarized here, listing the name, pin number, if the pin is an input, and the type and drive of the outputs. The type definitions are listed in Table 2-1. A more complete description follows the summary.

Table 2-1. Pin Type Definitions

Type	Definition
I	Input
3ST/#	Tri-state Output/Min Drive Current in mA
OD/#	Open Drain Output/Min Drive Current in mA
NOD/#	Negation Capable Open Drain Output/Min Sink Current in mA

Host Interface

Table 2-2. Host Interface

Signal Name	Pin Number	I/O Driver PCI Master	I/O Driver PCI Target	Notes
AD[31:00]	29-31, 33, 34, 36-38, 43-45, 48, 49, 52, 53, 56, 74-77, 79-82, 85, 87-90, 92-94	I, 3ST/6	I, 3ST/6	
CBE[3:0]#	40, 57, 72, 83	3ST/6	I	
DEVSEL#	65	I	3ST/6	1
FRAME#	59	3ST/6	I	1
GNT#	27	I	NA	
IDSEL	41	NA	I	
IRDY#	63	3ST/6	I	1
PAR	70	I, 3ST6	I, 3ST6	
PCLK	23	I	I	
PREQ#	28	3ST/6	NA	
PERR#	68	I, 3ST/6	I, 3ST/6	1
RST#	26	I	I	
SERR#	25	NA	OD/6	1
STOP#	67	I	3ST/6	1
TRDY#	64	I	3ST/6	1
IRQA#	95	3ST/4	3ST/4	1

[1] Signal requires external pull-up resistors.

SCSI Interface

Table 2-3. SCSI Interface

Signal Name	Pin Number	I/O Driver	Notes
SCD15#-SCD8#	15, 16, 18, 19, 144, 146, 145, 149	I, NOD/48	1, 2
SCD7#-SCD0#	3-5, 7-9, 11, 13	I, NOD/48	2
SCDPH#	14	I, NOD/48	1, 2
SCDPL#	1	I, NOD/48	2
CD#	150	I, NOD/48	2
IO#	148	I, NOD/48	2
MSG#	153	I, NOD/48	2
REQ#	152	I, NOD/48	2
ACK#	156	I, NOD/48	2
RESET#	158	I, OD/48	
SEL#	154	I, OD/48	
BSY#	159	I, OD/48	
ATN#	160	I, NOD/48	2
DIFFDAT(3:0)	69, 66, 61, 54	3ST/4	
DIFFADR(1:0)	46, 42	3ST/4	
DIFFSTRB#	50	3ST/4	3
LED#	142	3ST/24	
STPWCTL	141	3ST/4	
WIDEPS#	99	I	3

[1] Pad output floats and input is disabled for 8-bit SCSI transfers.
[2] Pad supports active negation.
[3] Pad contains an internal pull-up.

Memory Port Interface

Table 2-4. Memory Port Interface

Signal Name	Pin Number	I/O Driver	Notes
EXTARBACK#	21	I	1
EXTARBREQ#	20	3ST/8	
MA[14:0]	111, 114, 108, 123, 124, 122, 121, 115, 113, 112, 109, 107, 106, 105, 103	3ST/8	
MD[7:0]	126-128, 130, 132-135	I, 3ST/8	1, 2
MDP	137	I, 3ST/8	2, 3, 4
MRW	119	3ST/24	
RAMCS#	117	3ST/24	5
ROMCS#	120	3ST/4	5
SEECS	139	3ST/4	5

[1] Pad contains an internal pull-up. Pad must be grounded for stand-alone mode.

[2] Pad for stand-alone mode when in idle state is driven for self termination. Note, for BRDCTL/SEECTL register use, or when no ROM/EEPROM and for multiuser mode, external pull-up resistors are required to prevent float condition on inputs.

[3] Pad(s) is/are multifunction.

[4] Pad always requires a pull-up resistor when 9-bit SRAM is not used.

[5] External resistors required to prevent external logic float condition.

Other Pins

Table 2-5. Other Pins

Signal Name	Pin Number	I/O Driver	Notes
CLKIN	101	I	
IDDAT	96	I	1
RAMPS#	98	I, 3ST/4	2, 3

[1] Pad contains an internal pull-up

[2] Pad for stand-alone mode when in idle state is driven for self termination. Note, for BRDCTL/SEECTL register use, or when no ROM/EEPROM and for multiuser mode, external pull-up resistors are required to prevent float condition on inputs.

[3] Pad always requires a pull-up resistor when 9-bit SRAM is not used.

Power Distribution Pins

Table 2-6. Power Distribution Pins

Signal Name	Pin Number	I/O Driver	Notes
CVCC(3:0)	138, 100, 60, 24		
CGND(3:0)	140, 102, 58, 22		
SVCC(2:0)	10, 155, 147		
SGND(6:0)	17, 12, 6, 2, 157, 151, 143		
PVCC(3:0)	86, 73, 51, 35		
PGND(14:0)	136, 131, 125, 118, 110, 97, 91, 84, 78, 71, 62, 55, 47, 39, 32		
MVCC(2:0)	129, 116, 104		

CVCC = Core Logic Power Pin
CGND = Core Logic Ground Pin
SVCC = SCSI Pad Ring Power Pin
SGND = SCSI Pad Ring Ground Pin
PVCC = PCI Pad Ring Power Pin
PGND = PCI Pad Ring Ground Pin
MVCC = Memory Port Pad Ring Power Pin
Note: The different ()VCC and ()GND pin strings are not internally connected

Pin Signal Description

The logical state of a signal name that does not end in a # symbol is asserted or active when high and is deasserted or inactive when low. The logical state of a signal name that ends in a # symbol is asserted or active when low and is deasserted or inactive when high.

Supported PCI-32 Pin Signals

Symbol: AD[31:00]
Type: in-t/s

Address and data are multiplexed on the same PCI bus pin. During the first clock of a transaction AD[31:00] contain a physical byte address of (32-bits) called low address 31:00 for Single Address Cycles (SAC). During subsequent clocks, AD[31:00] contain data of (32-bits) called low data 31:00, except for Dual Address Cycles (DAC) where both the first (low address 31:00) and second (high address 63:32) clocks of a transaction contain address and the remaining clocks contain data (low data 31:00). The turn-around PCLK period for AD[31:00] is the idle cycle between transactions.

A PCI read or write bus transaction consist of one Address phase (SAC) or two Address phases (DAC) followed by one or more Data phases. Each PCI Data phase may consist of one or more PCLK periods. Little-endian byte ordering is used. AD[07:00] define the least significant byte and AD[31:24] the most significant byte. All 32 AD[31:00] bits must be driven to stable values (excluding turn around PCLK periods) during every Address and Data phase, to enable even-parity checking. All AD[31:00] bits must be decoded for memory and I/O phases to allow for future address expansion.

The use of AD[01:00] varies in the Address phase of the three different PCI address spaces:

- In the PCI Configuration address space, AD[01:00] are used to identify the type of configuration space the access is intended for. AD[01:00] are a value of 0h to identify the configuration space as type 0 and a value of 1h for type 1, with values of 2h and 3h reserved. Type 0 configuration accesses are not propagated beyond the local PCI bus and must be claimed by a local device or terminated with master-abort. Type 1 configuration accesses are for targets that do not reside on the local PCI bus. For type 0, AD[07:02] define a 32-bit register address within the configuration address space. Thus, configuration address space defaults to Double Word (DWD) addressing aligned to the DWD boundary. Targets with multiple functions must contain a configuration space for each function. The value supplied on AD[10:08] is used to point to each space. The AIC-7870, as a single function target supporting type 0 address space, accesses with a single configuration space. The AIC-7870 as a target, uses positive address decoding over AD[07:02] along with CBE[3:0]# (command is CRDC or CWRC), IDSEL, AD[01:00] = 0h and FRAME# to validate the configuration register address decode, then asserts DEVSEL# to claim the transaction. IDSEL is normally connected to an ADn signal in the range of AD[31:11] of the PCI bus.

AIC-7870 Configuration Space Address Format

Reserved for IDSEL (single-bit)	Function Number	Register Number	Configuration Space Type
AD[31:11]	AD[10:08]	AD[07:02]	AD[01:00]

- In the IO address space, all 32 AD lines are used to provide for direct byte address decoding. The AIC-7870 as a target uses positive address decoding over BASEADR0 register (stored value), AD[31:08] (for mapping), CBE[3:0]# (for command), AD[07:00] (for register address) and FRAME# to validate the Device Space register 256 decodes.

  When the AIC-7870 as a target is enabled to allow access to its Device Space registers from the PCI IO address space, the use of AD[01:00] during the Address phase allows the AIC-7870 to validate the register address decode and claim the transaction (assert DEVSEL# = medium speed). TRDY# is deasserted for the first Data phase to allow the Data phase CBE value to become valid for byte steering to the internal 8-bit register path for a three PCLK data transfer cycle for register write. For register read, TRDY# is deasserted for the required AD[31:00] turn-around cycle, plus an additional PCLK for H/W registers, plus one or more to enable internal or external RAM data to become valid. Note, additional PCLKs are required for some addresses due to mode conditions.

- In the PCI memory address space, AD[01:00] are excluded from the address decode and as such, the address defaults to Double Word (DWD) addressing aligned to the DWD boundary. The value AD[01:00] are used in the memory address space to indicate different Memory Address Transfer modes. A value of 0h indicates linear address increment mode, a value of 1h indicates Address Cache Line Toggle mode, and the values of 2h and 3h are reserved. The AIC-7870 as a master or target only supports the linear address increment mode.

  When the AIC-7870 as a target is enabled to allow access to its Device Space registers from the PCI memory address space, it will use positive address decoding over BASEADR1 register (stored value), AD[31:02], CBE[3:0]# (command) and FRAME# to obtain the DWD access decode and claim the transaction by asserting (DEVSEL# = medium speed). Then use the CBE[3:0]# (data) value to complete the Device Space register decode. TRDY# is deasserted for the first Data phase to allow the Data phase CBE value to become valid for byte steering to the internal 8-bit register path for a three PCLK data transfer cycle for register write. For register read TRDY# is deasserted for the required AD[31:00] turn-around cycle, plus an additional PCLK for H/W registers, plus one or more to enable internal or external RAM data to become valid. Note, additional PCLKs are required for some addresses due to mode conditions.

  When the AIC-7870 as a target is enabled to allow access to the expansion ROM address space through the AIC-7870 memory port to a local 8-bit ROM/EEPROM, the AIC-7870 will use positive address decoding over EXROMCTL register (stored value), AD[31:02], CBE[3:0]# (command) and FRAME# to obtain the DWD access decode and claim the transaction by asserting (DEVSEL# = medium speed).

  For Memory Read commands, the AIC-7870 will perform a burst of four read accesses of the external ROM/EEPROM with MDP (MA15 equal to AD15, MA[14:2] equal to AD[14:02] and MA[1:0]) incrementing with values of 0, 1, 2, 3 to assemble a 32-bit value for the PCI Read command, regardless of the CBE[3:0]# value for the requested data. TRDY# is deasserted for additional PCLKs for the required AD[31:00] turn-around cycle and to enable the external ROM (150 nsec per cycle) data to become valid, and will be asserted when the 32-bits are valid.

  For Memory Write commands, the AIC-7870 will write only the byte associated with the single asserted CBE[3:0]# to the external EEPROM. When no CBE[3:0]#

are asserted, the write will be treated as an NOP. TRDY# will be deasserted for the external EEPROM write cycle access time (150 nsec) then will be asserted to terminate the PCI write access.

Note: The ROM/EEPROM type and its connection to the AIC-7870 should be such that inadvertent writing when the type is ROM will not allow contention of the memory data bus.

For both Read or Write commands, access will be extended with TRDY# deasserted when an Automatic Access Pause (see HCNTRL register) is required, and/or until EXTARBACK# becomes active.

When a PCI burst is indicated it will be disconnected after the first Data phase.

The use of AD[31:00] varies in the Data phase of transactions as follows when the AIC-7870 is a bus master or a bus target:

- Bus target Data phase transactions to the AIC-7870's 8-bit Device Space registers will use the AD[31:00] byte indicated by a single asserted CBE# for all registers. Should more than one CBE# be asserted (indicates a nonsupported data width), the AIC-7870 will indicate Target Abort. When more than one Data phase is indicated (burst operation), the AIC-7870 will indicate Disconnect and only accept the first Data phase (except for SCB double word write, where bursting is allowed, with linear burst order only). When no CBE bits are asserted, the AIC-7870 will not store the associated data for (write) and will supply all AD[31:00] bytes with 0h value for (read).
- Bus target Data phase transactions to the AIC-7870's configuration space supports up to (32-bit) data transfers on AD[31:00] with the valid data bytes indicated with the CBE[3:0]# value for (write), for (read) the AIC-7870 will always source all bytes of the addressed register. Reading reserved configuration space register bytes/bits always return zero for the value. Data written to reserved configuration space register bits or bytes is discarded No error indication is made for reading or writing to reserved registers. When more than one Data phase is indicated (burst operation), the AIC-7870 will indicate Disconnect and only accept the first Data phase.
- The AIC-7870 as a master will always transfer leading offset data bytes, if they exist, to reach the next DWD boundary in the first Data phase of a transaction, providing the byte count is sufficient. Then four bytes will be transferred at a time from DWD boundary to DWD boundary until the last Data phase, which will transfer any trailing offset bytes that may exist, to expire the byte count.

Symbol: CBE[3:0]#
Type: in-t/s

Bus Command and Byte Enables are multiplexed on the same PCI pins. During the Address phase of a transaction, CBE[3:0]# contain a Bus command that defines the function to be performed during the transaction. CBE[3:0]# command encodings are viewed on the bus where a 1 indicates a high voltage and 0 is a low voltage. During the Data phase of a transaction, CBE[3:0]# define which data bytes of AD[31:00] contain valid data. CBE0# applies to AD[07:00] and CBE3# to AD[31:24]. The AIC-7870 asserts CBE[3:0]# when in master mode to indicate the location of the first byte in a 32-bit boundary space and to match the data width being transferred. CBE[3:0]# are asserted (=0) by the system board or a bus master to access the AIC-7870 as a bus slave. No more than one CBE# may be asserted at a time for transactions to the AIC-7870's 8-bit Device registers without causing a Target Abort reply (except for SCB double word writes where bursting is allowed).

> **Note:** Data phases that do not have at least one asserted CBEn# do not transfer data, however all data bytes must be stable so that parity may be developed, and appear as NOPs on the bus.

The turn-around PCLK period for CBE[3:0]# is the idle cycle between transactions. The commands assigned to CBE[3:0]# in the PCI specification are as follows:

Address Phase

CBE[3:]#	Command Type		AIC-7870 Support Target	Master
0000	IAC	Interrupt Acknowledge	No	No
0001	SSC	Special Cycle	No	No
0010	IORDC	IO Read	Yes	No
0011	IOWRC	IO Write	Yes	No
0100	RSVD		No	No
0101	RSVD		No	No
0110	MRDC	Memory Read	Yes	Yes
0111	MWRC	Memory Write	Yes	Yes
1000	RSVD		No	No
1001	RSVD		No	No
1010	CRDC	Configuration Read	Yes	No
1011	CWRC	Configuration Write	Yes	No
1100	MRDMC	Memory Read Multiple	**	Yes
1101	DAC	Dual Address Cycle	No	Yes
1110	MRDLC	Memory Read Line	**	Yes
1111	MWRIC	Memory Write and Invalidate	*	Yes

* Defaults to Memory Write
** Defaults to Memory Read

The AIC-7870 CBE[3:0]# values used/accepted during a Data phase to indicate the valid data bytes are as follows:

Data Phase

CBE[3:]#	AIC-7870 Support Target	Master
1110	acd	m
1101	acd	m
1011	acd	m
0111	acd	m
1100	ac	m
1001	ac	m
0011	ac	m
1000	ac	m
0001	ac	m
0000	abc	m
1111	n	-

a = External ROM/EEPROM read access
b = SCB doubleword write access
c = Configuration register 32-bit registers access
d = Device space 8-bit register or external EEPROM write access
m = System memory access (AIC-7870 outputs only listed CBE[3:0] values)
n = No data transfers (Note all AD[31:00] are driven for parity checking) with normal data phase timing

Symbol: DEVSEL#
Type: in-s/t/s

Device Select#. When asserted, indicates the driving device has decoded its address as the selected target of the current bus transaction. DEVSEL# once asserted cannot be deasserted until FRAME# is sampled deasserted, except for the target-abort case. Also DEVSEL# must be asserted for one or more PCLKs before a target-abort condition may be signaled. The AIC-7870 as a slave asserts DEVSEL# with medium speed timing when responding as a result of a valid and supported command directed to the AIC-7870's Configuration register space, and when enabled, to the Device register space, or to the external ROM. The AIC-7870 as a master, samples DEVSEL# when initiating a transaction to a selected target to determine if the target is capable of proceeding with the current transaction. In the case when DEVSEL# is not asserted by the selected target for six PCLKs (SAC) or seven PCLKs (DAC) after FRAME# is asserted, the AIC-7870 will perform a master-abort (on PCLK seven (SAC) or eight (DAC) deassert FRAME# on the next PCLK (if still asserted) and on the next PCLK deassert IRDY#). Alternately where FRAME# was deasserted after one PCLK (indicates only one Data phase in the transaction), the AIC-7870 will perform a master-abort (on PCLK seven (SAC) or eight (DAC) deassert IRDY#). The AIC-7870 will not retry transactions that resulted in a master-abort (no response from target) and will generate an interrupt to the driver with RMA status active. Intervention is required for the AIC-7870 to continue with bus master transactions. The turn-around PCLK period for DEVSEL# is the Address phase of a transaction. The AIC-7870 as a master never asserts DEVSEL#.

Symbol: FRAME#
Type: in-s/t/s

Frame#. Asserted by the current master to indicate the duration of a bus transaction. The assertion of FRAME# identifies an Address phase of a transaction. The deassertion of FRAME# identifies the final data phase of the transaction (FRAME# cannot be deasserted while IRDY# is deasserted for the final data phase). An idle cycle (PCI bus free) occurs when both FRAME# and IRDY# are deasserted. The turn-around PCLK period for FRAME# is the idle cycle between transactions. See DEVSEL# for the AIC-7870 Master-Abort conditions. The AIC-7870 as a target never asserts FRAME#.

The AIC-7870 asserts FRAME# with the same PCLK that asserts an address value on AD[31:00] for SAC or DAC. When the transaction is a DAC, FRAME# remains asserted for the second PCLK Address phase and for all Data phases that follow, until the last data phase where FRAME# will be deasserted.

Symbol: GNT#
Type: in

Grant#. Asserted indicates to a master that a bus transaction may be performed. This is a point-to-point signal with every master having its own GNT# signal. Only one GNT# may be asserted by the PCI System Board Central Resource Arbitrator at a time. The Arbitrator may deassert GNT# at any time (one PCLK period minimum assertion) and may also assert GNT# when the master is not asserting REQ# (park the bus) to require the master to drive bus signals AD[31:00], CBE[3:0]# (and PAR delayed by one PCLK) within eight PCLK (recommended value is by two to three PCLKs) to prevent bus float conditions. In the case where GNT# is deasserted and FRAME# is asserted on the same PCLK, the bus transaction is valid and will continue. One GNT# can be deasserted coincident with another GNT# being asserted if the bus is not idle. Otherwise a one PCLK delay is required between the deassertion of one GNT# and the assertion of the next GNT#. The newly granted master may not start a transaction until an IDLE cycle (FRAME# and IRDY# deasserted) is sampled. GNT# is held in a tri-stated condition while RST# is asserted. The AIC-7870 extends an asserted RST# internally and thus will not recognize GNT# in any state until after the extension expires. Except for the RST# condition, the AIC-7870 will drive AD[31:00] and CBE[3:0]# on the first PCLK, GNT# is sampled asserted with the PCI bus idle, then PAR one PCLK later. The AIC-7870 will also drive and assert FRAME# if PREQ# is asserted.

Symbol: IDSEL
Type: in

Initialization Device Select. Used in lieu of the upper 24 ADn address signals and is valid only during configuration read and write transactions and is validated with FRAME# assertion and valid CBEn# values. IDSEL is a point-to-point signal with each agent having its own IDSEL. PCI convention is to connect a different AD[31:11] line to IDSEL input of each device on the bus, the AIC-7870 will respond to all accesses in its configuration address range.

Symbol: IRDY#
Type: in-s/t/s

Initiator Ready#. Asserted to indicate the current master's ability to complete the current Data phase of a transaction. During a write, IRDY# indicates that the master is asserting valid data on AD[31:00]. During a read, it indicates the master is prepared to accept data on AD[31:00]. It is used in conjunction with TRDY#. Wait cycles are inserted until both IRDY# and TRDY# are asserted together. A Data phase is completed on any PCLK when IRDY# and TRDY# are both sampled asserted. An idle cycle (PCI bus free) occurs when both FRAME# and IRDY# are deasserted. The turn-around PCLK period for IRDY# is the Address phase of a transaction.

The AIC-7870 as a master asserts IRDY# with the same PCLK that starts a Data phase on AD[31:00].

Symbol: PAR
Type: in-t/s

Parity. The even-parity bit that protects both AD[31:00] and C/BE[3:0]# signals. PAR is generated by the agent that is sourcing the 32-bit address of the transaction and/or the data of the transaction and includes the CBE[3:0] values even if not sourcing them. The state of PAR is valid for the value on AD[31:00] and CBE[3:0] during the previous PCLK period for address and for the PCLK that transferred data, excluding PAR turn-around PCLK periods which occur in the PCLK following the turn-around PCLK period of AD[31:00]. The agent detecting parity errors will set the DPE bit in the Configuration Status register without regard for the state of PERRESPEN bit. The AIC-7870 as a target indicates SERR for address parity error detection when both PERRESPEN and SERRESPEN are active. The AIC-7870 as a target will assert PERR for data parity errors when PERRESPEN is active.

Symbol: PCLK
Type: cin

PCI Bus Clock Input. Provides timing for all transactions on the PCI bus. All other PCI signals are sampled on the rising edge of PCLK, and all parameters are defined with respect to this edge. PCLK is a controlled skew, point-to-point signal to each agent and is only driven by the PCI System Board Central Resource. The PCLK signal for the AIC-7870 is a maximum rate of 33.3 MHz. When POWRDN is active, the AIC-7870 restricts the use of PCLK to only Configuration address space and the AIC-7870 Device registers which are accessible only from the PCI bus (not from the internal sequencer). PCLK signal used by other logic is maintained in the active state (=1) when POWRDN is active. Note, all internal logic is static allowing the PCLK and the CLKIN signals to be stopped externally if desired when no active SCSI commands are in process for maximum power down mode. See RAMPS# for a buffered version of PCLK for external adapter board use.

Symbol: PREQ#
Type: t/s

PCI Request#. When asserted, indicates to the PCI System Board Arbitrator that a master desires use of the bus. This is a point-to-point signal with every master having its own PREQ#. Arbitration for the PCI bus is performed either when the bus is idle or in parallel with the transaction in process. When only a single Data phase is to be performed, PREQ# should be deasserted with the same PCLK that asserts FRAME#. When a

transaction is terminated by a target, the master must deassert its PREQ# for a minimum of two PCLK periods (one period must include the bus idle period). This allows another agent to use the bus while the previous target (that requested the STOP) prepares to continue.

Note: This is not required where the master deasserted FRAME# indicating the last Data phase of a transaction is in process. In this case, provided GNT# is still asserted, the master could start another transaction without deasserting PREQ#.

PREQ# is asserted by the AIC-7870 to become a bus master provided that the MASTEREN bit is active and either ISPACEEN or MSPACEEN bits (to allow access to the AIC-7870's Device registers) is/are active in the Configuration Command register for the following conditions:

- For system memory to data FIFO transfers (HDMAEN and DIRECTION are active and CACHETHEN inactive), when DFTHRSH or FIFOEMP becomes active. In this case, PREQ# remains asserted until DFSDH or FIFOFULL becomes active or STOP# is asserted, as long as HDMAEN is active, HCNT is not zero, and GNT# is asserted (or, if deasserted, until the latency timer has expired for pre-empt conditions and if performing a cache line referenced command MRDMC, until the in-process cache line transfer has been completed with MRDCEN not active. When MRDCEN is active, the in-process cache line transfer will be truncated).
- For system memory to data FIFO transfers (HDMAEN and DIRECTION are active and CACHETHEN inactive), when DFCACHETH becomes active. PREQ# will be asserted until the cache line is completed and DFCACHETH is inactive or (STOP# is asserted), as long as HDMAEN is active, HCNT is not zero, and GNT# is asserted (or, if deasserted, until the latency timer has expired for preempt conditions and if performing a Cache Line Referenced command MRDLC until the in-process cache line transfer has been completed with MRDCEN not active. When MRDCEN is active, the in-process cache line transfer will be truncated).
- For data FIFO to system memory transfers (HDMAEN active and DIRECTION and CACHETHEN inactive), when DFTHRSH, DFSXDONE, or FIFOFULL becomes active. In this case, PREQ# remains asserted until FIFOEMP is active or STOP# is asserted, as long as HDMAEN is active, HCNT is not zero, and GNT# is asserted (or, if not asserted, until the latency timer has expired for pre-empt conditions and if performing a Cache Line Referenced command or (MWRIC), until the cache line transfer has been completed).
- For data FIFO to system memory transfers (HDMAEN and CACHETHEN active and DIRECTION inactive), when DFCACHETH becomes active. In this case, PREQ# remains asserted until the cache line is completed (or multiple cache lines are completed if DFCACHETH remains active), or STOP# is asserted, as long as HDMAEN is active, HCNT is not zero, and GNT# is asserted (or, if not asserted, until the latency timer has expired for pre-empt conditions and if performing a Cache Line Referenced command (MWRIC), until the cache line transfer has been completed).

  Note: PREQ# output will be floated whenever RST# is asserted.

Symbol: PERR#
Type: in-s/t/s

Parity Error#. May be asserted (pulsed for one PCLK period for each detected error provided that the Parity Error Response bit, PERRESPEN, is active in the Configuration Command register) only by the agent receiving the data. Also, a target cannot assert PERR# until it has claimed the access by asserting DEVSEL# and completing the data transfer. The turn-around PCLK for PERR# is the third PCLK period after the last address PAR period of an agent. PERR# is asserted for detected errors two PCLK periods after the data that contained the error as per the following sequence:

1. From PCLK N that transfers the data (both TRDY# and IRDY# are asserted) that data parity is to be generated for.
2. From PCLK N+1 PAR is asserted for the data asserted in PCLK N.
3. From PCLK N+2 assert PERR# when an even-parity error was indicated on the PCLK N+1 sample of data.
4. From PCLK (N+2)+1 deassert PERR# and when the last PERR# cycle wait for one more PCLK period before floating PERR#.

The AIC-7870 asserts PERR# only for detected data parity errors for received data when PERRESPEN is active but always sets DPE bit active in the Configuration Status register.

Symbol: RST#
Type: in

Reset#. When asserted forces agents to a known initialization state. RST# may be asynchronous to PCLK when asserted or deasserted. Deassertion is guaranteed to be a clean, bounce-free edge.

- All n/o/d, t/s and s/t/s type signals are forced to a high impedance state.
- All o/d type signals are forced to float.
- All agent internal registers (Device and Configuration) are forced to specified states.
- All internal RAM data values should be considered indeterminate.

> **Note:** The AIC-7870's internal Power-On-Reset (POR) activated by assertion of RST# is extended two CLKIN rising edges to ensure complete internal initialization, should short RST# assertions occur, and to provide an internally synchronized inactivation of the AIC-7870 initialization. POR is also activated by a write to the CHIPRST bit in Device HCNTRL register.

Symbol: SERR#
Type: in-o/d

System Error#. May be asserted by a PCI agent that detects an address parity error (provided that PERRESPEN and SERRESPEN are active) during the Address phase of a transaction or for data parity errors on special cycles and for any other system error where the result will be a catastrophic error. The transaction master is solely responsible for reporting master or target aborts; targets do not assert SERR# when using target-abort termination. SERR# is restored only by a weak pull-up on the system board, and

may take several PCLK periods to recover to a deasserted state. SERR# is asserted (pulsed for one PCLK period) for detected address errors two PCLK periods after the address that contained the error as per the following sequence:

1 From PCLK N for the address that is being asserted, for which parity is to be generated for.

2 From PCLK N+1 PAR is asserted for the address asserted in PCLK N.

3 From PCLK N+2 assert SERR# when an even-parity error was indicated on the PCLK N+1 sample of the address.

4 From PCLK (N+2)+1 deassert SERR#.

The AIC-7870 as a target only asserts SERR# for all detected address parity errors when PERRESPEN and SERRESPEN are active. In all cases, DPE will be set active without regard for the enables.

Symbol: STOP#
Type: in-s/t/s

Stop. When asserted, indicates the current target is requesting the master to stop the current Data phase of a transaction in process. STOP#, once asserted, must remain asserted until FRAME# is deasserted and data may or may not be transferred in the final Data phase of the transaction. The turn-around PCLK period for STOP# is the Address phase of a transaction.

The AIC-7870, when attempting to perform a transaction to a target that responds with Target-Stop (disconnect) or Target-Retry, will retry the transaction with the next address to be transferred. When the response is Target-Abort, the AIC-7870 will not retry and will set the RTA bit in the Configuration Status register active and generate an interrupt IRQA#. See the Device CLRINT register for clearing this interrupt.

> **Note:** For Target-Abort this means the SCSI data segment transfer will stall and software/firmware intervention is required.)

The AIC-7870 as a target asserts STOP# (disconnect with data transferred) when FRAME# is indicating burst cycles (except for SCB double word write transactions where bursting is allowed). This is not an error condition.

Typical AIC-7870 PCI Data Transfer Status

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Normal Master Ending
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts SAC
Y	Y	N	N	N	
Y	Y	Y	N	N	Target responds
Y	Y	Y	Y	N	Data transferred
Y	Y	Y	Y	N	Data transferred
N	Y	Y	Y	N	Last data transferred
N	N	N	N	N	Bus idle

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Normal Master Ending
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts DAC
Y	N	N	N	N	
Y	Y	N	N	N	
Y	Y	Y	N	N	Target responds
Y	Y	Y	Y	N	Data transferred
Y	Y	Y	Y	N	Data transferred
N	Y	Y	Y	N	Last data transferred
N	N	N	N	N	Bus idle

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Target Disconnected with Data
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts SAC
Y	Y	N	N	N	
Y	Y	Y	N	N	Target responds
Y	Y	Y	Y	N	Data transferred
Y	Y	Y	Y	Y	Last data transferred
N	Y	N	N	Y	No data transferred
N	N	N	N	N	Bus idle

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Target Disconnected without Data
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts SAC
Y	Y	N	N	N	
Y	Y	Y	N	N	Target responds
Y	Y	Y	Y	N	Data transferred
Y	Y	Y	N	Y	No data transferred
N	Y	N	N	Y	
N	N	N	N	N	Bus idle

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Target Retry
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts SAC
Y	Y	N	N	N	
Y	Y	Y	N	Y	Target responds
N	Y	Y	N	Y	No data transferred
N	N	N	N	N	Bus idle

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Target Abort
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts SAC
Y	Y	N	N	N	
Y	Y	Y	N	N	Target responds
Y	Y	N	N	Y	No data transferred
N	Y	N	N	Y	No data transferred
N	N	N	N	N	Bus idle

FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#	Condition Master Abort
N	N	N	N	N	Bus idle
Y	N	N	N	N	Master starts SAC
Y	Y	N	N	N	
Y	Y	N	N	N	
Y	Y	N	N	N	
Y	Y	N	N	N	
Y	Y	N	N	N	
Y	Y	N	N	N	
Y	Y	N	N	N	Master declares abort
N	Y	N	N	N	FRAME/IRDY ending sequence
N	N	N	N	N	Bus idle

Symbol: TRDY#
Type: in-s/t/s

Target Ready#. Asserted to indicate the current slave's ability to complete the current Data phase of a transaction. During a read, TRDY# indicates that the slave is asserting valid data on AD[31:00]. During a write it indicates the slave is prepared to accept data. It is used in conjunction with IRDY#. Wait cycles are inserted until both IRDY# and TRDY# are asserted together. Wait cycles should be minimized, when more than eight are expected, except for the first transfer, the transaction should be disconnected by the target and retried by the master. A Data phase is completed on any PCLK when IRDY# and TRDY# are both sampled asserted. The turn-around PCLK period for TRDY# is the Address phase of a transaction.

The AIC-7870 as a target will always have TRDY# deasserted in the first PCLK period of the first Data phase so that the asserted CBE# value may be decoded for internal byte steering and for read access required AD[31:00] turn-around. TRDY# will also be deasserted for additional PCLKs for RAM data to become valid, and when PAUSEACK is not active and the access is to a register that requires it to be active for the host driver to access the register (see PAUSE[ACK] bit in the Device HCNTRL register), for SCB access (when RAMPSM is asserted), or for ROM/EEPROM access (see EXTARBREQ#/EXTARBACK#). When TRDY# is not asserted within eight PCLKs for Data phases, the AIC-7870 will respond with Disconnect when the data transfers. Additionally, should TRDY# not be asserted before the Exceed Timer expires, the AIC-7870 will respond with Retry. The Exceed Timer is an escape timer that should only time out should a requested AIC-7870 address/register not be available for long periods of time (not a normal condition), and is provided to prevent hanging the system. Also see EXTARBACK#.

Non-supported PCI-32 Pin Signals

IRQA#	SDONE#
IRQB#	TCK
IRQC#	TDI
IRQD#	TDO
LOCK#	TMS
SBO#	TRST#

Non-supported PCI-64 Extension Signals

ACK64#	PAR64
BE#[7:4]	REQ64#
D[63:32]	

Supported PCI Pin Side Band Signal

Table 2-7. Supported PCI Pin Side Band Signals

Symbol	Type	Definition
IRQA#	o/d	**Interrupt Request A.** IRQA# assertion state changes are synchronized to PCLK for PCI type errors and parity errors. The AIC-7870 interrupt conditions cannot assert IRQA# when the INTEN bit is not active or the POWRDN bit is active in the HCNTRL register. For IRQA# assertion conditions see the INTSTAT register. Note IRQA# output is floated when RST# is asserted and also used in test modes.

External PCI-32 Pin Signal Type Definition

Table 2-8. External PCI-32 Pin Signal Type Definition

Type	Definition
cin	**Clock Input** is a special clock high drive input-only receive signal.
in	**Input** is a standard input-only receive signal.
t/s	**Three-state** is an output signal with control enabled output driver.
in-t/s	**In-three-state** is a bidirectional signal, with control enabled output driver and an internally connected standard input.
in-s/t/s	**Sustained Three-state** is a bidirectional signal, active low three-state signal and driven by one, and only one, agent at a time. The agent that drives an in-s/t/s signal low must drive it high for at least one PCLK before letting it float. A new agent cannot start driving an in-s/t/s signal any sooner than one clock after the previous owner floats it. A pull-up sustains the inactive state until another agent drives it and is provided by the PCI bus central resource.
o/d	**Open Drain** is an open drain output. The system board provides a light pull-up for o/d signals. Multiple devices share the signal as a wire-OR. The signal is asserted synchronous with PCLK for one PCLK period. The signal deassertion period is determined by the RC time period provided by the pull-up.

SCSI Interface Pin Signals

Table 2-9. SCSI Interface Pin Signals

Symbol	Type	Definition
SCD[15:0]#	in-n/o/d	**SCSI Data [15:0] #.** The SCSI data lines drive the ID during arbitration and selection, and command and data information as well as status and messages. SCD[7:0] are used for 8-bit data transfers while SCD[15:8] are floated with their inputs disabled.
SCDPH#	in-n/o/d	**SCSI High Byte Parity #.** This bit provides odd parity for SCD[15:8]#. SCDPH# is floated for 8-bit data transfers with its input disabled.
SCDPL#	in-n/o/d	**SCSI Low Byte Parity #.** This bit provides odd parity for SCD[7:0]#.
CD#	in-n/o/d	**Command/Data #.** This control line is received when in Initiator mode or driven when in Target mode. It indicates Command or Message phase when asserted, and Data phase when deasserted. This control signal is used for 8 or 16-bit transfers.
IO#	in-n/o/d	**In/Out #.** This control line is received when in Initiator mode or driven when in Target mode. It indicates the In direction when asserted, and the Out direction when deasserted. This control signal is used for 8 or 16-bit transfers.
MSG#	in-n/o/d	**Message #.** This control line is received when in Initiator mode or driven when in Target mode. It indicates a Message phase when asserted, and a Command or Data phase when deasserted. This control signal is used for 8 or 16-bit transfers.
REQ#	in-n/o/d	**Request #.** This control line is received by the device when in Initiator mode and driven when in Target mode. A Target will assert REQ# to indicate a byte is ready or is needed by the Target. This control signal is used for 8 or 16-bit transfers.
ACK#	in-n/o/d	**Acknowledge #.** This control line is received by the device when in Target mode and driven when in Initiator mode. An Initiator will assert ACK# to indicate a byte is ready for or was received from the Target. This control signal is used for 8 or 16-bit transfers.
RESET#	in-o/d	**Reset #.** This line is received and/or driven. It is interpreted as a hard reset and will clear all commands pending on the SCSI bus. This control signal is used for 8 or 16-bit transfers.
SEL#	in-o/d	**Select #.** This line is driven after a successful arbitration to Select as an Initiator or Reselect as a Target, and otherwise it is received. This control signal is used for 8 or 16-bit transfers.
BSY#	in-o/d	**Busy #.** This line is driven by the Initiator as a handshake during arbitration, and received for the rest of the transfer. As a Target, it is driven also as a handshake during Arbitration, and then is driven for the rest of the transfer. This control signal is used for 8 or 16-bit transfers.
ATN#	in-n/o/d	**Attention #.** This line is driven as an Initiator when a special condition occurs. It is received by the Target. This control signal is also used for 16-bit transfers.
DIFFDAT[3:0]	t/s	**Differential Control Data.** These lines contain information which is latched by outside circuitry to provide control of differential drivers. The definition of these bits depends on the value of DIFFADR[1:0]. This control signal is used for 8 or 16-bit transfers. Note the DIFFDAT[3:0] outputs are floated when RST# is asserted.
DIFFADR[1:0]	t/s	**Differential Control Address.** These lines determine the meaning of DIFFDAT[3:0]. This control signal is used for 8 or 16-bit transfers. Note the DIFFADR[1:0] outputs are floated when RST# is asserted.
DIFFSTRB#	in-t/s	**Differential Control Strobe #.** This signal clocks the data present on DIFFDAT[3:0] into the addressed latch specified by DIFFADR[1:0] on its rising edge. Both DIFFDAT[3:0] and DIFFADR[1:0] are stable for the duration of DIFFSTRB#. This control signal is used for 8 or 16 -bit transfers. Note DIFFSTRB# output is floated when RST# is asserted and its internal pull-up will keep DIFFSTRB# at a high level. DIFFSTRB# is also used for setting TESTBITEN status, used during component testing.

Table 2-9. SCSI Interface Pin Signals *(Continued)*

Symbol	Type	Definition
LED#	t/s	**LED#**. Output provides three functions: 1 To indicate (when asserted =0) that the AIC-7870 is actively connected to the SCSI bus. LED asserted state is latched with the ORed result of active bits SELINGO, SELDI and SELDO in the SSTAT0 register and deasserted by the following SCSI bus free condition. LED# may be used to provide system status of the AIC-7870 SCSI bus activity and may directly drive an indicator (LED) providing the current is limited to a maximum of 20 mA. 2 As a clock to (20 MHz) shift-in an external device ID value from input IDDAT, to replace the internal default Device ID value. This use of LED# is triggered as a result of RST# assertion. See IDDAT pin for details. 3 For diagnostic support or general purpose output control bit, see Device SBLKCTL register. Note the LED# output is floated when RST# is asserted. After RST# is deasserted the specified IDDAT clocking occurs, then LED# will be asserted continuously until cleared by software/firmware, to provide an indication that normal run mode has been entered. Should a PCI target access be attempted to the AIC-7870 while the IDDAT shift-in process is active, a RETRY response will be returned. Note the fourth IDDAT clock present on LED# internally samples the input state of RAMPS# to store the MPORTMODE state.
STPWCTL	t/s	**SCSI Termination Power Down Control**. Provides the capability to enable or disable the external SCSI bus termination power source. The enable/disable polarity of STPWCTL may be selected with the STPWLEVEL bit in the Configuration DEVCONFIG register and the actual enable/disable state is selected with the STPWEN bit in the Device SXFRCTL1 register. CHIPRST forces STPWCTL to the selected disabled state and STPWEN to the inactive state. While RST# assertion forces STPWCTL to be floated, and both STPWLEVEL and STPWEN to be inactive. STPWCTL may also be used for a general purpose output control bit.
WIDEPS#	input	**Wide Present#**. When asserted (=0) indicates that a wide (16-bit) cable connector is present. WIDEPS# input contains an internal pull-up and only needs to be connected for 16-bit operation. See the AIC-7870 Device SBLKCTL register.

SCSI Pin Signal Type Definition

Table 2-10. SCSI Pin Signal Type Definition

Type	Definition
inpu	**Input-pull-up** is a standard input-only receive signal with an internal pull-up.
t/s	**Three-state** is an output signal with control enabled output driver.
inpu-s/t	**Input-pull-up-three-state** is a bidirectional signal, with control enabled output driver and an internally connected input with an internal pull-up.
in-n/o/d	**Active Negation** capable **Sustained Open Drain** is a *hybrid* of a three-state and open drain, with internally connected input receiver (with hysteresis and input disable). External logic normally provides pull-up for n/o/d signals and this pull-up may be enhanced by use of the Active Negation function. Multiple devices share the signal as a wire-OR. The output will only provide Active Negation function during SCSI Synchronous Data transfer. Also see DIFACTNEGEN in the Configuration DEVCONFIG register.
in-o/d	**Sustained Open Drain** is an open drain output, with internally connected input receiver (with hysteresis and input disable). External logic provides pull-up for o/d signals. Multiple devices share the signal as a wire-OR.

Memory Port Pin Signals

Symbol: EXTARBACK#
Type: inpu

External Arbitration Acknowledge. A status input to the AIC-7870 with an internal pull-up. When EXTARBACK# is asserted (=0) it indicates to the AIC-7870 that it may drive its memory port outputs and access externally connected devices (SEEPROM, ROM/EEPROM, SRAM, board logic devices). EXTARBACK#, once asserted, is expected to remain asserted until some other user needs access to the memory port external shared devices, which is detected by the external memory port arbitrator (a separate device). EXTARBACK# has two operational modes as follows:

- Single-user memory mode: EXTARBACK# is permanently asserted (external connection). No external arbitrator is required and no arbitration overhead is experienced in this mode. The memory port is continuously driven (RST# is not asserted and CHIPRST is not written =1) unless a read cycle is to be performed. This mode of operation may be determined following RST# assertion by reading the MPORTMODE value (=1) in the DEVCONFIG register. The current value of EXTARBACK# may also be determined by reading SEECTL register bit 7.
- Multiuser mode: Requires an external arbitration controller (a separate component) and termination for the memory port signal lines (MD[7:0], MDP, RAMCS#, and ROMCS# lines require pull-up termination. The SEECS line requires pull-down termination).

  Each device connected to the arbitration controller requires individual point-to-point EXTARBACK# pin. The external arbitration controller and other devices (i.e. multiple AIC-7870s) must use the rising edge of the same 40 MHz clock source, as connected to the AIC-7870's CLKIN pin. This mode of operation may be determined following RST# assertion by reading the MPORTMODE value (=0) in the DEVCONFIG register. The current value of EXTARBACK# may also be determined by reading SEECTL register bit 7. This indicates a multiuser configuration and the memory port is not driven by a user until a request is made with its EXTARBREQ# assertion (first user) and use granted by its EXTARBACK# assertion (first user) as a response from the arbitrator. When the external arbitrator asserts EXTARBACK# (first user) to one of the multiusers it will continue to assert it until another multiuser makes a request by asserting its EXTARBREQ# (second user). When the arbitrator samples a new additional request it will deassert its asserted EXTARBACK# (first user) and wait for the associated EXTARBREQ# (first user) to be deasserted. This indicates that the first (old) user has stopped driving the memory port. The arbitrator now asserts the EXTARBACK# associated with the second (new) user. When the new user samples its EXTARBACK# asserted it will drive the memory port and perform accesses until sampling its EXTARBACK# deasserted.

  This clock interlocked protocol insures the present user's current access is completed before deasserting its EXTARBREQ#, prevents memory port drive contention, and reduces the arbitration overhead for multiple accesses by the same user. Memory port access by a user is expected to be bursty (i.e. loading new SCBs).

Note: When an EXTARBACK# is deasserted it cannot be re-asserted until the associated EXTARBREQ# is deasserted. Minimum arbitration overhead for this mode is two CLKIN periods, plus that portion of the current users cycle when a memory port cycle is in process.

The cycle proceeds as follows:

- The AIC-7870 requires an access of the memory port and asserts EXTARBREQ# with a rising edge of CLKIN.
- The arbitrator samples EXTARBREQ# asserted and when all other EXTARBACK#s are deasserted, asserts the EXTARBACK# for the related EXTARBREQ# with a rising edge of CLKIN.
- When the AIC-7870 samples its EXTARBACK# asserted with a rising edge of CLKIN, the AIC-7870 enables drive to its memory port outputs as required for the current access.
- The AIC-7870 can now perform accesses to the memory port as required until sampling its EXTARBACK# deasserted with a rising edge of CLKIN. When EXTARBACK# is deasserted and a memory port cycle is currently in process, EXTARBREQ# will be deasserted with the same rising edge of CLKIN that completes the cycle. If no cycle is in process, the AIC-7870 will deassert its EXTARBREQ# and stop driving the memory port outputs with the same rising edge of CLKIN that sampled EXTARBACK# deasserted. Note that when EXTREQLCK in the DSCOMMAND register is active, EXTARBREQ# will not deassert until EXTREQLCK is sampled deasserted.

Additionally, for SCB SRAM accesses only, the memory port cycle time may be doubled by setting the EXTSCBTIME bit in DEVCONFIG register, for large multiuser configurations or slower SRAM devices. See SEECTL/BRDCTL registers for SEEPROM/board logic device access procedure, and the EXTROMCTL register for ROM/EEPROM device access procedure.

Multiuser sequencer (SCSI PhaseEngine) accesses (PAUSE inactive) to external SRAM(SCB) /SEEPROM /board logic devices are compatible with the selected sequencer instruction cycle (8 or 10 MHz determined by the FASTMODE bit in SEQCTL register). The sequencer instruction cycle to an SCB address is stretched upon decoding the address until EXTARBACK# is sampled asserted. Then the time lost in the cycle for decoding and arbitration is reinserted in the cycle, so that the active sequencer cycle will contain the original time period as if no arbitration was performed. PAUSE requests that become active during the stretch time will not be acted upon by the sequencer until the stretch is completed. Additionally, should the previous sequencer cycle have been a write to the INTSTAT register that set SEQINT active and the current sequencer cycle is in the SCB address range, it will not cause EXTARBREQ# to be asserted. Sequencer accesses of the external SEEPROM/board logic devices does not utilize cycle stretches nor the EXTSCBTIME bit to double the cycle time. See the SEECTL/BRDCTL registers for SEEPROM/board logic access procedure.

Multiuser host accesses (PAUSEACK is active) to external SRAM(SCB) ROM/EEPROM /SEEPROM /board control devices are the same as for the sequencer multiuser accesses except that the host access request begins by synchronizing the request for arbitration to CLKIN rising edge before asserting EXTARBREQ#. The cycle will continue until the host request becomes inactive and is synchronized as such then EXTARBREQ# deasserted with a CLKIN rising edge.

Note: The time lost in the host cycle for arbitration is reinserted in the cycle, so that the active host cycle will contain the original time period as if no arbitration was performed. The selection of FASTMODE bit in SEQCTL register has no effect on host mode cycle time except for AAP action.

Symbol: EXTARBREQ#
Type: t/s

External Arbitration Request. A status output that when asserted (low) indicates to an external Arbitrator that the AIC-7870 requires access to the memory port external devices (SRAM(SCB) /ROM/EEPROM /SEEPROM /board logic devices). Each device (i.e. the AIC-7870s) desiring access to the shared external devices requires individual point-to-point EXTARBREQ# pin. All sequencer or host requests for memory port access will be delayed (stretched) until access is initially granted by sampling EXTARBACK# asserted. EXTARBREQ# will remain asserted until EXTARBACK# is sampled deasserted (see EXTARBACK#). When EXTREQLCK is set active it will not affect the assertion of EXTARBREQ# (only access requests from the sequencer or host will cause this to occur), but once EXTARBREQ# is asserted, it will extend its assertion until EXTREQLCK is inactive. EXTARBREQ# output is floated during assertion of RST# or CHIPRST written (=1).

Symbol: MA[14:0]
Type: t/s

Memory Address [14:0]. The memory port address bus outputs to the externally connected devices. MA[14:0] outputs are floated during assertion of RST# and become driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion, they will continue to be driven if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted.

Note: MA[7:0] are also used in test modes, see SFUNCT register.

MA[14:0] (along with MDP which is used as MA15) are used to directly address ROM/EEPROM devices up to 64 KBytes for access through the stored expansion ROM address range when the AIC-7870 is a target (see AD[15:02]). MA[12:0] are used to directly address SRAM devices up to 8 KBytes for SCB page expansion access by the sequencer or when the AIC-7870 is a target (see AD[12:0] and SCBPTR register). When SEEMS is active MA[14:0] are not used (see SEECTL and BRDCTL registers).

Symbol: MD[7:0]
Type: in-t/s

Memory Data[7:0]. The memory port data bus is used to read or write external ROM/EEPROM byte data when the AIC-7870 is the PCI target, to read or write external SRAM SCB page byte data by the sequencer or when the AIC-7870 is the PCI target, to read or write external SEEPROM bit data or board control device data by the sequencer or when the AIC-7870 is the PCI target. MD[7:0] outputs are floated during assertion of RST# and become driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion, they will continue to be asserted if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted. Whenever a read cycle is to be performed the MD[7:0] outputs are floated during that cycle.

For SEEPROM/board logic cycles, SEEMS is actively generating a request for arbitration. This request is synchronized to CLKIN rising edge and EXTARBREQ# is asserted. When EXTARBACK# is sampled asserted, MD[2:0] are reconfigured for access and control through the SEECTL register. Normally, MD2 is used for a SEEPROM shift clock, MD1 is used to output serial data bits to the SEEPROM, and MD0 is used to input serial data bits from the SEEPROM (a pull-up resistor on MD0 is required for this usage). See SEECTL register for control and use information. MD[7:3] are reconfigured for access and control through the BRDCTL register to access external board logic devices (pull-up resistors on MD[7:5] are required for this usage). See BRDCTL register for control and use information.

> **Note:** When BRDCTL/SEECTL operations are being performed bidirectional pins may be in the input state for long periods of time and need to be terminated.

Symbol: MDP
Type: in-t/s

Memory Data Parity. Optionally used for parity protection of SCB data stored in the external SRAM (9-bit device). Odd-parity data will always be generated and parity checking is enabled when EXTSCBPEN is active in the Configuration DEVCONFIG register. MDP output is floated during assertion of RST# and becomes driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion, they will continue to be asserted if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted. Whenever a SRAM read cycle is to be performed, the MDP output is floated during that cycle. When the external SRAM is an 8-bit device, EXTSCBPEN must be inactive and a pull-up resistor present to prevent float condition.

> **Note:** When BRDCTL/SEECTL operations are being performed, bidirectional pins may be in the input state for long periods of time and need to be terminated.

When SEEMS is active, MDP is reconfigured for access and control through the BRDCTL register to access external board logic devices. See BRDCTL register for control and use information. A pull-up resistor is required on MDP for this usage.

When ROM/EEPROM accesses are being performed, MDP is reconfigured to become MA15 and will contain the value provided by AD15.

Symbol: MRW
Type: t/s

Memory Port Read Write. Determines whether the memory port cycle is a read or write cycle and is driven when the AIC-7870 has been granted the arbitration for an access to the external SRAM/ROM/EEPROM memory devices. A read cycle will result when output MRW is at a high level (=1) while RAMCS# or ROMCS# is asserted. A write cycle will result when output MRW is at a low level (=0) while RAMCS# or ROMCS# is asserted. A read cycle, a write cycle, or a read-modify-write cycle may be performed by the sequencer with memory port timed control of MRW and RAMCS# in a single sequencer instruction cycle access of the memory port. The AIC-7870 as a target may only perform a read or a write cycle access through the memory port with timing following the source of the access time. MRW output is floated during assertion of RST# and becomes driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion they will continue to be asserted if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted.

Symbol: RAMCS#
Type: t/s

RAM Chip Select#. Driven when the AIC-7870 has been granted the arbitration for an access to the external memory SRAM device and is asserted (=0) for an access to the external SRAM. RAMCS# is in a float condition and becomes driven only after EXTARBACK# is asserted due to the AIC-7870 EXRARBREQ# being asserted or when MPORTMODE is active. RAMCS# output is floated during assertion of RST# and becomes driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion they will continue to be asserted if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted. RAMCS# requires an external pull-up resistor to control the external SRAM chip select when the AIC-7870 output RAMCS# is in a float condition. The SRAM cycle access time is 20 nsec.

Symbol: ROMCS#
Type: t/s

ROM Chip Select#. Driven when the AIC-7870 has been granted the arbitration for an access to the external ROM/EEPROM and is asserted (=0) for access of the external memory ROM/EEPROM device. ROMCS# is in a float condition and becomes driven only after EXTA-RBACK# is asserted due to the AIC-7870 EXTARBREQ# being asserted or when MPORTMODE is active. ROMCS# output is floated during assertion of RST# and become driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion, they will continue to be asserted if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted. The cycle access time is hardware controlled for a 150 nsec device.

> **Note:** The external ROM control interface should be such that when MRW is low (write access), then ROMCS# is asserted, that the ROM's data outputs are not also enabled, then the cycle becomes a NOP with no contention with the AIC-7870's driven MD[7:0] outputs. ROMCS# requires an external pull-up resistor to control the external ROM/EEPROM chip select when the AIC-7870 output ROMCS# is in a float condition.

Symbol: SEECS
Type: t/s

Serial EEPROM Chip Select. Asserted (=1) provided SEEMS is active and the AIC-7870 has been granted the arbitration for an access to the external memory SEEPROM device. SEECS output may now be controlled by the state stored in bit SEECS in the SEECTL register. SEECS output is floated during assertion of RST# and becomes driven only after EXTARBACK# is asserted (with RST# deasserted and CHIPRST not written (=1)). Once they have become driven due to EXTARBACK# assertion they will continue to be asserted if EXTARBACK# becomes deasserted until EXTARBREQ is deasserted. SEECS# requires an external pull-up resistor to control the external SEEPROM chip select when the AIC-7870 output SEECS# is in a float condition. See the SEECTL and BRDCTL registers for additional information.

Memory Port Pin Signal Type Definition

Table 2-11. Memory Port Pin Signal Type Definition

Type	Definition
inpu	**Input** is a standard input-only receive signal with an internal pull-up.
t/s	**Three-state** is an output signal with control enabled output driver.
in-t/s	**In-three-state** is a bidirectional signal, with control enabled output driver and an internally connected standard input. When the output driver is in a floated state the pin is being used as an input, and is waiting to become driven or is not being used. In this case, the pin must be connected to an external device whose output has been enabled or an external pull-up is required.

Other Pin Signals

Table 2-12. Other Pin Signals

Symbol	Type	Definition
CLKIN	in	**Clock Input.** Standard input (AIC-7870) - 40 MHz nominal input frequency. Used internally by the SCSI, sequencer, Data FIFO, memory port and the PCI host blocks for timing.
IDDAT	inpu	**Identification Data.** Provides for shifting-in an external device ID value to replace the internal default device ID value readable from the DEVICEID1 Configuration register or the DSDEVID1 Device register. The external IDDAT data source should be initialized (loaded) with the desired identification data to be shifted-in when RST# is asserted. The loaded data will be shifted-in with the rising edge of 8 IDDAT shift clocks (20 MHz rate) provided on LED# commencing (2 to 3 CLKINs periods) after sampling RST# deasserted. LED# transitions after the 8 IDDAT shift clocks are ignored by the IDDAT data load logic. IDDAT data bits are shifted-in LSB first referenced to bit 0 of DEVICEID1/ DSDEVID1. Shifted-in data value of FFh is reserved and causes the internal default value to be maintained (see DEVICEID1 register). This allows the IDDAT pin (with its internal pull-up) to be left floating (no external logic) to use the internal default value. Additionally, after the 8 IDDAT shift clocks and prior to exchanging the default value for the shifted-in value, tests are performed on the low and high nibbles. When a shifted-in nibble value is Fh, the internal default value for that nibble is maintained. Thus an external 4-bit shift register with its serial input pulled up to a one level, could enable selection of one of fourteen external identification values with only the low nibble exchanged for the default value. An 8-bit shift register is required for both nibbles to be changed. Should a PCI target access be attempted to the AIC-7870 while the IDDAT shift-in process is active, a RETRY response will be returned.
RAMPS#	inpu-s/t	**RAM Present#.** Asserted (=0) to enable access of an external SRAM for expanded SCB Array data storage. When RAMPS# is deasserted, the internal AIC-7870 SCB Array RAM is used for SCB storage of 16 SCBs (0-15), the SCBPTR register maximum stored value is FFh, and the QINFIFO/ QINCNT/ QOUTFIFO/ QOUTCNT storage and count (0 = empty, 16 = full) support only 16 SCBs. When RAMPS# is asserted, an external SRAM is required. The size of the SRAM is 4 KBytes for 128 SCBs and 8 KBytes for 256 SCBs. The software driver must scan to determine the actual installed SRAM size (i.e. 1, 2, 4, 8KByte). Also, when RAMPS# is asserted, the SCB Array RAM internal to the AIC-7870 is reconfigured for QINFIFO and QOUTFIFO data (SCB values of 0-255) storage and provide 256 bytes each. The QINCNT/ QOUTCNT count (0 = empty, 255 = full) supports 255 SCBs in the Q(IN/OUT)FIFOs at one time. The SCBPTR register maximum stored value is FFh. RAMPS# input has an internal pull-up and only needs to be connected when an external SRAM is present. Alternately, RAMPS# may be forced low during RST# assertion then placed at a high =1 level prior to the fourth IDDAT clock provided by LED# after a RST# assertion. The fourth IDDAT clock internally samples the input state of RAMPS# and only when high will start sourcing a buffered PCLK which may be used by external board logic. The state of RAMPS# during RST# assertion is remembered in the RAMPSM bit in the DEVSTAT register. The state of RAMPSM determines the SCB configuration to be internal (=0) or external (=1).

Other Pin Signal Type Definition

Table 2-13. Other Pin Signal Type Definition

Type	Definition
in	**Input** is a is a standard input-only receive signal.
inpu-s/t	**Input-three-state** is a bidirectional signal, with control enabled output driver and an internally connected standard input with internally connected pull-up.
t/s	**Three-state** is an output signal with control enabled output driver.
in-t/s	**In-three-state** is a bidirectional signal, with control enabled output driver and an internally connected standard input.

Power Distribution Pin Signals

Table 2-14. Core Logic Power Pins

Pin Name	Usage	Definition
CVCCn	PWR	Core Logic Positive Voltage Supply
CGNDn	PWR	Core Logic Ground

No other external pins are connected to or utilize power supplied by Core Logic power pin except pin 23.

Table 2-15. SCSI Interface Power Pins

Pin Name	Usage	Definition
SVCCn	PWR	SCSI Bus Driver Positive Voltage Supply
SGNDn	PWR	SCSI Bus Driver Ground

The following external pins are connected to and utilize power supplied by these pins:

ACK#	IO#	RESET#	SEL#
ATN#	LED#	SCD[15:0]#	STPWCTL
BSY#	MSG#	SCDPH#	
CD#	REQ#	SCDPL#	

Table 2-16. PCI Interface Power Pins

Pin Name	Usage	Definition
PVCCn	PWR	PCI Bus Driver Positive Voltage Supply
PGNDn	PWR	PCI Bus Driver Ground

The following external pins are connected to and utilize power supplied by the PVCCn pins:

AD[31:00]	DIFFADR[1:0]	GNT#	PERR#
CBE[3:0]#	DIFFSTRB#	IDSEL	PREQ#
CLKIN	EXTARBACK#	IRDY#	RST#
DEVSEL#	EXTARBREQ#	IRQA#	SERR#
DIFFDAT[3:0]	FRAME#	PAR	STOP#
			TRDY#

The following external pins are connected to and utilize power supplied by the PGNDn pins:

AD[31:00]	EXTARBREQ#	MD[7:0]	RAMPS#
CBE[3:0]#	FRAME#	MDP	ROMCS#
CLKIN	GNT#	MRW	RST#
DEVSEL#	IDDAT	PAR	SEECS
DIFFDAT[3:0]	IDSEL	PCLK	SERR#
DIFFADR[1:0]	IRDY#	PERR#	STOP#
DIFFSTRB#	IRQA#	PREQ#	TRDY#
EXTARBACK#	MA[14:0]	RAMCS#	WIDEPS#

Table 2-17. Memory Port Interface Power Pins

Pin Name	Usage	Definition
PVCCn	PWR	PCI Bus Driver Positive Voltage Supply
PGNDn	PWR	PCI Bus Driver Ground

The following external pins are connected to and utilize power supplied by these pins:

IDDAT	MD[7:0]	RAMCS#	SEECS#
IRQA#	MDP	RAMPS#	WIDEPS#
MA[14:0]	MRW	ROMCS#	

3 Register Description

About This Chapter

Read this chapter to find out

- A summary of the complete name and mnemonic for each register in the AIC-7870 divided into groups with a common address range
- A summary of the AIC-7870 Register Bit Definitions

▼▼▼▼▼ 3

AIC-7870 Register Block Address Map

The AIC-7870 internal device register block address map (for PCI bus access) is contained within a 256 address range of M00h-MFFh where M is a prefix base address supplied in BASEADR0 or BASEADR1 Configuration space registers.

The AIC-7870 internal configuration register block address map (for PCI bus access) is contained within a 256 address range of N00h-NFFh where N is a prefix address supplied by one of AD[31:11] which is expected to be connected to the AIC-7870 pin IDSEL.

For sequencer access, the Device register block address map is 256 addresses for reading and 256 addresses for writing, both active at the same time, with the range of 00h-FFh and indicated as DS-xx-xx. The block ranges are summarized below.

Register Block	Address Range
SCSI	M00-M1F, DS-00-1F
Scratch RAM	M20-M5F, DS-20-5F
Sequencer (SCSI PhaseEngine)	M60-M7F, DS-60-7F
Host, SCB, Data FIFO	M80-M9F, DS-80-9F
SCB Array	MA0-MBF, DS-A0-BF
RSVD	MC0-MDF, DS-C0-DF
RSVD	ME0=MFF, DS-E0-FF
Configuration (defined header)	N00-N3F
Configuration (device specific) AIC-7870 Not Used	 N40 N41-NFF

AIC-7870 Register Summary

The summary below lists the complete name and mnemonic of each register in the chip. The list is divided into groups with a common address range. When the host must access these registers the sequencer must be paused, except when noted otherwise.

SCSI Device Registers

M00-M1F, DS-00-1F*

Table 3-1. SCSI Device Registers

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
SCSI Sequencer Control	SCSISEQ	R/W	8/8	00	M00	
SCSI Transfer Control 0	SXFRCTL0	R/W	6/6	00	M01	
SCSI Transfer Control 1	SXFRCTL1	R/W	8/8	00	M02	
SCSI Signal In	SCSISIGI	R/	8/	XX	M03	
SCSI Signal Out	SCSISIGO	/W	/8	–	M03	
SCSI Rate	SCSIRATE	R/W	8/8	00	M04	
SCSI ID	SCSIID	R/W	8/8	00	M05	
SCSI Latched Data Low	SCSIDATL	R/W	8/8	XX	M06	
SCSI Latched Data High	SCSIDATH	R/W	8/8	XX	M07	
SCSI Transfer Count 0	STCNT0	R/W	8/8	00	M08	
SCSI Transfer Count 1	STCNT1	R/W	8/8	00	M09	
SCSI Transfer Count 2	STCNT2	R/W	8/8	00	M0A	
Clear SCSI Interrupt 0	CLRSINT0	/W	/5	–	M0B	
Clear SCSI Interrupt 1	CLRSINT1	/W	/7	–	M0C	
SCSI Status 0	SSTAT0	R/	8/	00	M0B	
SCSI Status 1	SSTAT1	R/	8/	00	M0C	
SCSI Status 2	SSTAT2	R/	6/	00	M0D	
SCSI Status 3	SSTAT3	R/	8/	00	M0E	
SCSI Test	SCSITEST	R/W	4/4	00	M0F	
SCSI Interrupt Mode 0	SIMODE0	R/W	7/7	00	M10	
SCSI Interrupt Mode 1	SIMODE1	R/W	8/8	00	M11	
SCSI Data Bus Low	SCSIBUSL	R/	8/	XX	M12	
SCSI Data Bus High	SCSIBUSH	R/	8/	XX	M13	
SCSI Count Host Address 0	SHADDR0	R/	8/	00	M14	Written when HADDLDSEL[1:0]=0 with same value as LHADDR0
SCSI Count Host Address 1	SHADDR1	R/	8/	00	M15	Written when HADDLDSEL[1:0]=0 with same value as LHADDR1
SCSI Count Host Address 2	SHADDR2	R/	8/	00	M16	Written when HADDLDSEL[1:0]=0 with same value as LHADDR2

Table 3-1. SCSI Device Registers *(Continued)*

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
SCSI Count Host Address 3	SHADDR3	R/	8/	00	M17	Written when HADDLDSEL[1:0]=0 with same value as LHADDR3
Selection Time-out Timer	SELTIMER	R/	7/	00	M18	
Selection/Reselection ID	SELID	R/	5/	00	M19	
SCSI Block Control	SBLKCTL	R/W	4/4	0[1]	M1F	

[1] Value is affected by external connections to WIDEPS#.

* The IOPAGE bit affects access to these registers. Also, address 1A and 1B are spare, 1C is SLEEPCTL, 1D is BRDCTL, and 1Eh is SEECTL.

Scratch RAM Device Register

M20-M5F, DS-20-5F

The scratch RAM area is addressed directly by the sequencer or host driver and is organized as 64 dual-ported byte locations. It is used to store information about the SCSI bus setup, the current operation, or system parameters. It is also used by the sequencer for temporary storage during operation. The sequencer may do read-modify-write accesses in a single access while the host driver may only do either a read or a write during an access. The scratch RAM also contains the capability to be parity protected, see MPARCKEN in the Device Command register.

Table 3-2. Scratch RAM Device

Register Name	Mnemonic	Read/Write	Comments
Scratch RAM (64 bytes)	Scratch	R/W	

Sequencer Device Registers

M60-M7F, DS-60-7F

Table 3-3. Sequencer Device Registers

Register name	Mnemonic	Read/Write	Size	Init	Address	Comments
Sequencer Control	SEQCTL	R/W	8/8	90	M60	
Sequencer RAM	SEQRAM	R/W	8/8	XX	M61	
Sequencer RAM Address Low	SEQADDRL	R/W	8/8	00	M62	
Sequencer RAM Address High	SEQADDRH	R/W	1/1	00	M63	
Accumulator	ACCUM	R/W	8/8	00	M64	
Source Index Register	SINDEX	R/W	8/8	00	M65	
Destination Index Register	DINDEX	R/W	8/8	00	M66	
Break Address 0	BRKADDR0	R/W	8/8	00	M67	
Break Address 1	BRKADDR1	R/W	2/2	80	M68	
All Ones	ALLONES	R/	8/	FF	M69	
All Zeros	ALLZEROS	R/	8/	00	M6A	
No destination	NONE	/W	/8	–	M6A	
Flags	FLAGS	R/	2/	00	M6B	
Source Indirect	SINDIR	R/	8/	00	DS-6C	Not usable by host driver
Destination Indirect	DINDIR	/W	/8	–	DS-6D	Not usable by host driver
Function 1	FUNCT1	R/W	3/3	XX	M6E	
Stack	STACK	R/	8/	00	M6F	

Host Device Registers

M80-M9F, DS-80-9F

Table 3-4. Host Device Registers

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
DSVendor Identification 0	DSVENDID0	R/	8/	04	M80	Host only
DSVendor Identification 1	DSVENDID1	R/	8/	90	M81	Host only
DSDevice Identification 0	DSDEVID0	R/	8/	78	M82	Host only
DSDevice Identification 1	DSDEVID1	R/	8/	[1]	M83	Host only
DSCommand	DSCOMMAND	R/W	8/4	00	M84	
DSLatency Timer	DSLATTIME	R/	8/	00	M85	
DSPCI Status	DSPCISTATUS	R/W	8/2	00	M86	
Host Control	HCNTRL	R/W	7/7	05	M87	Host only
Low Host Address 0	LHADDR0	R/W	8/8	00	M88	HADDLDSEL[1:0] = 0
Low Host Address 1	LHADDR1	R/W	8/8	00	M89	
Low Host Address 2	LHADDR2	R/W	8/8	00	M8A	
Low Host Address 3	LHADDR3	R/W	8/8	00	M8B	
High Host Address 0	HHADDR0	R/W	8/8	00	M88	
High Host Address 1	HHADDR1	R/W	8/8	00	M89	
High Host Address 2	HHADDR2	R/W	8/8	00	M8A	
High Host Address 3	HHADDR3	R/W	8/8	00	M8B	
Host Byte Count 0	HCNT0	R/W	8/8	00	M8C	
Host Byte Count 1	HCNT1	R/W	8/8	00	M8D	
Host Byte Count 2	HCNT2	R/W	8/8	00	M8E	
Interrupt Status	INTSTAT	/W	/6	00	M91	
Interrupt Status	INTSTAT	R/	8/6	00	M91	Host only
Clear Interrupt	CLRINT	/W	/4	–	M92	Host only
Error	ERROR	R/	7/	00	M92	Host only
Data FIFO Control	DFCNTRL	R/W	7/7	00	M93	
Data FIFO Status	DFSTATUS	R/	7/	21	M94	
Queue Out FIFO	QOUTFIFO	R/	8/	00	M9C	Read by host only
Queue Out Count	QOUTCNT	R/	8/	00	M9E	Read by host only
Special Function	SFUNCT	R/W	8/8	00	M9F	
Sleep Control	SLEEPCTL	R/W	3/3	00	M1C	

[1] Internal default value (70) or external entered value.

Host Configuration Registers

N00-NFF

Note: Register bytes not listed are reserved, read only and always return 0h when read.

Table 3-5. Host Configuration Registers

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
Vendor Identification 0	VENDORID0	R/	8/	04	00	Byte 0 host only
Vendor Identification 1	VENDORID1	R/	8/	90	N00	Byte 1 host only
Device Identification 0	DEVICEID0	R/	8/	78	N00	Byte 2 host only
Device Identification 1	DEVICEID1	R/	8/	[1]	N00	Byte 3 host only
Command 0	COMMAND0	R/W	8/5	00	N04	Byte 0 host only
Command 1	COMMAND1	R/W	8/1	00	N04	Byte 1 host only
PCI Status 0	STATUS0	R/	8/	00	N04	Byte 2 host only
PCI Status 1	STATUS1	R/[2]	8/6	02	N04	Byte 3 host only
Device Revision ID	DEVREVID	R/	8/	00	N08	Byte 0 host only
Programming Interface	PROGINFC	R/	8/	00	N08	Byte 1 host only
Sub Class	SUBCLASS	R/	8/	00	N08	Byte 2 host only
Base Class	BASECLASS	R/	8/	00	N08	Byte 3 host only
Cache Line Size	CACHESIZE	R/W	8/4	00	N0C	Byte 0 host only
Latency Time	LATTIME	R/W	8/6	00	N0C	Byte 1 host only
Header Type	HDRTYPE	R/	8/	00	N0C	Byte 2 host only
Base Address 0 (32-bits)	BASEADR0	R/W	32/24	02	N10	Byte 0-3 host only
Base Address 1 (32-bits)	BASEADR1	R/W	32/20	00	N14	Byte 0-3 host only
External ROM Control (32-bits)	EXROMCTL	R/W	32/17	00	N30	Byte 0-3 host only
Interrupt Line Select	INTLINSEL	R/W	8/8	00	N3C	Byte 0 host only
Interrupt Pin Select	INTPINSEL	R/	8/	01	N3C	Byte 1 host only
Min_Gnt Status	MINGNT	R/	8/	08	N3C	Byte 2 host only
Max_Lat Status	MAXLAT	R/	8/	08	N3C	Byte 3 host only
Device Configuration	DEVCONFIG	R/W	8/8	00	N40	Byte 0 host only
Device Status	DEVSTATUS	R/	3/	[3]	N40	Byte 1 host only

[1] Internal (70) or external entered value.
[2] Write of a one value forces active read value to be inactive. Internal default value.
[3] Depends on external conditions (PCI bus voltage, RAMPS#, EXTARBACK#).

SCB Device Registers

M90, M9A-M9D, DS-90, 9A-9D

Table 3-6. SCB Device Registers

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
SCB Pointer	SCBPTR	R/W	8/8	00	M90	
SCB Counter	SCBCNT	R/W	6/6	00	M9A	
Queue In FIFO	QINFIFO	R/W	8/8	00	M9B	
Queue In Count	QINCNT	R/	8/	00	M9C	
Queue Out FIFO	QOUTFIFO	/W	/8	–	M9D	

Memory Port Device Registers

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
Serial EEPROM Control	SEECTL	R/W	8/4	[1]	M1E	
Board Control	BRDCTL	R/W	8/6	00	M1D	8/6 00 M1D

[1] Depends on external conditions (EXTARBACK#, EXTARBREQ#).

Data FIFO Device Registers

M95, M97, M99 DS-95, 97, 99

Table 3-7. Data FIFO Device Registers

Register Name	Mnemonic	Read/Write	Size	Init	Address	Comments
Data FIFO Write Address	DFWADDR	R/W	7/6	00	M95	
Data FIFO Read Address	DFRADDR	R/W	7/6	00	M97	
Data FIFO Data	DFDAT	R/W	8/8	XX	M99	

SCB Array

MA0-MBF, DS-A0-BF

The SCB Array contains a number of 32 byte pages. SCB page selection is determined by the value stored in the Device SCBPTR register. SCB Array pages that have been loaded with valid SCB data are indicated by loading their SCB Array page values into the Queue In FIFO. The sequencer may do read-modify-write instructions in a single access to a selected page. The host driver can only do a read or a write instruction during an access to a selected page, and only when the sequencer is paused (see the PAUSE bit in the device HCNTRL register).

The number of SCBs that may be stored at one time is affected by the state of RAMPSM. When RAMPSM is inactive, sixteen SCBs may be stored in the AIC-7870 internal SCB Array. The SCB Array may be expanded to a maximum of 256 pages with the addition of external SRAM and setting RAMPSM active. The size of the external ram must be verified by the software driver prior to loading SCB data. The sequencer will process SCBs

located in external RAM the same as located in internal RAM. For additional information see the RAMPS# and *Memory Port* section.

The SCB Array also contains the capability to be parity protected, see MPARCKEN in the Device Command register and EXTSCBPEN in the Configuration DEVCONFIG register. The external SCB RAM may be shared with other devices, see EXTARBREQ#/ EXTARBACK#. The lower 5-bits of the SCB Array address are the same as bits AD[4:2] of the host address bus with bits [1:0] decoded from the single asserted CBE[3:0]# bit, or bits [4:0] of the sequencer address buses, or bits SCBCNT[4:0] of the SCBCNT register.

Table 3-8. SCB Array

Register Name	Mnemonic	Read/Write	Comments
SCB Array (pages 0-15) Inactive	SCB	R/W	Internal RAM, RAMPSM
SCB Array (pages 0-255) RAMPSM Active	SCB	R/W	External RAM

AIC-7870 Register Bit Definition Summary

Device register address access source is indicated by adding a prefix to the basic register address: PCI source (M, N), Sequencer (DS). All unused and reserved register bits read as 0, and should be written with 0. All unassigned register locations return 0. Reserved (RSVD) denotes anticipated usage. Unused bits are available for future definition.

SCSI Registers

SCSISEQ R/W M/DS-00 (b)	SXFRCTL0 R/W M/DS-01 (b)	SXFRCTL1 R/W M/DS-02 (b)	SCSISIGO W M/DS-03 (b)	SCSISIGI R M/DS-03 (b)	SCSIRATE R/W M/DS-04 (b)
7 TEMODEO	7 DFON	7 BITBUCKET	7 CDO	7 CDI	7 WIDEXFER
6 ENSELO	6 DFPEXP	6 SWRAPEN	6 IOO	6 IOI	6 SXFR(2)
5 ENSELI	5	5 ENSPCHK	5 MSGO	5 MSGI	5 SXFR(1)
4 ENRSELI	4 CLRSTCNT	4 STIMESEL(1)	4 IATNO	4 ATNI	4 SXFR(0)
3 ENAUTOATNO	3 SPIOEN	3 STIMESEL(0)	3 SELO	3 SELI	3 SOFS(3)
2 ENAUTOATNI	2 SCAMEN	2 ENSTIMER	2 BSYO	2 BSYI	2 SOFS(2)
1 ENAUTOATNP	1 CLRCHN	1 ACTNEGEN	1 TREQO	1 REQI	1 SOFS(1)
0 SCSIRSTO	0	0 STPWEN	0 ACKO	0 ACKI	0 SOFS(0)

SCSIID R/W M/DS-05 (b)	SCSIDATL R/W M/DS-06 (b)	SCSIDATH R/W M/DS-07 (b0	STCNT0 R/W M/DS-08 (b)	STCNT1 R/W M/DS-09 (b)	STCNT2 R/W M/DS-0A (b)
7 TID(3)	7 DB(07)	7 DB(15)	7 STCNT(07)	7 STCNT(15)	7 STCNT(23)
6 TID(2)	6 DB(06)	6 DB(14)	6 STCNT(06)	6 STCNT(14)	6 STCNT(22)
5 TID(1)	5 DB(05)	5 DB(13)	5 STCNT(05)	5 STCNT(13)	5 STCNT(21)
4 TID(0)	4 DB(04)	4 DB(12)	4 STCNT(04)	4 STCNT(12)	4 STCNT(20)
3 OID(3)	3 DB(03)	3 DB(11)	3 STCNT(03)	3 STCNT(11)	3 STCNT(19)
2 OID(2)	2 DB(02)	2 DB(10)	2 STCNT(02)	2 STCNT(10)	2 STCNT(18)
1 OID(1)	1 DB(01)	1 DB(09)	1 STCNT(01)	1 STCNT(09)	1 STCNT(17)
0 OID(0)	0 DB(00)	0 DB(08)	0 STCNT(00)	0 STCNT(08)	0 STCNT(16)

CLRSINT0 W M/DS-0B (b)	SSTAT0 R M/DS-0B (b)	CLRSINT1 W M/DS-0C (b)	SSTAT1 R M/DS-0C (b)	SSTAT2 R M/DS-0D (b)	SSTAT3 R M/DS-0E (b)
7	7 TARGET	7 CLRSELTIMO	7 SELTO	7 OVERRUN	7 SCSICNT(3)
6 CLRSELDO	6 SELDO	6 CLRATNO	6 ATNTARG	6	6 SCSICNT(2)
5 CLRSELDI	5 SELDI	5 CLRSCSIRSTI	5 SCSIRSTI	5	5 SCSICNT(1)
4 CLRSELINGO	4 SELINGO	4	4 PHASEMIS	4 SFCNT(4)	4 SCSICNT(0)
3 CLRSWRAP	3 SWRAP	3 CLRBUSFREE	3 BUSFREE	3 SFCNT(3)	3 OFFCNT(3)
2	2 SDONE	2 CLRSCSIPERR	2 SCSIPERR	2 SFCNT(2)	2 OFFCNT(2)
1 CLRSPIORDY	1 SPIORDY	1 CLRPHASECHG	1 PHASECHG	1 SFCNT(1)	1 OFFCNT(1)
0	0 DMADONE	0 CLRREQINIT	0 REQINIT	0 SFCNT(0)	0 OFFCNT(0)

SCSITEST R/W M/DS-0F (b)	SIMODE0 R/W M/DS-10 (b)	SIMODE1 R/W M/DS-11 (b)	SCSIBUSL R M/DS-12 (b)	SCSIBUSH R M/DS-13 (b)	SHADDR0 R M/DS-14 (bi)
7	7	7 ENSELTIMO	7 SDB(07)	7 SDB(15)	7 SHADDR(07)
6	6 ENSELDO	6 ENATNTARG	6 SDB(06)	6 SDB(14)	6 SHADDR(06)
5	5 ENSELDI	5 ENSCSIRST	5 SDB(05)	5 SDB(13)	5 SHADDR(05)
4 DATALOOP	4 ENSELINGO	4 ENPHASEMIS	4 SDB(04)	4 SDB(12)	4 SHADDR(04)
3	3 ENSWRAP	3 ENBUSFREE	3 SDB(03)	3 SDB(11)	3 SHADDR(03)
2 RQAKCNT	2 ENSDONE	2 ENSCSIPERR	2 SDB(02)	2 SDB(10)	2 SHADDR(02)
1 CNTRTEST	1 ENSPIORDY	1 ENPHASECHG	1 SDB(01)	1 SDB(09)	1 SHADDR(01)
0 CTSTMODE	0 ENDMADONE	0 ENREQINIT	0 SDB(00)	0 SDB(08)	0 SHADDR(00)

SHADDR1 R M/DS-15 (bi)	SHADDR2 R M/DS-16 (bi)	SHADDR3 R M/DS-17 (bi)	SELTIMER R M/DS-18 (b)	SELID R M/DS-19 (b)	SBLKCTL R/W M/DS-1F (b)
7 SHADDR(15)	7 SHADDR(23)	7 SHADDR(31)	7 CLKOUT	7 SELID(3)	7 DIAGLEDEN
6 SHADDR(14)	6 SHADDR(22)	6 SHADDR(30)	6	6 SELID(2)	6 DIAGLEDON
5 SHADDR(13)	5 SHADDR(21)	5 SHADDR(29)	5 STAGE 6	5 SELID(1)	5 AUTOFLUSHDIS
4 SHADDR(12)	4 SHADDR(20)	4 SHADDR(28)	4 STAGE 5	4 SELID(0)	4
3 SHADDR(11)	3 SHADDR(19)	3 SHADDR(27)	3 STAGE 4	3 ONEBIT	3 SELBUSB=0
2 SHADDR(10)	2 SHADDR(18)	2 SHADDR(26)	2 STAGE 3	2	2
1 SHADDR(09)	1 SHADDR(17)	1 SHADDR(25)	1 STAGE 2	1	1 SELWIDE
0 SHADDR(08)	0 SHADDR(16)	0 SHADDR(24)	0 STAGE 1	0	0

Sequencer Registers

SEQCTL R/W M/DS-60 (b)	SEQRAM R/W M/DS-61 (b)	SEQADDR0 R/W M/DS-62 (b)	SEQADDR1 R/W M/DS-63 (b)	ACCUM R/W M/DS-64 (b)	SINDEX R/W M/DS-65 (b)
7 PERRORDIS	7 SEQRAM(7)	7 SEQADDR(07)	7	7 ACCUM(7)	7 SINDEX(7)
6 PAUSEDIS	6 SEQRAM(6)	6 SEQADDR(06)	6	6 ACCUM(6)	6 SINDEX(6)
5 FAILDIS	5 SEQRAM(5)	5 SEQADDR(05)	5 RSVD	5 ACCUM(5)	5 SINDEX(5)
4 FASTMODE	4 SEQRAM(4)	4 SEQADDR(04)	4 RSVD	4 ACCUM(4)	4 SINDEX(4)
3 BRKADRINTEN	3 SEQRAM(3)	3 SEQADDR(03)	3 RSVD	3 ACCUM(3)	3 SINDEX(3)
2 STEP	2 SEQRAM(2)	2 SEQADDR(02)	2 RSVD	2 ACCUM(2)	2 SINDEX(2)
1 SEQRESET	1 SEQRAM(1)	1 SEQADDR(01)	1 RSVD	1 ACCUM(1)	1 SINDEX(1)
0 LOADRAM	0 SEQRAM(0)	0 SEQADDR(00)	0 SEQADDR(08)	0 ACCUM(0)	0 SINDEX(0)

DINDEX R/W M/DS-66 (b)	BRKADDR0 R/W M/DS-67 (b)	BRKADDR1 R/W M/DS-68 (b)	ALLONES R M/DS-69 (b)	ALLZEROS R M/DS-6A (b)	NONE W M/DS-6A (b)
7 DINDEX(7)	7 BRKADDR(07)	7 BRKDIS	7 (1)	7 (0)	7
6 DINDEX(6)	6 BRKADDR(06)	6	6 (1)	6 (0)	6
5 DINDEX(5)	5 BRKADDR(05)	5 RSVD	5 (1)	5 (0)	5
4 DINDEX(4)	4 BRKADDR(04)	4 RSVD	4 (1)	4 (0)	4
3 DINDEX(3)	3 BRKADDR(03)	3 RSVD	3 (1)	3 (0)	3
2 DINDEX(2)	2 BRKADDR(02)	2 RSVD	2 (1)	2 (0)	2
1 DINDEX(1)	1 BRKADDR(01)	1 RSVD	1 (1)	1 (0)	1
0 DINDEX(0)	0 BRKADDR(00)	0 BRKADDR(08)	0 (1)	0 (0)	0

FLAGS R M/DS-6B (b)	SINDIR R DS-6C	DINDIR W DS-6D	FUNCTION1 W M/DS-6E (b)	FUNCTION1 R M/DS-6E (b)	STACK R M/DS-6F (b)
7	CONTENTS	CONTENTS	7 RSVD	1 OF 8	7 STACK(07)
6	POINTED TO BY	POINTED TO BY	6 FUN1DAT(2)	DECODED	6 STACK(06)
5	SINDEX	DINDEX	5 FUN1DAT(1)	VALUE OF	5 STACK(05)
4			4 FUN1DAT(0)	FUN1DAT2-0	4 STACK(04)
3			3 RSVD		3 STACK(03)
2			2 RSVD		2 STACK(02)
1 ZERO			1 RSVD		1 STACK(01)
0 CARRY			0 RSVD		0 STACK(00)

Host Configuration Registers

DEVICEID1 R N00h **+	DEVICEID0 R N00h **	VENDORID1 R N00h **	VENDORID0 R N00h **
31 DID15=0	23 DID07=0 \	15 VID15=1	07 VID07=0
30 DID14=1	22 DID06=1 | MS	14 VID14=0	06 VID06=0
29 DID13=1	21 DID05=1 |	13 VID13=0	05 VID05=0
28 DID12=1	20 DID04=1 /	12 VID12=1	04 VID04=0
27 DID11=0 \	19 DID03=1	11 VID11=0	03 VID03=0
26 DID10=0 | LS	18 DID02=0	10 VID10=0	02 VID02=1
25 DID09=0 |	17 DID01=0	09 VID09=0	01 VID01=0
24 DID08=0 /	16 DID00=0	08 VID08=0	00 VID00=0

STATUS1 R/W N04h **	STATUS0 R N04h	COMMAND1 R/W N04h..**	COMMAND0 R/W N04h **
31 DPE	23 TFBFBC=1	15 RSVD	07 WAITCTLEN=0
30 SSE	22 RSVD	14 RSVD	06 PERRESPEN
29 RMA	21 RSVD	13 RSVD	05 VSNOOPEN=0
28 RTA	20 RSVD	12 RSVD	04 MWRICEN
27 STA	19 RSVD	11 RSVD	03 SPCYCEN=0
26 DST1=0	18 RSVD	10 RSVD	02 MASTEREN
25 DST0=1	17 RSVD	09 MFBFEN=0	01 MSPACEEN
24 DPR	16 RSVD	08 SERRESPEN	00 ISPACEEN

BASECLASS R N08h	SUBCLASS R N08h	PROGINFC R N08h	DEVREVID R N08h ++
31 BCLASS7=0	23 SCLASS7=0	15 PINFC7=0	07 DRID7=0
30 BCLASS6=0	22 SCLASS6=0	14 PINFC6=0	06 DRID6=0
29 BCLASS5=0	21 SCLASS5=0	13 PINFC5=0	05 DRID5=0
28 BCLASS4=0	20 SCLASS4=0	12 PINFC4=0	04 DRID4=0
27 BCLASS3=0	19 SCLASS3=0	11 PINFC3=0	03 DRID3=0
26 BCLASS2=0	18 SCLASS2=0	10 PINFC2=0	02 DRID2=0
25 BCLASS1=0	17 SCLASS1=0	09 PINFC1=0	01 DRID1=0
24 BCLASS0=1	16 SCLASS0=0	08 PINFC0=0	00 DRID0=1

RSVD R N0Ch	HDRTYPE R N0Ch	LATTIME R/W N0Ch **	CACHESIZE R/W N0Ch
31 RSVD	23 MFDEV=0	15 LATT7	07 RSVD
30 RSVD	22 HTYPE6=0	14 LATT6	06 RSVD
29 RSVD	21 HTYPE5=0	13 LATT5	05 CDWDSIZE5
28 RSVD	20 HTYPE4=0	12 LATT4	04 CDWDSIZE4
27 RSVD	19 HTYPE3=0	11 LATT3	03 CDWDSIZE3
26 RSVD	18 HTYPE2=0	10 LATT2	02 CDWDSIZE2
25 RSVD	17 HTYPE1=0	09 LATT1=0	01 CDWDSIZE1=0
24 RSVD	16 HTYPE0=0	08 LATT0=0	00 CDWDSIZE0=0

BASEADR0 R/W N10h	BASEADR0 R/W N10h	BASEADR0 R/W N10h	BASEADR0 R/W N10h
31 IBMADR29	23 IBMADR21	15 IBMADR13	07 IBMADR05=0
30 IBMADR28	22 IBMADR20	14 IBMADR12	06 IBMADR04=0
29 IBMADR27	21 IBMADR19	13 IBMADR11	05 IBMADR03=0
28 IBMADR26	20 IBMADR18	12 IBMADR10	04 IBMADR02=0
27 IBMADR25	19 IBMADR17	11 IBMADR09	03 IBMADR01=0
26 IBMADR24	18 IBMADR16	10 IBMADR08	02 IBMADR00=0
25 IBMADR23	17 IBMADR15	09 IBMADR07	01 RSVD
24 IBMADR22	16 IBMADR14	08 IBMADR06	00 ISPACEIND=1

BASEADR1 R/W N14h	BASEADR1 R/W N14h	BASEADR1 R/W N14h	BASEADR1 R/W N14h
31 MBMADR27	23 MBMADR19	15 MBMADR11	07 MBMADR03=0
30 MBMADR26	22 MBMADR18	14 MBMADR10	06 MBMADR02=0
29 MBMADR25	21 MBMADR17	13 MBMADR09	05 MBMADR01=0
28 MBMADR24	20 MBMADR16	12 MBMADR08	04 MBMADR00=0
27 MBMADR23	19 MBMADR15	11 MBMADR07=0	03 PREFETCH=0
26 MBMADR22	18 MBMADR14	10 MBMADR06=0	02 MSPACTYP1=0
25 MBMADR21	17 MBMADR13	09 MBMADR05=0	01 MSPACTYP0=0
24 MBMADR20	16 MBMADR12	08 MBMADR04=0	00 MSPACEIND=0

EXROMCTL R/W N30h	EXROMCTL R/W N30h	EXROMCTL R/W N30h	EXROMCTL R/W N30h
31 MBAXROM20	23 MBAXROM12	15 MBAXROM04=0	07 RSVD
30 MBAXROM19	22 MBAXROM11	14 MBAXROM05=0	06 RSVD
29 MBAXROM18	21 MBAXROM10	13 MBAXROM02=0	05 RSVD
28 MBAXROM17	20 MBAXROM09	12 MBAXROM01=0	04 RSVD
27 MBAXROM16	19 MBAXROM08	11 MBAXROM00=0	03 RSVD
26 MBAXROM15	18 MBAXROM07	10 RSVD	02 RSVD
25 MBAXROM14	17 MBAXROM06	09 RSVD	01 RSVD
24 MBAXROM13	16 MBAXROM05	08 RSVD	00 EXROMEN

MAXLAT R N3Ch	MINGNT R N3Ch	INTPINSEL R N3Ch	INTLINSEL R/W N3Ch
31 MAXLAT7=0	23 MINGNT7=0	15 INTPS7=0	07 INTLS7
30 MAXLAT6=0	22 MINGNT6=0	14 INTPS6=0	06 INTLS6
29 MAXLAT5=0	21 MINGNT5=0	13 INTPS5=0	05 INTLS5
28 MAXLAT4=0	20 MINGNT4=0	12 INTPS4=0	04 INTLS4
27 MAXLAT3=1	19 MINGNT3=1	11 INTPS3=0	03 INTLS3
26 MAXLAT2=0	18 MINGNT2=0	10 INTPS2=0	02 INTLS2
25 MAXLAT1=0	17 MINGNT1=0	09 INTPS1=0	01 INTLS1
24 MAXLAT0=0	16 MINGNT0=0	08 INTPS0=1	00 INTLS0

RSVD R N40h	RSVD R N40h	DEVSTATUS R N40h	DEVCONFIG R/W N40h
31 RSVD	23 RSVD	15	07 DEVCONFIG7
30 RSVD	22 RSVD	14	06 MRDCEN
29 RSVD	21 RSVD	13	05 EXTSCBTIME
28 RSVD	20 RSVD	12	04 EXTSCBPEN
27 RSVD	19 RSVD	11	03 BERREN
26 RSVD	18 RSVD	10 MPORTMODE	02 DACEN
25 RSVD	17 RSVD	09 RAMPSM	01 STPWLEVEL
24 RSVD	16 RSVD	08 VOLSENSE	00 DIFACTNEGEN

Host Device Registers

DSVENDID0 R M/DS-80 (ae)	DSVENDID1 R M/DS-81 (ae)	DSDEVID0 R M/DS-82 (ae)	DSDEVID1 R M/DS-83 (aeg)	DSCOMMAND R/W M/DS-84 (be)	DSLATTIME R M/DS-85 (bef)
7 DSVID07=0	7 DSVID15=1	7 DSDID07=0	7 DSDID15=0	7 CACHETHEN *	7 DSLATT7
6 DSVID06=0	6 DSVID14=0	6 DSDID06=1	6 DSDID14=1	6 DPARCKEN *	6 DSLATT6
5 DSVID05=0	5 DSVID13=0	5 DSDID05=1	5 DSDID13=1	5 MPARCKEN *	5 DSLATT5
4 DSVID04=0	4 DSVID12=1	4 DSDID04=1	4 DSDID12=1	4 EXTREQLCK*	4 DSLATT4
3 DSVID03=0	3 DSVID11=0	3 DSDID03=1	3 DSDID11=0	3 DSSERRESPEN	3 DSLATT3
2 DSVID02=1	2 DSVID10=0	2 DSDID02=0	2 DSDID10=0	2 DSPERRESPEN	2 DSLATT2
1 DSVID01=0	1 DSVID09=0	1 DSDID01=0	1 DSDID09=0	1 DSMWICEN	1 HADDLDSEL1 *
0 DSVID00=0	0 DSVID08=0	0 DSDID00=0	0 DSDID08=0	0 DSMASTEREN	0 HADDLDSEL0 *

PCISTATUS R/W M/DS-86 (be)	HCNTRL R/W M/DS-87 (a)	LHADDR0 R/W M/DS-88 (bcdh)	LHADDR1 R/W M/DS-89 (bch)	LHADDR2 R/W M/DS-8A (bch)	LHADDR3 R/W M/DS-8B (bch)
7 DFTHRSH1 *	7	7 LHADDR07	7 LHADDR15	7 LHADDR23	7 LHADDR31
6 DFTHRSH0 *	6 POWRDN	6 LHADDR06	6 LHADDR14	6 LHADDR22	6 LHADDR30
5 DSDPR	5	5 LHADDR05	5 LHADDR13	5 LHADDR21	5 LHADDR29
4 DSDPE	4 SWINT	4 LHADDR04	4 LHADDR12	4 LHADDR20	4 LHADDR28
3 DSSSE	3 HCNTRL3	3 LHADDR03	3 LHADDR11	3 LHADDR19	3 LHADDR27
2 DSRMA	2 PAUSE[ACK]	2 LHADDR02	2 LHADDR10	2 LHADDR18	2 LHADDR26
1 DSRTA	1 INTEN	1 LHADDR01	1 LHADDR09	1 LHADDR17	1 LHADDR25
0 DSSTA	0 CHIPRST[ACK]	0 LHADDR00	0 LHADDR08	0 LHADDR16	0 LHADDR24

		HHADDR0 R/W M/DS-88 (bc)	HHADDR1 R/W M/DS-89 (bc)	HHADDR2 R/W M/DS-8A (bc)	HHADDR3 R/W M/DS-8B (bc)
		7 HHADDR07	7 HHADDR15	7 HHADDR23	7 HHADDR31
		6 HHADDR06	6 HHADDR14	6 HHADDR22	6 HHADDR30
		5 HHADDR05	5 HHADDR13	5 HHADDR21	5 HHADDR29
		4 HHADDR04	4 HHADDR12	4 HHADDR20	4 HHADDR28
		3 HHADDR03	3 HHADDR11	3 HHADDR19	3 HHADDR27
		2 HHADDR02	2 HHADDR10	2 HHADDR18	2 HHADDR26
		1 HHADDR01	1 HHADDR09	1 HHADDR17	1 HHADDR25
		0 HHADDR00	0 HHADDR08	0 HHADDR16	0 HHADDR24

HCNT0 R/W M/DS-8C (bh)	HCNT1 R/W M/DS-8D (bh)	HCNT2 R/W M/DS-8E (bh)	RSVD R/W M/DS-8F (b)	INTSTAT R /W M/DS-91 (bkl)	CLRINT W M/DS-92 (a)
7 HCNT07	7 HCNT15	7 HCNT23	7 RSVD	7 INTCODE3	7
6 HCNT06	6 HCNT14	6 HCNT22	6 RSVD	6 INTCODE2	6
5 HCNT05	5 HCNT13	5 HCNT21	5 RSVD	5 INTCODE1	5
4 HCNT04	4 HCNT12	4 HCNT20	4 RSVD	4 INTCODE0	4 CLRPARERR
3 HCNT03	3 HCNT11	3 HCNT19	3 RSVD	3 BRKINT	3 CLRBRKADRINT
2 HCNT02	2 HCNT10	2 HCNT18	2 RSVD	2 SCSIINT	2 CLRSCSIINT
1 HCNT01	1 HCNT09	1 HCNT17	1 RSVD	1 CMDCMPLT	1 CLRCMDINT
0 HCNT00	0 HCNT08	0 HCNT16	0 RSVD	0 SEQINT	0 CLRSEQINT

ERROR R M/DS-92 (a)	DFCNTRL R/W M/DS-93 (b)	DFSTATUS R M/DS-94 (b)	QOUTFIFO R M/DS-9D (jkl)	QOUTCNT R M/DS-9E (jk)	SFUNCT R/W M/DS-9F (b)
7	7	7	7 QOUT7	7 QOUTCNT7	7 SFUNCT2
6 PCIERRSTAT (g)	6 WIDEODD	6 DFCACHETH	6 QOUT6	6 QOUTCNT6	6 SFUNCT1
5 MPARERR	5 SCSIEN[ACK]	5 FIFOQWDEMP	5 QOUT5	5 QOUTCNT5	5 SFUNCT0
4 DPARERR	4 SDMAEN[ACK]	4 MREQPEND	4 QOUT4	4 QOUTCNT4	4 TESTRAM
3 SQPARERR	3 HDMAEN[ACK]	3 HDONE	3 QOUT3	3 QOUTCNT3	3 TESTHOST
2 ILLOPCODE	2 DIRECTION[ACK]	2 DFTHRSH	2 QOUT2	2 QOUTCNT2	2 TESTSEQ
1	1 FIFOFLUSH[ACK]	1 FIFOFULL	1 QOUT1	1 QOUTCNT1	1 TESTFIFO
0	0 FIFORESET(cd)	0 FIFOEMP	0 QOUT0	0 QOUTCNT0	0 TESTSCSI

SLEEPCTL R/W M/DS-1C (b)					
7 SLEEPDIS 6 5 4 3 2 1 SLP1 0 SLP0					

SCB Device Registers

SCBPTR R/W M/DS-90 (bj)	SCBCNT R/W M/DS-9A (bj)	QINFIFO R/W M/DS-9B (bj)	QINCNT R M/DS-9C (bj)	QOUTFIFO W M/DS-9D (bj)	
7 SCBVAL7	7 SCBAUTO	7 QIN7	7 QINCNT7	7 QOUT7	
6 SCBVAL6	6 RSVD	6 QIN6	6 QINCNT6	6 QOUT6	
5 SCBVAL5	5 RSVD	5 QIN5	5 QINCNT5	5 QOUT5	
4 SCBVAL4	4 SCBCNT4	4 QIN4	4 QINCNT4	4 QOUT4	
3 SCBVAL3	3 SCBCNT3	3 QIN3	3 QINCNT3	3 QOUT3	
2 SCBVAL2	2 SCBCNT2	2 QIN2	2 QINCNT2	2 QOUT2	
1 SCBVAL1	1 SCBCNT1	1 QIN1	1 QINCNT1	1 QOUT1	
0 SCBVAL0	0 SCBCNT0	0 QIN0	0 QINCNT0	0 QOUT0	

BRDCTL R/W M/DS-1D (b)	SEECTL R/W M/DS-1E (b)				
7 BRDDAT7	7 EXTARBACK				
6 BRDDAT6	6 EXTARBREQ				
5 BRDDAT5	5 SEEMS *				
4 BRDSTB *-	4 SEERDY				
3 BRDCS *-	3 SEECS *				
2 BRDRW *	2 SEECK *				
1 BRDCTL1 =0	1 SEEDO				
0 BRDCTL0 =0	0 SEEDI				

Data FIFO Device Registers

DFWADDR0 R/W M/DS-95 (bh)	RSVD R/W M/DS-96 (b)	DFRADDR0 R/W M/DS-97 (bh)	RSVD R/W M/DS-98 (b)	DFDAT R/W M/DS-99 (bh)	
7 DFCACHETHLA	7 RSVD	7 DFSDH	7 RSVD	7 FDAT7	
6	6 RSVD	6	6 RSVD	6 FDAT6	
5 DFWADDR05	5 RSVD	5 DFRADDR05	5 RSVD	5 FDAT5	
4 DFWADDR04	4 RSVD	4 DFRADDR04	4 RSVD	4 FDAT4	
3 DFWADDR03	3 RSVD	3 DFRADDR03	3 RSVD	3 FDAT3	
2 DFWADDR02	2 RSVD	2 DFRADDR02	2 RSVD	2 FDAT2	
1 DFWADDR01	1 RSVD	1 DFRADDR01	1 RSVD	1 FDAT1	
0 DFWADDR00	0 RSVD	0 DFRADDR00	0 RSVD	0 FDAT0	

a Register may be read or written by host driver without access or latched pausing the sequencer.
b Register may only be read or written by host driver when sequencer is access or latch paused.
c Access affected by the state of HADDLDSEL[1:0] value in the DSCOMMAND register.
d Also used to load starting byte offsets to quad word boundary.
e Allows dual access read of Configuration space registers or bits.
g ORed status read only.
h Register may only be accessed when the H/SDMAEN bits in the DFCNTRL register are not active.
i Write access same as LHADDR[3:0].
j Maximum value read or written affected by state of RAMPSM.
k Register may be read by host driver without access or latch pausing the sequencer.
l Register may be written by host driver only with access or latch pausing the sequencer.
* R/W bit
** Indicates dual read access to this register byte through the AIC-7870 Device register space.
+ Indicates this register byte read default value after RST# assertion may be changed.
++ Easy value update in metal
- External pin level is inverted of this bit when SEEMS is active.

4 Detailed Register Description

About This Chapter

Read this chapter to find out

- A detailed description of each of the registers in the AIC-7870

▼▼▼▼▼ 4

The following conventions are used throughout this section:

- **set:** Indicates that the bit was loaded with a 1
- **cleared:** Indicates that the bit was loaded with a 0
- is a **one:** Indicates a status of 1
- is a **zero:** Indicates a status of 0
- **(0):** Indicates that the bit is cleared when the reset pin is active
- **(1):** Indicates that the bit is set when the reset pin is active
- **(x):** Indicates that the bit is in an unknown state after the reset condition
- **Mxxh:** Indicates PCI address decode with address prefix stored in either BASEADR0 or BASEADR1 registers in the configuration space.
- **Nxxh:** Indicates PCI address decode with address prefix defined externally to the AIC-7870 by a connection to pin IDSEL, this is normally a single address line in the range of AD[31:11]. PCI convention for access will be to only have one address line active (=1) per device access.
- **DS-xxh:** Device space, internal address decode. It represents Mxxh without the M extension and is the value used by the internal sequencer (SCSI PhaseEngine).

SCSI Device Space Register Definition

SCSI Sequence Control (SCSISEQ)

Type: R/W
Address: M00h, DS-00h

Each bit, when set, enables the specified hardware sequence. The register is readable to allow bit manipulation instructions without saving a register image in scratch RAM. All bits except SCSIRSTO are cleared by SCSI bus reset.

SCSISEQ R/W	
7	TEMODEO
6	ENSELO
5	ENSELI
4	ENRSELI
3	ENAUTOATNO
2	ENAUTOATNI
1	ENAUTOATNP
0	SCSIRSTO

Bit		Name	Definition
7	(0)	TEMODEO	**Target Enable Mode Out.** This bit is used to select whether ENSELO will start a Selection Out (TEMODEO = 0) or a Reselection Out (TEMODEO = 1) SCSI bus sequence.
6	(0)	ENSELO	**Enable Selection Out.** When this bit is set to one, it will allow the SCSI logic to perform a Selection sequence (TEMODEO = 0) as an Initiator (ID = OID field of SCSIID register) and select a Target (ID = TID field of the SCSIID register), or to perform a Reselection sequence (TEMODEO=one) as a Target (ID = OID field of SCSIID register) and reselect an Initiator (ID = TID field of the SCSIID Register). The SELINGO status (bit 4, SSTAT0) is one when the SCSI logic has entered the Selection/Reselection phase and is waiting for BSY back from the Target/Initiator. The sequencer must wait for SELDO status (bit 6, SSTAT0) to be one or SELTO (bit 7, SSTAT1) to be one if the hardware selection time-out is enabled (bit 2, SXFRCTL1 is set to one), or for the software selection time-out if the hardware time-out is not enabled. This control is set to zero by the sequencer, or by a hard reset.
5	(0)	ENSELI	**Enable Selection In.** When this bit is set to one, it will allow the SCSI logic to respond to a valid Selection sequence. When selected, the SELDI status (bit 5, SSTAT0) is set to one and TARGET status (bit 7, SSTAT0) is set to one. This control is only set to zero by the sequencer when no more selections are wanted.
4	(0)	ENRSELI	**Enable Reselection In.** When this bit is set to one, it will allow the SCSI logic to respond to a valid Reselection sequence. When reselected the SELDI status (bit 5, SSTAT0) is one and TARGET status (bit 7, SSTAT0) is set to zero. This control is reset to zero by writing a zero to this bit.
3	(0)	ENAUTOATNO	**Enable Auto Attention Out.** When this bit is set to one, SCSI ATN will be asserted when a Selection sequence (ENSELO=1, TEMODEO=0) is executed. This is used when you are an Initiator and want to follow the Selection with a Message Out. SCSI ATN may be cleared by the sequencer by writing one to CLRATNO (bit 6, CLRSINT1), or by a Bus Free state on the SCSI bus. Writing a zero to this bit does not clear ATN.
2	(0)	ENAUTOATNI	**Enable Auto Attention In.** When this bit is set to one, SCSI ATN will be asserted when you are reselected by a Target (ENRSELI=1). This is used when you are an Initiator and want to follow the Reselection with a Message Out (refer to SCSI-2 Spec). SCSI ATN may be cleared by the sequencer by writing one to CLRATNO (bit 6, CLRSINT1), or by a Bus Free state on the SCSI bus. Writing a zero to this bit does not clear ATN.
1	(0)	ENAUTOATNP	**Enable Auto Attention Parity.** When this bit is set to one with ENSPCHK (bit 5, SXFRCTL1) and you are an Initiator, SCSI ATN will be asserted during information transfer in phases (Data In, Message In, Status In) if a parity error is detected on SCD[7:0] or SCD[15:8]. SCSI ATN may be cleared by the sequencer by writing one to CLRATNO (bit 6, CLRSINT1), or by a Bus Free state on the SCSI bus. Writing a zero to this bit does not clear ATN.
0	(0)	SCSIRSTO	**SCSI Reset Out.** When this bit is set to one, SCSI RESET# is asserted on the SCSI bus. It must be cleared by the sequencer with a write of 0 to this bit. This control is not gated with the Target/Initiator mode.

SCSI Transfer Control 0 (SXFRCTL0)

Type: R/W
Address: M01h, DS-01h

This register together with SXFRCTL1 are used to control the SCSI module data path.

SXFRCTL0 R/W	
7	DFON
6	DFPEXP
5	
4	CLRSTCNT
3	SPIOEN
2	SCAMEN
1	CLRCHN
0	

Bit		Name	Definition
7	(0)	DFON	**Digital Filtering On.** When this bit is set to one, digital filtering is enabled for the incoming REQ# and ACK# signals. The filter period is determined by the transfer rate stored in the SCSIRATE register and the DFPEXP bit (bit 6, SXFRCTL0) as follows:

DFON	DFPEXP	Transfer Rate	Min. Filter Period	Max. Filter Period
0	X	—	0 nsec	0 nsec
1	X	> 5 MBytes/sec	12.5 nsec	25.0 nsec
1	0	=< 5 MBytes/sec	50.0 nsec	62.5 nsec
1	1	=< 5 MBytes/sec	62.6 nsec	75.0 nsec

6	(0)	DFPEXP	**Digital Filtering Period Expanded.** When this bit is set to one, it increases the minimum filter period from 50.0 to 62.5 nsec provided the SCSI transfer rate is set to =< 5 MBytes/sec. When the SCSI transfer rate is > 5 MBytes/sec, this bit has no effect on the filter period.
5	(0)	Not Used	Always reads 0.
4	(0)	CLRSTCNT	**Clear SCSI Transfer Count.** When set to one, both the SCSI transfer counter (STCNT) and the host address counter (SHADDR) are reset to 000000h. This bit is self-clearing and need not be toggled. This bit is always read back as zero.
3	(0)	SPIOEN	**SCSI PIO Enable.** When set to one, automatic PIO is enabled on the SCSI bus. This bit must remain set for the entire PIO transfer. The individual PIO transfers are triggered by reading or writing to SCSIDATL register depending on data direction and Target/Initiator mode. Writing a zero to this bit will stop any further PIO transfers without corrupting any valid data in the SCSIDATL register. This bit may be left on even when in DMA mode since SCSIEN or SDMAEN override this bit.

(Continued)

Bit		Name	Definition
2	(0)	SCAMEN	**SCSI Configured AutoMagically Enable.** When active (=1), enables the AIC-7870 SCSI module to perform level 1 SCAM protocol with HIM driver assistance, by allowing full SCSI bus signal (BSY#, SEL#, CD#, IO#, MSG#, RST#, REQ# and SCD[7:0]#) control without going through the normal SCSI 1 or 2 protocol with devices connected to the AIC-7870 SCSI bus. When SCAMEN is active, a timer (CLKOUT in SELTIMER register) is enabled for timing events of the SCAM protocol and selection time-outs will not cause RST# to be asserted.
1	(0)	CLRCHIN	**Clear Channel In.** When set to one, the SCSI FIFO and the Synchronous REQ/ACK offset counter will be cleared. The transfer control logic will also be initialized to a reset state. The SCSI transfer counters (STCNT and SHADDR) will not be changed. This is used to initialize the channel for a transfer. This bit is self-clearing.
0	(0)	Not Used	Always reads 0.

SCSI Transfer Control 1 (SXFRCTL1)

Type: R/W
Address: M02h, DS-02h

This register together with SXFRCTL0 are used to control the SCSI module data path.

SXFRCTL1 R/W	
7	BITBUCKET
6	SWRAPEN
5	ENSPCHK
4	STIMESEL(1)
3	STIMESEL(0)
2	ENSTIMER
1	ACTNEGEN
0	STPWEN

Bit		Name	Definition
7	(0)	BITBUCKET	**SCSI Bit Bucket Mode.** When this bit is set to one, it enables the SCSI logic to read data from the SCSI bus and throw it away or supply 00h write data. No data is saved and no transfer stops occur because of SCSI FIFO full/empty conditions. This only applies while in Initiator mode.
6	(0)	SWRAPEN	**SCSI Wrap Enable.** When this bit is set to one, the STCNT register is allowed to wrap past zero to allow the transfer count to exceed a 24-bit value. The status SWRAP will be one when the wrap occurs. If it is not the last wrap, clear the SWRAP status by writing a one to CLRSWRAP control (bit 3, CLRSINT0) and wait for the next SWRAP interrupt. If it is the last wrap clear SWRAP by setting CLRSWRAP (bit 3, CLRSINT0) and clearing SWRAPEN, and then wait for the SDONE interrupt (bit 2, SSTAT0).
5	(0)	ENSPCHK	**Enable SCSI Parity Checking.** When set to one, parity checking is enabled on the SCSI bus during Selection, Reselection, and Information Transfer cycles. If a parity error is detected, SCSIPERR (bit 2, SSTAT1) is set and if ENAUTOATNP (bit 1, SCSISEQ) is set, then ATN is driven active on the SCSI bus. When set to a zero, SCSIPERR will always read as a zero.
4-3	(0)	STIMESEL[1:0]	**SCSI Time-out Selection.** These bits define the selection time-out time used by the hardware selection timer. The selection time-out timer may be monitored via SELTIMER (bits 5-0).

4 3	-	Bit
0 0	-	256 s
0 1	-	128 ms
1 0	-	64 ms
1 1	-	32 ms

2	(0)	ENSTIMER	**Enable Selection Timer.** When set to one, enables the hardware selection timer. During Selection or Reselection Out, if the timer times out, SEL will be turned off, and SELTO will be set to one in SSTAT1. If this bit is set to zero, SEL will remain on the bus until it is cleared by the sequencer.
1	(0)	ACTNEGEN	**Active Negation Enable.** When active (=1) allows the SCSI outputs (REQ#, ACK#, SCD[15:0]#, SCDPL#, SCDPH#) to perform active negation when the output is switching from asserted (=0) to deasserted condition (=1) to improve the SCSI bus signal rising transition time during any Information Transfer phase. When in 8-bit mode, SCD[15:8] and SCDPH inputs are disabled. See DIFACTNEGEN for differential type installations.
0	(0)	STPWEN	**SCSI Termination Power Enable.** When in the active state (=1) will cause output STPWCTL to be in the asserted state selected by STPWLEVEL bit in the Configuration DEVCONFIG register. STPWCTL output may be used to enable or disable the external SCSI bus termination power source. When output STPWEN is inactive the external termination power device is expected to be off or disabled. STPWEN may also be used for a general purpose control bit for external logic.

SCSI Control Signal Read Register (SCSISIGI)

Type: R
Address: M03h, DS-03h

The SCSISIGI register reads the actual state of the signals on the SCSI bus pins.

SCSISIGI R	
7	CDI
6	IOI
5	MSGI
4	ATNI
3	SELI
2	BSYI
1	REQI
0	ACKI

Bit		Name	Definition
7	(x)	CDI	Reads the state of the CD# signal on the SCSI bus.
6	(x)	IOI	Reads the state of the IO# signal on the SCSI bus.
5	(x)	MSGI	Reads the state of the MSG# signal on the SCSI bus.
4	(x)	ATNI	Reads the state of the ATN# signal on the SCSI bus.
3	(x)	SELI	Reads the state of the SEL# signal on the SCSI bus.
2	(x)	BSYI	Reads the state of the BSY# signal on the SCSI bus.
1	(x)	REQI	Reads the state of the REQ# signal on the SCSI bus.
0	(x)	ACKI	Reads the state of the ACK# signal on the SCSI bus.

SCSI Control Signal Write Register (SCSISIGO)

Type: WSCSI
Address: M03h, DS-03h

The SCSISIGO write register lets the sequencer set the state of the SCSI bus control signals. However, only those control signals appropriate to the current mode (Target, Initiator or SCAM) are enabled onto the SCSI bus. The most significant three bits (CDO, IOO, and MSGO) are used for SCSI bus phase comparison in Initiator mode. All SCSISIGO write register bits are cleared by chip reset, SCSI bus reset, or SCSI bus free.

	SCSISIGO W
7	CDO
6	IOO
5	MSGO
4	ATNO
3	SELO
2	BSYO
1	REQO
0	ACKO

Bit		Name	Definition
7	(0)	CDO	**CD Out.** If in Target mode, sets CD# on SCSI bus. If in Initiator mode, sets the state of CD# expected on the next REQ# pulse. If in SCAM mode, sets CD# on the SCSI bus.
6	(0)	IOO	**IO Out.** If in Target mode, sets IO# on SCSI bus. If in Initiator mode, sets the state of IO# expected on the next REQ# pulse. If in SCAM mode, sets IO# on the SCSI bus.
5	(0)	MSGO	**MSG Out.** If in Target mode sets MSG# on SCSI bus. If in Initiator mode, sets the state of MSG# expected on the next REQ# pulse. If in SCAM mode, sets MSG# on the SCSI bus.
4	(0)	ATNO	**ATN Out.** In Target mode, this bit is not used. In Initiator mode, writing one to this bit sets ATN# on the SCSI bus. Writing a zero to this bit has no effect. ATN# may be cleared by writing one to CLRATNO (bit 6 in CLRSINT1).
3	(0)	SELO	**SEL Out.** When set to one, asserts SEL# on the SCSI bus. Can be used to negate SEL#. If in SCAM mode, sets SEL# on the SCSI bus.
2	(0)	BSYO	**BSY Out.** When set to one, asserts BSY# on the SCSI bus. May also be used to negate BSY#. When BSYO is set to one and the DIAGLEDEN bit in the Device SBLKCTL register is not active, LED# output is asserted to indicate the AIC-7870 is connected to the SCSI bus. If in SCAM mode, sets BSY# on the SCSI bus.
1	(0)	REQO	**REQ Out.** If in Target mode, sets REQ# on the SCSI bus. It is not functional in Initiator mode. If in SCAM mode, sets REQ# on the SCSI bus.
0	(0)	ACKO	**ACK Out.** If in Initiator mode, sets ACK# on the SCSI bus. It is not functional in Target mode.

SCSI Rate (SCSIRATE)

Type: R/W
Address: M04h, DS-04h

The contents of this register determine the synchronous SCSI data transfer rate and the maximum synchronous REQ/ACK offset. An offset value of 0 in the SOFS [3:0] disables synchronous data transfers. Any offset value greater than 0 enables synchronous transfers.

SCSIRATE R/W	
7	WIDEXFER
6	SXFR(2)
5	SXFR(1)
4	SXFR(0)
3	SOFS(3)
2	SOFS(2)
1	SOFS(1)
0	SOFS(0)

Bit		Name	Definition
7	(0)	WIDEXFER	**Wide SCSI Transfer.** When SELWIDE (bit 1, SBLKCTL) and this bit are set, 16-bit transfers will take place on the SCSI bus.
6-4	(0)	SXFR(2:0)	**Synchronous SCSI Transfer Rate 2:0.** These bits select the Data phase transfer rate per the table below. Times are shown for a 40 MHz clock with clock period T.

SXFR	REQ#/ACK# Width	REQ#/ACK# Period	Rate (MHz)
000	50 nsec (2T)	100 nsec (4T)	10
001	50 nsec (2T)	125 nsec (5T)	8.0
010	50 nsec (2T)	150 nsec (6T)	6.7
011	50 nsec (2T)	175 nsec (7T)	5.7
100	100 nsec (4T)	200 nsec (8T)	5.0
101	100 nsec (4T)	225 nsec (9T)	4.4
110	100 nsec (4T)	250 nsec (10T)	4.0
111	100 nsec (4T)	275 nsec (11T)	3.6

For transfers below 3.6 MB/S use asynchronous transfer mode.

3-0	(0)	SOFS(3:0)	**SCSI Offset (3:0).** When set to 0000, the SCSI transfer mode is asynchronous. When set to any other value the transfer mode is synchronous with the indicated offset. Valid ranges (besides 0000) are 0001 through 1000 for Wide SCSI transfers and 0001 through 1111 for normal 8-bit transfers. This field only applies to Data phases. It should be set up properly per the SCSI device synchronous negotiation since the target could force a Data phase even though a different phase may be expected. Refer to the SCSI Operation Sections.

SCSI ID (SCSIID)

Type: R/W
Address: M05h, DS-05h

This register contains the devices own ID of the selected channel (OID) and the ID of the SCSI device that you want to communicate with (TID).

SCSIID R/W	
7	TID(3)
6	TID(2)
5	TID(1)
4	TID(0)
3	OID(3)
2	OID(2)
1	OID(1)
0	OID(0)

Bit		Name	Definition
7-4	(0)	TID(3:0)	**Target (Other) ID.** This is a binary representation of the other device ID on the SCSI bus during any Selection/Reselection sequence. It is the other Target ID during Selection Out (ENSELO=1, TEMODEO=0) and Reselection In (ENRSELI). It is the other Initiator ID during Selection In (ENSELI) and Reselection Out (ENSELO, TEMODEO=1). In any case, it is *the other ID.*
3-0	(0)	OID(3:0)	**Own ID.** This is a binary representation of your own device ID on the SCSI bus during any Selection/Reselection sequence. It is your own Initiator ID during Selection Out (ENSELO=1, TEMODEO=0) and Reselection In (ENRSELI). It is your own Target ID during Selection In (ENSELI) and Reselection Out (ENSELO, TEMODEO=1). In any case, it is *your own ID.*

SCSI Latched Data (SCSIDATL,[H])

Type: R/W
Address: M06/M07h, DS-06/07h

This is a read/write latch used to transfer data on the SCSI bus during Automatic or Manual SCSI PIO transfer. Bit 7 is the MSB. These registers are used in both 8-bit and 16-bit data transfer modes. In 8-bit mode, data is written to or read from SCSIDATL only. The SCSI ACK (as Initiator) or REQ (as Target) is driven active when the write or read occurs. In 16-bit mode, SCSIDATH should be written to or read from before SCSIDATL. Direct access to the SCSI bus is provided via read of SCSIBUS register. The initial read value after a chip reset is unknown. Valid data will be loaded after the first REQ#/ACK# In.

SCSIDATL R/W		SCSIDATH R/W	
7	DB(07)	7	DB(15)
6	DB(06)	6	DB(14)
5	DB(05)	5	DB(13)
4	DB(04)	4	DB(12)
3	DB(03)	3	DB(11)
2	DB(02)	2	DB(10)
1	DB(01)	1	DB(09)
0	DB(00)	0	DB(08)

SCSI Transfer Count (STCNT[n])

Type: R/W
Address: M08/M09/M0Ah, DS-08/09/0Ah

These registers contain the DMA or Automatic PIO byte transfer count on the SCSI interface. STCNT0 is the least significant byte, STCNT1 is the mid byte, and STCNT2 is the most significant byte. If Initiator mode is enabled, it is loaded with the number of ACKs to send out on the SCSI bus. If Target mode is enabled, it is loaded with the number of REQs to send out on the SCSI bus. In Automatic PIO mode, STCNT is used as a counter only and need not be initialized to transfer data with Automatic PIO handshakes. Loading 000000h will give a byte transfer count of 16777216 decimal (16M Hex) if SWRAPEN (bit 6, SXFRCTL1) is set, and a transfer count of 0 if SWRAPEN is cleared.

The counter counts down by one when a SCSI byte is transferred. When sending data to the bus, a byte is considered transferred when the appropriate handshake signal is received (REQ#/ACK#). When receiving data from the bus, a byte is considered transferred when it has been written to the data FIFO. Two counters are maintained for this purpose, and the sense of DIRECTION (bit 2, DFCNTRL) which is latched by the last positive edge of SCSIEN or SDMAEN determines which one is used.

SDONE (bit 2, SSTAT0) is set when the transfer count is zero, SWRAPEN is zero, and SDMAEN is one. SWRAP is set when SWRAPEN is set and the transfer counter counts from 000000h to FFFFFFh. SWRAP should then be cleared via CLRSWRAP (bit 3, CLRSINT0) before the next wrap (that time is 16M times the SCSI bus transfer period). The sequencer must keep track of the number of wraps. The count is set to zero on a chip reset.

STCNT0 R/W		STCNT1 R/W		STCNT2 R/W	
7	STCNT(07)	7	STCNT(15)	7	STCNT(23)
6	STCNT(06)	6	STCNT(14)	6	STCNT(22)
5	STCNT(05)	5	STCNT(13)	5	STCNT(21)
4	STCNT(04)	4	STCNT(12)	4	STCNT(20)
3	STCNT(03)	3	STCNT(11)	3	STCNT(19)
2	STCNT(02)	2	STCNT(10)	2	STCNT(18)
1	STCNT(01)	1	STCNT(09)	1	STCNT(17)
0	STCNT(00)	0	STCNT(08)	0	STCNT(16)

Clear SCSI Interrupt 0 (CLRSINT0)

Type: W
Address: M0Bh, DS-0Bh

Writing a one to a bit in this register clears the associated SCSI interrupt bit in SSTAT0. Writing a zero to any bit in this register will have no effect. Each bit is self-clearing and writing a zero to any bit in this register will have no effect.

CLRSINT0 W	
7	
6	CLRSELDO
5	CLRSELDI
4	CLRSELINGO
3	CLRSWRAP
2	
1	CLRSPIORDY
0	

Bit		Name	Definition
7	(0)	Not Used	
6	(0)	CLRSELDO	Clears the SELDO interrupt and status.
5	(0)	CLRSELDI	Clears the SELDI interrupt and status.
4	(0)	CLRSELINGO	Clears the SELINGO interrupt and status.
3	(0)	CLRSWRAP	Clears SWRAP interrupt and status.
2	(0)	Not Used	
1	(0)	CLRSPIORDY	Clears SPIORDY interrupt and status.
0	(0)	Not Used	

SCSI Status 0 (SSTAT0)

Type: R
Address: M0Bh, DS-0Bh

This register contains the status of SCSI interrupt bits. Any status bit may be read at any time whether or not it has been enabled in SIMODE0. If an interrupt bit is enabled and set to one, the SCSIINT interrupt line will be driven to the active state (except TARGET which is a status bit only).

SSTAT0 R	
7	TARGET
6	SELDO
5	SELDI
4	SELINGO
3	SWRAP
2	SDONE
1	SPIORDY
0	DMADONE

Bit		Name	Definition
7	(0)	TARGET	**Target.** When this bit is one, it signals that you are a Target. It is only valid after a Selection or Reselection is completed and before bus free.
6	(0)	SELDO	**Select Out.** This bit is a one when you have successfully done a Select Out or Reselect Out sequence. TARGET will decide whether it was Select (TARGET=0) or Reselect (TARGET=1). It is cleared by a Bus Free condition or by setting CLRSELDO (bit 6,CLRSINT0). Interrupts may be enabled by setting ENSELDO (bit 6, SIMODE0)) to one. SELDO is a wake up condition, when enabled, for the sequencer SLEEP mode.
5	(0)	SELDI	**Select In.** This bit is a one when you have been selected or reselected. If TARGET is a one, you have been selected, and if zero, you have been reselected. It is cleared by a Bus Free condition or by setting CLRSELDI (bit 5, CLRSINT0). Interrupts may be enabled by setting ENSELDI (bit 5, SIMOED0) to one. SELDI is a wake up condition, when enabled, for the sequencer Sleep mode.
4	(0)	SELINGO	**Selecting Out.** After successful arbitration, this bit is set to one when you start the attempt to select or reselect another device. This interrupt is used to start looking for SELDO or bus time-out. When a successful selection has been completed (SELDO is one), this bit will be cleared. This bit may also be cleared by setting CLRSELINGO (bit 4, CLRSINT0).
3	(0)	SWRAP	**SCSI Count Wrap.** This bit is a one when STCNT counts from 000000h to FFFFFFh and SWRAPEN is set. SWRAPEN (bit 6, SZFRCTL1) must be set to enable the counter to count down past 000000h. SWRAP will also be set if SDMAEN is set to one, and both STCNT is equal to zero and SWRAPEN is one. This bit may be cleared by setting CLRSWRAP (bit 3, CLRSINT0).
2	(0)	SDONE	**SCSI Done.** This bit is set to one when STCNT=000000h, SWRAPEN is cleared, SDMAEN or SPIOEN is set, and the last transfer has completed. It may also be set when SPIORDY is set. This bit may be cleared by writing a nonzero count to STCNT, setting SRAPEN, clearing SDMAEN, or if set by SPIORDY, it may be cleared by clearing SPIORDY, SCSIEN (bit 5, DFCNTRL) should be cleared before this bit is cleared in Target mode to prevent false transfers.
1	(0)	SPIORDY	**SCSI PIO Ready.** When this bit is one, the Automatic SCSI PIO function has been enabled and data is ready or needed by the SCSI data transfer logic. As an Initiator, this bit is set to one on the leading edge of REQ. In Target mode, this bit is set on the leading edge of ACK. In both Initiator and Target mode, during a transfer to SCSI, the bit is cleared on a write to SCSIDATL. During a transfer from SCSI, it is cleared on a read from SCSIDATL. It may also be cleared by setting CLRSPIORDY (bit 1, CLRSINT0) or by clearing SPIOEN (bit 3, SXFRCTL0). This bit sets SDONE when it is set.
0	(0)	DMADONE	**DMA Done.** This bit is the logical AND of HDONE (bit 3, DFSTATUS) and SDONE. It indicates the current transfer has completely finished. DMADONE is a wake up condition, when enabled, for the sequencer Sleep mode.

Clear SCSI Interrupt 1 (CLRSINT1)

Type: W
Address: M0Ch, DS-0Ch

Writing a one to a bit in this register clears the associated SCSI interrupt bit in SSTAT1. Each bit is self-clearing and writing a zero to any bit in this register will have no effect.

CLRSINT1 W	
7	CLRSELTIMO
6	CLRATNO
5	CLRSCSIRSTI
4	
3	CLRBUSFREE
2	CLRSCSIPERR
1	CLRPHASECHG
0	CLRREQINIT

Bit		Name	Definition
7	(0)	CLRSELTIMO	Clears the SELTO interrupt and status.
6	(0)	CLRATNO	In Initiator mode, clears the SCSI ATN bit if set by the sequencer or any automatic mode. ATN is also cleared by the Bus Free condition. In Target mode, clears ATNTARG interrupt and status.
5	(0)	CLRSCSIRSTI	Clears SCSIRSTI interrupt and status.
4	(0)	Not Used	
3	(0)	CLRBUSFREE	Clears BUSFREE interrupt and status.
2	(0)	CLRSCSIPERR	Clears SCSIPERR interrupt and status.
1	(0)	CLRPHASECHG	Clears PHASECHG interrupt and status.
0	(0)	CLRREQINIT	Clears REQINIT interrupt and status.

SCSI Status 1 (SSTAT1)

Type: R
Address: M0Ch, DS-0Ch

This register contains the status of SCSI interrupt bits. Any interrupt bit may be read at any time whether or not it has been enabled in SIMODE1. If enabled and set to one, it will cause the interrupt line to go to the active state. All are cleared by the corresponding bits in CLRSINT1 register (except PHASEMIS and SCSIPERR).

	SSTAT1 R
7	SELTO
6	ATNTARG
5	SCSIRSTI
4	PHASEMIS
3	BUSFREE
2	SCSIPERR
1	PHASECHG
0	REQINIT

Bit		Name	Definition
7	(0)	SELTO	**Selection Time-out.** This bit is set when the hardware selection timer is enabled and a Selection or Reselection time-out occurs. The timer is enabled by setting ENSTIMER (bit 2, SXFRCTL1) to one along with the time-out value in bits 3 and 4. The bit is cleared by setting CLRSELTIMO (bit 7, CLRSINT1) to one.
6	(0)	ATNTARG	**Attention Target.** This bit is set to one when you are a Target and the Initiator asserts ATN. It is latched and will be cleared when ATN is deasserted or when CLRATNO (bit 6, CLRSINT1) is set.
5	(0)	SCSIRSTI	**SCSI Reset In.**This bit is set to one when another device asserts RST# on the SCSI bus. It remains set until cleared by writing a one to CLRSCSIRSTI (bit 5, CLRSINT1).
4	(0)	PHASEMIS	**Phase Mismatch.** Initiator mode only. This bit is set to one when the last phase on the SCSI bus sampled by REQ does not match the expected phase which is in the SCSISIGO register. It is qualified with REQINIT (bit 0, SSTAT1) and is cleared by writing the matching phase in SCSISIGO or by clearing REQINIT. PHASEMIS is a wake up condition, when enabled, for the sequencer Sleep mode.
3	(0)	BUSFREE	**Bus Free.** This bit is set to one when the BSY and SEL signals have been negated on the SCSI bus for 400ns. This signal is latched and may be cleared by setting CLRBUSFREE in CLRSINT1 to one. This bit will be initially set to zero, but will reflect the state of the SCSI bus after 400ns.
2	(0)	SCSIPERR	**SCSI Parity Error.** This bit is set to one when a parity error is detected on the incoming SCSI information transfer. Parity is sampled on the leading edge of REQ if in Initiator mode or the leading edge of ACK if in Target mode. If WIDEXFER in SXFRCTL0 is set, then parity will be checked on the upper byte of the SCSI bus during the Data phase only. If parity is enabled (ENSPCHK in SXFRCTL1 is set to one), a parity error will cause a one to be latched in this bit until cleared by writing one to CLRSCSIPERR in CLRSINT1. After writing to CLRSCSIPERR, this bit reflects the status of the parity of the last byte transferred on the bus. If ENSPCHK is set to zero, this bit will always be read as a zero.
1	(0)	PHASECHG	**SCSI Phase Change.** This bit is set to one when the phase on the SCSI bus changes to a phase that does not match the expected phase which is in the SCSISIGO register. It is not qualified with REQ. It can be cleared by writing a one to CLRPHASECHG in CLRSINT1.
0	(0)	REQINIT	**Request Initialize.** Initiator mode only. This bit is set to one in asynchronous transfer mode on the leading edge of a REQ being asserted on the SCSI bus or when in synchronous transfer mode and the offset count is greater than one. In asynchronous transfer mode it is cleared on the leading edge of any ACK sent out on the SCSI bus, or when in synchronous transfer mode the offset count is equal to zero, or with CLRREQINIT (bit 0, CLRSINT1).

SCSI Status 2 (SSTAT2)

Type: R
Address: M0Dh, DS-0Dh

These bits are read only and give the status of the SCSI FIFOs.

	SSTAT2 R
7	OVERRUN
6	
5	
4	SFCNT(4)
3	SFCNT(3)
2	SFCNT(2)
1	SFCNT(1)
0	SFCNT(0)

Bit		Name	Definition
7	(0)	OVERRUN	**Overrun Detect.** During synchronous transfers, this bit is set to one when an offset overrun is detected in the read direction for Initiator mode only. An offset overrun is defined as the situation where the maximum offset has been reached and another REQ is detected before an ACK is asserted on the SCSI bus. This status bit is cleared with CLRCHN (bit 1, SXFRCTL0).
6-5	(0)	Not Used	Always reads 0.
4-0	(0)	SFCNT(4:0)	**SCSI FIFO Byte Count.** Shows the count of bytes in the SCSI FIFO. A value of 0h means the SCSI FIFO is empty. A count of 16 means the SCSI FIFO is full.

SCSI Status 3 (SSTAT3)

Type: R
Address: M0Eh, DS-0Eh

This register is the status of the current Synchronous SCSI Information Transfer phase.

SSTAT3 R	
7	SCSICNT(3)
6	SCSICNT(2)
5	SCSICNT(1)
4	SCSICNT(0)
3	OFFCNT(3)
2	OFFCNT(2)
1	OFFCNT(1)
0	OFFCNT(0)

Bit		Name	Definition
7-4	(0)	SCSICNT(3:0)	**SCSI Count (3:0).** Gives the difference between what the offset count says is in the SCSI FIFO and what the OFFCNT says is in the SCSI FIFO. Used by hardware to prevent SCSI FIFO overrun. Do not read this counter unless transfers are stopped.
3-0	(0)	OFFCNT(3:0)	**SCSI Offset Count (3:0).** Gives the current SCSI offset count. Do not read this counter unless transfers are stopped.

SCSI Test Control (SCSITEST)

Type: R/W
Address: M0Fh, DS-0Fh

This register is used to force test modes in the SCSI Module Logic.

SCSITEST R/W	
7	
6	
5	
4	DATALOOP
3	
2	RQAKCNT
1	CNTRTEST
0	CMODE

Bit		Name	Definition
7-5	(0)	Not Used	Always read 0.
4	(0)	DATALOOP	**Data Loop.** When active (=1) enables data to be transferred from the data FIFO through the SCSI FIFO to the SCSI output data registers (SCSIBUSH and SCSIBUSL) and then read out by the sequencer or software driver. Alternately, data may be written to the SCSI input data registers (SCSIDATH and SCSIDATL) by the sequencer or the software driver for transfer to the SCSI FIFO then to the data FIFO. In either direction the SCSI bus is isolated and unaware of this activity.
3	(0)	Not Used	Always reads 0.
2	(0)	RQAKCNT	**Request/Acknowledge Count.** This bit inverts the meaning of the DIRECTION input signal for STCNT. If DIRECTION =1 (write) and this bit is set, then reading the STCNT register will access the STCNT counter instead of RQAKCNT. If DIRECTION=0 (read) and this bit is set, then reading the STCNT register will access the RQAKCNT counter instead of STCNT.
1	(0)	CNTRTEST	**Counter Test.** When set to one, the SCSI transfer counter STCNT and the Selection time-out counter SELTIMER are put into a mode where they count down at the input clock rate, and the SCSI host address counter SHADDR is put into a mode where it counts up at the input clock rate.
0	(0)	CMODE	**Carry Mode.** When set to one, forces a stage-to-stage carry true in STCNT, SHADDR, and SELTIMER. During the Transfer Count test, the counter contents can be monitored by reading the desired stage.

SCSI Interrupt Mode 0 (SIMODE0)

Type: R/W
Address: M10h, DS-10h

Setting any bit will enable the corresponding function in SSTAT0 to interrupt via the IRQA# pin.

SIMODE0 R/W	
7	
6	ENSELDO
5	ENSELDI
4	ENSELINGO
3	ENSWRAP
2	ENSDONE
1	ENSPIORDY
0	ENDMADONE

Bit		Name	Definition
7	(0)	Not Used	Always reads 0.
6	(0)	ENSELDO	Enables SELDO status to assert SCSIINT.
5	(0)	ENSELDI	Enables SELDI status to assert SCSIINT.
4	(0)	ENSELINGO	Enables SELINGO status to assert SCSIINT.
3	(0)	ENSWRAP	Enables SWRAP status to assert SCSIINT.
2	(0)	ENSDONE	Enables SDONE status to assert SCSIINT.
1	(0)	ENSPIORDY	Enables SPIORDY status to assert SCSIINT.
0	(0)	ENDMADONE	Enables DMADONE status to assert SCSIINT.

SCSI Interrupt Mode 1 (SIMODE1)

Type: R/W
Address: M11h, DS-11h

Setting any bit will enable the corresponding function in SIMODE1 to interrupt via the IRQA# pin.

SIMODE1 R/W	
7	ENSELTIMO
6	ENATNTARG
5	ENSCSIRST
4	ENPHASEMIS
3	ENBUSFREE
2	ENSCSIPERR
1	ENPHASECHG
0	ENREQINIT

Bit		Name	Definition
7	(0)	ENSELTIMO	Enables the SELTO status to assert SCSIINT.
6	(0)	ENATNTARG	Enables ATNTARG status to assert SCSIINT.
5	(0)	ENSCSIRST	Enables SCSIRST status to assert SCSIINT.
4	(0)	ENPHASEMIS	Enables PHASEMIS status to assert SCSIINT.
3	(0)	ENBUSFREE	Enables BUSFREE status to assert SCSIINT.
2	(0)	ENSCSIPERR	Enables the latched SCSIPERR status to assert SCSIINT.
1	(0)	ENPHASECHG	Enables PHASECHG status to assert SCSIINT.
0	(0)	ENREQINIT	Enables REQINIT status to assert SCSIINT.

SCSI Data Bus (SCSIBUSL, [H])

Type: R
Address: M12/M13h, DS-12/13h

This register reads data on the SCSI data bus directly. Data is gated from the SCSI data bus to the internal data bus, it is not latched in the SCSI Module.

SCSIBUSL R		SCSIBUSH R	
7	SDB(07)	7	SDB(15)
6	SDB(06)	6	SDB(14)
5	SDB(05)	5	SDB(13)
4	SDB(04)	4	SDB(12)
3	SDB(03)	3	SDB(11)
2	SDB(02)	2	SDB(10)
1	SDB(01)	1	SDB(09)
0	SDB(00)	0	SDB(08)

SCSI Count Host Address (SHADDR[0:3])

Type: R
Address: M14/M15/M16/M17h, DS-14/15/16/17h

These registers are written when the Host Address registers (M88h - M8Bh) are written and HADDLDSEL[1:0] value is 0. They are counted up in the same manner that STCNT is counted down. This value is saved when the Save Data Pointers message is received. The value is set to zero when reset.

SHADDR0 R	SHADDR1 R	SHADDR2 R	SHADDR3 R
7 SHADDR(07)	7 SHADDR(15)	7 SHADDR(23)	7 SHADDR(31)
6 SHADDR(06)	6 SHADDR(14)	6 SHADDR(22)	6 SHADDR(30)
5 SHADDR(05)	5 SHADDR(13)	5 SHADDR(21)	5 SHADDR(29)
4 SHADDR(04)	4 SHADDR(12)	4 SHADDR(20)	4 SHADDR(28)
3 SHADDR(03)	3 SHADDR(11)	3 SHADDR(19)	3 SHADDR(27)
2 SHADDR(02)	2 SHADDR(10)	2 SHADDR(18)	2 SHADDR(26)
1 SHADDR(01)	1 SHADDR(09)	1 SHADDR(17)	1 SHADDR(25)
0 SHADDR(00)	0 SHADDR(08)	0 SHADDR(16)	0 SHADDR(24)

Selection Time-out Timer (SELTIMER)

Type: R
Address: M18h, DS-18h

This register is used to monitor the state of the hardware selection time-out timer and the SCAM timer.

SELTIMER R/W	
7	CLKOUT
6	
5	STAGE 6
4	STAGE 5
3	STAGE 4
2	STAGE 3
1	STAGE 2
0	STAGE 1

Bit		Name	Definition
7	(0)	CLKOUT	**Clock Out.** A timer provided for the HIM driver to use when performing SCAM protocol. It is enabled to run whenever SCAMEN (SXFRCTL0 register) is active. The frequency is determined by the CLKIN source (40 MHz) divided to provide a 102.40 µs period.
6	(0)	Not Used	Always reads 0.
5	(0)	STAGE 6	(divide by 2, output)
4	(0)	STAGE 5	(divide by 2, output)
3	(0)	STAGE 4	(divide by 2, output)
2	(0)	STAGE 3	(divide by 10, output)
1	(0)	STAGE 2	(divide by 256, output)
0	(0)	STAGE 1	(divide by 256, output)

Selection/Reselection ID (SELID)

Type: R
Address: M19h, DS-19h

This register contains the SCSI ID of the selecting or reselecting device which was asserted during the last Selection/Reselection SCSI bus phase. Hardware will remove the device ID and decode the remaining ID. After a Selection/Reselection in has taken place, the ID may be read from this register to determine the ID of the device which initiated the Selection/Reselection. If a selection occurred by a SCSI device which did not set its own ID, then ONEBIT will be set to indicate that condition. If ONEBIT is zero, then 2 bits were active on the SCSI bus.

SELID R	
7	SELID(3)
6	SELID(2)
5	SELID(1)
4	SELID(0)
3	ONEBIT
2	
1	
0	

Bit		Name	Definition
7-4	(0)	SELID(3:0)	**Selection ID.** This is the ID of the selecting or reselecting SCSI device.
3	(0)	ONEBIT	This bit is set when only one bit is detected on the lower 8 bits (if SELWIDE=0) during Selection. It is zero when 2 bits are detected during a Selection.
2-0	(0)	Not Used	Always reads 0.

SCSI Block Control (SBLKCTL)

Type: R/W
Address: M1Fh, DS-1Fh

This register controls the hardware selection options outside of the SCSI cells. This control includes address decodes and data multiplexing.

SBLKCTL R/W	
7	DIAGLEDEN
6	DIAGLEDON
5	AUTOFLUSHDIS
4	
3	SELBUSB
2	
1	SELWIDE
0	

Bit		Name	Definition
7	(1)	DIAGLEDEN	**Diagnostic LED Enable.** When this bit is active (=1), the LED# assertion output state reflects the state of the DIAGLEDON bit, except during and after RST# assertion. During and immediately after RST# assertion when the LED output is used for clocking in IDDAT, see section on IDDAT. When DIAGLEDEN bit is not active (normal operation), the LED# assertion output state reflects the OR result of active bits SELINGO, SELDI and SELDO in the SSTAT0 register.
6	(1)	DIAGLEDON	**Diagnostic LED On.** When this bit is active (=1) and DIAGLEDEN is active, the LED# output will be asserted. When this bit is inactive and DIAGLEDEN is active, the LED# output will not be asserted. When DIAGLEDEN is inactive, the state of DIAGLEDON has no effect on the assertion state of LED#. The use of DIAGLEDEN and DIAGLEDON provide a diagnostic capability where the assertion state of LED# may be used for an external trigger for occurrence of internal events.
5	(0)	AUTOFLUSHDIS	**Auto Flush Disable.** When active (=1) prevents the SCSI hardware activated autoflush action of the data FIFO to the PCI bus system memory due to SCSI PHASEMIS or SDONE becoming active. FIFOFLUSH in the DFCNTRL register may then be used to activate the flush action as needed.
4	(0)	Not Used	Always reads 0.
3	(0)	SELBUSB	This bit always read as 0 and writing to this bit has no effect.
2	(0)	Not Used	Always reads 0.

(Continued)

Bit		Name	Definition
1	(*)	SELWIDE	**Select Wide.** When this bit is set, the internals of the device are configured for one 16-bit Wide SCSI channel with the control lines for the one 8-bit channel (not wide selection) used for phase detection and transfer control of data (SCD[15:8]# with SCDPH# and SCD[7:0]# with SCDPL#). It is expected that the external bus is Wide when SELWIDE is enabled. When this bit is cleared, the device is configured for one 8-bit channel using only SCD[7:0]# and SCDPL#, with SCD[15:08]# and SCDPH# not used. When the device is configured for one 8-bit channel the I/O inputs SCD[15:08]# and SCDPH# are internally terminated so that external termination is not required. Note: (*) SELWIDE may be initialized at chip reset to indicate the hardware connection to the WIDEPS# input. When WIDEPS# is deasserted, SELWIDE will be cleared. When WIDEPS# is asserted, it indicates a wide connection and SELWIDE will be set. After this chip reset initialization SELWIDE may be written to either state when desired.
0	(0)	Not Used	Always reads 0.

Sequencer Device Space Register Definition

Sequencer Control (SEQCTL)

Type: R/W
Address: M60h, DS-60h

SEQCTL R/W	
7	PERRORDIS
6	PAUSEDIS
5	FAILDIS
4	FASTMODE
3	BRKADRINTEN
2	STEP
1	SEQRESET
0	LOADRAM

Bit		Name	Definition
7	(1)	PERRORDIS	**Parity Error Disable.** When cleared, allows sequencer RAM parity errors to be detected. When set, disables Parity Error detection.
6	(0)	PAUSEDIS	**Pause Disable.** If set, disables the Pause function when PAUSE (bit 2, HCNTRL) is set. Pause due to interrupts or error conditions is still enabled. SCSI interrupts, an illegal opcode interrupt, or a sequencer RAM parity error interrupt resets this bit. Host software may not write to this bit.
5	(0)	FAILDIS	**Fail Disable.** If set, disables parity and PCI error status interrupt feature. If cleared, the detection of an parity or PCI error status will cause a BRKADRINT to occur and will latch-pause the sequencer.
4	(1)	FASTMODE	**Fast Mode.** If set to one, then the clock to the sequencer is divided by four from the input clock. If set to zero, then the clock is divided by five.
3	(0)	BRKADRINTEN	**Break Address Interrupt Enable.** When set, the breakpoint status is enabled to drive the interrupt pin. When cleared and the breakpoint is enabled (clear BRKDIS in BRKADDR1), BRKADRINT (bit 3, INTSTAT) will be set, but IRQA# will not be asserted.
2	(0)	STEP	**Step.** When set, the sequencer will execute one instruction and then self-pause. The sequencer should be paused before using this bit. This bit will remain set until cleared by the software driver. Multiple single steps may be done by clearing PAUSE (bit 2, HCNTRL) multiple times.
1	(0)	SEQRESET	**Sequencer RAM Address Reset.** When set, the address pointer for the sequencer RAM is cleared and program execution starts at location zero. This bit is self-clearing. The sequencer must be paused before setting this bit.
0	(0)	LOADRAM	**Load Sequencer RAM.** When set, allows the sequencer RAM to be loaded or read by writing or reading a series of bytes to or from SEQRAM. This bit should be cleared for normal operation. When switching between reads and writes, this bit should be first cleared, then set again.

Sequencer RAM Data (SEQRAM)

Type: R/W
Address: M61h, DS-61h

This register is a port to the sequencer RAM area. The RAM may be loaded by first writing a starting address in SEQADDR0 and SEQADDR1, then sending a stream of bytes to this register. The byte ordering should be from the least significant byte to the most significant. The address will be auto-incremented after the most significant byte is written to facilitate loading the program.

SEQRAM R/W	
7	SEQRAM(7)
6	SEQRAM(6)
5	SEQRAM(5)
4	SEQRAM(4)
3	SEQRAM(3)
2	SEQRAM(2)
1	SEQRAM(1)
0	SEQRAM(0)

Sequencer RAM Address (SEQADDR[1:0])

Type: R/W
Address: M62/63h, DS-62/63h

These registers contain the address of the sequencer RAM that will be executed on the next clock edge. They may be written to for the purpose of changing the execution location after first pausing the sequencer. They are also used to specify the starting location when loading the program. The address will automatically increment while loading the program after every fourth byte. The fourth byte index is cleared when this register is written. SEQADDR[1:0] are cleared to 0h by RST# or CHIPRST. See *Sequencer Loading* on page 5-10.

SEQADDR0 R/W		SEQADDR1 R/W	
7	SEQADDR(07)	7	
6	SEQADDR(06)	6	
5	SEQADDR(05)	5	RSVD
4	SEQADDR(04)	4	RSVD
3	SEQADDR(03)	3	RSVD
2	SEQADDR(02)	2	RSVD
1	SEQADDR(01)	1	RSVD
0	SEQADDR(00)	0	SEQADDR(08)

Accumulator (ACCUM)

Type: R/W
Address: M64h, DS-64h

This register is a temporary holding place for arithmetic or logical operations.

	ACCUM R/W
7	ACCUM(7)
6	ACCUM(6)
5	ACCUM(5)
4	ACCUM(4)
3	ACCUM(3)
2	ACCUM(2)
1	ACCUM(1)
0	ACCUM(0)

Source Index Register (SINDEX)

Type: R/W
Address: M65h, DS-65h

This register is a temporary holding register or may be used as an indirect address for source operands for some ALU operations.

	SINDEX R/W
7	SINDEX(7)
6	SINDEX(6)
5	SINDEX(5)
4	SINDEX(4)
3	SINDEX(3)
2	SINDEX(2)
1	SINDEX(1)
0	SINDEX(0)

Destination Index Register (DINDEX)

Type: R/W
Address: M66h, DS-66h

This register is a temporary holding register or may be used as an indirect address for destination operands for some ALU operations.

	DINDEX R/W
7	DINDEX(7)
6	DINDEX(6)
5	DINDEX(5)
4	DINDEX(4)
3	DINDEX(3)
2	DINDEX(2)
1	DINDEX(1)
0	DINDEX(0)

Break Address 0 (BRKADDR0)

Type: R/W
Address: M67h, DS-67h

This register is used for diagnostic purposes to halt the sequencer at a specific address. It is loaded with the lower byte of the break address. BRKADDR0 is cleared to 0h by RST# or CHIPRST. See *Breakpoint* on page 5-11.

	BRKADDR0 R/W
7	BRKADDR(07)
6	BRKADDR(06)
5	BRKADDR(05)
4	BRKADDR(04)
3	BRKADDR(03)
2	BRKADDR(02)
1	BRKADDR(01)
0	BRKADDR(00)

Break Address 1 (BRKADDR1)

Type: R/W
Address: M68h, DS-68h

This register is used for diagnostic purposes to halt the sequencer at a specific address. It is loaded with the upper byte of the break address. In addition, bit 7 is a break condition disable. See *Breakpoint* on page 5-11.

BRKADDR1 R/W	
7	BRKDIS
6	
5	RSVD
4	RSVD
3	RSVD
2	RSVD
1	RSVD
0	BRKADDR(08)

Bit		Name	Definition
7	(1)	BRKDIS	**Break Disable.** When set, it disables the break on compare feature of the sequencer. When cleared, this feature is enabled.
6	(0)	Not Used	Always reads 0.
5	(0)	RSVD	
4	(0)	RSVD	
3	(0)	RSVD	
2	(0)	RSVD	
1	(0)	RSVD	
0	(0)	BRKADDR(08)	**Break Address.** Address bit 08 used for comparison with BRKADDR0.

All Ones (ALLONES)

Type: R
Address: M69h, DS-69h

This port returns all ones when read. It may be used for certain logical and arithmetic functions.

All Zeros (ALLZEROS)

Type: R
Address: M6Ah, DS-6Ah

This port returns all zeros when read. It may be used for certain logical and arithmetic functions.

No Destination (NONE)

Type: W
Address: M6Ah, DS-6Ah

When this port is selected as the destination, no change will be made to any location.

Flags (FLAGS)

Type: R
Address: M6Bh, DS-6Bh

This register returns the flag values.

FLAGS R	
7	
6	
5	
4	
3	
2	
1	ZERO
0	CARRY

Source Index Indirect (SINDIR)

Type: R
Address: DS-6Ch

When a transfer is done from this port, the contents of SINDEX is used as the source address. After the transfer is completed, SINDEX is incremented. This register is usable only by the sequencer.

SINDIR R	
	CONTENTS POINTED TO BY SINDEX

Destination Index Indirect (DINDIR)

Type: W
Address: DS-6Dh

When a transfer is done to this port, the contents of DINDEX is used as the destination address. After the transfer is completed, DINDEX is incremented. This register is usable only by the sequencer.

DINDIR W	
	CONTENTS POINTED TO BY DINDEX

Function1 (FUNCTION1)

Type: R/W
Address: M6Eh, DS-6Eh

This register provides a specific function for use by the sequencer code to minimize the number of instructions. Data is written to FUNCT1 with valid data in bits 6-4. This octal value is decoded into a 1 of 8 bit position. A value of 0 gives a 1 in bit position 0, a value of 1 gives a 1 in bit position 1, etc. with all other bit positions having a value of 0.

FUNCTION1 W		FUNCTION1 R
7		1 of 8 decoded value of FUN1DAT(2:0)
6	FUN1DAT(2)	
5	FUN1DAT(1)	
4	FUN1DAT(0)	
3		
2		
1		
0		

Stack (STACK)

Type: R
Address: M6Fh, DS-6Fh

The contents of the stack is reported one byte at a time starting from the last location pushed on the stack until all entries are reported. The stack entries are reported by consecutive reads alternating low byte then high byte. Location 0 points to the last pushed entry, location 1 points to the entry pushed before that, etc. The sequence of bytes returned is as follows:

Byte Number	Stack Location	Low/High
0	0	Low
1	0	High
2	1	Low
3	1	High
4	2	Low
5	2	High
6	3	Low
7	3	High

The stack pointer will be incremented after a read of the high byte, therefore eight reads must be made in order to restore the location of the stack pointer to the original value if it is intended to continue program execution.

STACK R	
7	STACK(7)
6	STACK(6)
5	STACK(5)
4	STACK(4)
3	STACK(3)
2	STACK(2)
1	STACK(1)
0	STACK(0)

Configuration Space Register Definition

Registers in the Configuration Space (CS) may only be written to and/or read from the external interface when IDSEL is asserted AD[1:0]=0 and the PCI command on CBE[3:0]#= CFRC or CFWC. Selected CS register bytes are readable from AIC-7870 Device register space for access by the driver without entering Configuration space. The CS registers follow the PCI-32 bus format requirements and also provide control for some AIC-7870 located die requirements. All bytes of CS 32-bit registers (N18h:N2Fh, N34h:N38h, N41h:7Fh) are reserved and return 0h when read.

In addition, the following CS register bytes are reserved and return 0h when read (N0Ch byte 3, N40h byte 3-2). All reserved CS registers, bytes and bits ignore data when written to. All writable CS register bits are forced to the initialized state when RST# is asserted.

Vendor Identification (VENDOR ID[1:0])

Type: R
Address: N00h bits 15-0 (dual access, see Device Register space)

VENDOR ID[1:0]. The PCI vendor identification registers contain product information for use by the host in initialization and configuration of the system. The vendor ID contains two bytes of a compressed bit representation (9004h) of the vendor ID. The vendor ID characters for the AIC-7870 are

- the first ID character = A
- the second ID character = D
- the third ID character = P

VENDOR ID[1:0] may be read at any time in Configuration space, and may be read in Device space, without consideration of the state of the PAUSE bit, only when bit 0 or bit 1 is set in the Configuration Command register.

VENDID1 R		VENDID0 R	
15	VID15=1	7	VID07=0
14	VID14=0	6	VID06=0
13	VID13=0	5	VID05=0
12	VID12=1	4	VID04=0
11	VID11=0	3	VID03=0
10	VID10=0	2	VID02=1
09	VID09=0	1	VID01=0
08	VID08=0	0	VID00=0

Bit		Name	Definition
15	(1)	VID15	Always reads 1, 2nd vendor ID character
14	(0)	VID14	Always reads 0, 2nd vendor ID character
13	(0)	VID13	Always reads 0, 2nd vendor ID character LSB.
12	(1)	VID12	Always reads 1, 3rd vendor ID character
11	(0)	VID11	Always reads 0, 3rd vendor ID character
10	(0)	VID10	Always reads 0, 3rd vendor ID character
09	(0)	VID09	Always reads 0, 3rd vendor ID character.
08	(0)	VID08	Always reads 0, 3rd vendor ID character LSB.
07	(0)	VID07	Always reads 0, (fill bit).
06	(0)	VID06	Always reads 0, 1st vendor ID character MSB.
05	(0)	VID05	Always reads 0, 1st vendor ID character.
04	(0)	VID04	Always reads 0, 1st vendor ID character.
03	(0)	VID03	Always reads 0, 1st vendor ID character.
02	(1)	VID02	Always reads 1, 1st vendor ID character LSB.
01	(0)	VID01	Always reads 0, 2nd vendor ID character MSB.
00	(0)	VID00	Always reads 0, 2nd vendor ID character.

Device Identification (DEVICE ID[1:0])

Type: R
Address: N00h bits31:16 (dual access, see Device Register space)

DEVICE ID[1:0.] The PCI device identification registers contain product information for use by the host in initialization and configuration of the system. The two device ID bytes contain an Adaptec product code. The product code for the AIC-7870 is 78h for DEVICEID0 and 70 for DEVICEID1.

DEVICE ID[1:0] may be read at any time in Configuration space, and may be read in AIC-7870 Device register space without consideration of the state of the PAUSE bit, only when bit 0 or bit 1 is set in the Configuration Command register. The DEVICEID1 default register value is read only, but may be changed during and immediately following RST# assertion if desired. See *IDDAT* section. The device ID value may be easily updated with only metal changes.

DEVID1 R+		DEVID0 R	
31	DID15=0	23	DID07=0 \
30	DID14=1	22	DID06=1 | MS
29	DID13=1	21	DID05=1 |
28	DID12=1	20	DID04=1 /
27	DID11=0 \	19	DID03=1
26	DID10=0 | LS	18	DID02=0
25	DID09=0 |	17	DID01=0
24	DID08=0 /	16	DID00=0

Bit		Name	Definition
31	(0)	DID15	Normally reads 0, 4th device ID character MSB.
30	(1)	DID14	Normally reads 1, 4th device ID character.
29	(1)	DID13	Normally reads 1, 4th device ID character.
28	(1)	DID12	Normally reads 1, 4th device ID character LSB.
27	(0)	DID11	Normally reads 0, 3rd device ID character MSB.
26	(0)	DID10	Normally reads 0, 3rd device ID character.
25	(0)	DID09	Normally reads 0, 3rd device ID character.
24	(0)	DID08	Normally reads 0, 3rd device ID character LSB.
23	(0)	DID07	Normally reads 0, 2nd device ID character MSB.
22	(1)	DID06	Normally reads 1, 2nd device ID character.
21	(1)	DID05	Normally reads 1, 2nd device ID character.
20	(1)	DID04	Normally reads 1, 2nd device ID character LSB.
19	(1)	DID03	Normally reads 1, 1st device ID character MSB.
18	(0)	DID02	Normally reads 0, 1st device ID character.
17	(0)	DID01	Normally reads 0, 1st device ID character.
16	(0)	DID00	Normally reads 0, 1st device ID character LSB.

Command (COMMAND[1:0])

Type: R/W
Address: N04h bits15-0 (dual access, see Device Register space)

The COMMAND register provides coarse control over a PCI device's ability to generate and respond to PCI transactions. When a zero is written to this register, the AIC-7870 is logically disconnected from the PCI bus transactions except for Configuration Space transactions. The COMMAND register value should be 0h until the base offset registers are loaded with their operating values, none of which should be the same value. The COMMAND register may be read at any time in Configuration Space.

COMMAND1 R/W		COMMAND0 R/W	
15	RSVD	07	WAITCTLEN=0
14	RSVD	06	PERRESPEN
13	RSVD	05	VSNOOPEN=0
12	RSVD	04	MWRICEN
11	RSVD	03	SPCYCEN=0
10	RSVD	02	MASTEREN
09	MFBTBEN=0	01	MSPACEEN
08	SERRESPEN	0	ISPACEEN

Bit		Name	Definition
15-10	(0)	Reserved	Always reads 0.
09	(0)	MFBFEN	**Master Fast Back-to-back Enable.** When active (=1), indicates a master can perform Fast Back-to-back transactions to different PCI targets. The AIC-7870 does not support this feature and MFBTBEN always reads as 0.
08	(0)	SERRESPEN	**System Error Response Enable.** When active (=1) and PERRESPEN is also active, enables output SERR# to be asserted when a PCI 36-bit even parity error is detected by a target during the address phase(s) of transactions, Special Cycle Transaction Data phase and for other errors where the result will be a catastrophic error. The AIC-7870 as a target only asserts SERR# for detected address parity errors and as a master does not assert SERR#. SERRESPEN is set inactive during and after assertion of RST#.
07	(0)	WAITCTLEN	**Wait Control Enable.** Always reads 0. (May only be set to one by those devices that do not meet the PCI output specification of 33-10.) The AIC-7870 does not support WAITCLTEN.
06	(0)	PERRESPEN	**Parity Error Response Enable.** When active (=1), enables PERR# to be asserted when a PCI 36-bit even parity error is detected during Data phases of transactions, except for Special Cycle Transaction Data phase. PERRESPEN must also be active for an address parity error to be reported on SERR#. The AIC-7870 will assert PERR# when PERRESPEN is active and a data parity error is detected as a target for write accesses or as an master for read commands. PERRESPEN must also be active to allow DPE or DPR active conditions to cause IRQA# (when enabled) to be asserted. PERRESPEN is set inactive during and after assertion of RST#.
05	(0)	VSNOOPEN	**VGA Snoop Enable.** Always reads 0. The AIC-7870 does not support VSNOOPEN.

(Continued)

Bit		Name	Definition
04	(0)	MWRICEN	**Memory Write and Invalidate Enable.** When active (=1), enables a PCI master to issue Memory Write and Invalidate commands to more optimally transfer data to system memory. When inactive, the Memory Write and Invalidate command will be replaced with Memory Write command. The AIC-7870 as a master will issue MWRIC commands when MWRICEN is active, the data FIFO contains stored data or space to store data that is equal to the selected cache size (not zero), the address is on the cache line start location, a FIFO flush condition is not indicated, and the HCNT value is also equal to or greater than the selected cache size. MWRICEN is set inactive during and after assertion of RST#.
03	(0)	SPCYCEN	**Special Cycle Enable.** Always reads 0. When active (=1), allows a target to monitor special cycle transactions broadcast on the PCI bus. The AIC-7870 does not support special cycles as an target or master.
02	(0)	MASTEREN	**Master Enable.** When active (=1), enables the AIC-7870 to perform bus master transactions on the PCI bus. Additional transactions to the AIC-7870 Device registers must be performed before the AIC-7870 may request to be a bus master. When inactive, the AIC-7870 bus master transactions are inhibited. MASTEREN is set inactive during and after assertion of RST#.
01	(0)	MSPACEEN	**Memory Space Enable.** When active (=1), enables the AIC-7870 to respond to Device register transactions through mapped memory space (see BASEADR1 register) or external ROM transaction through mapped memory space (see EXROMCTL register). When MSPACEEN is inactive, the AIC-7870 will not respond to device space accesses from memory mapped addresses. MSPACEEN is set inactive during and after assertion of RST#.
00	(0)	ISPACEEN	**IO Space Enable.** When active (=1), enables the AIC-7870 to respond to device register transactions through mapped IO space (see BASEADR0 register). When inactive, the AIC-7870 will not respond to device space accesses from IO mapped addresses. ISPACEEN is set inactive during and after assertion of RST#.

PCI Status (STATUS[1:0])

Type: R/W
Address: N04h bits 31-16 (dual access (STATUS1 only), see Device Register space)

The STATUS register is used to record status information for PCI bus related events. Read transactions of the STATUS register will access the currently stored status information. Write transactions to the STATUS register are not used to store data but to change selected active bits to be inactive (=0). To change a bit to be inactive, the data value written for that bit (=1) with all other bits not being changed inactive (=0). Any bit 30:27, 24 active and enabled in STATUS1 will cause the PCIERRSTAT bit in the ERROR register to be active and an interrupt to be generated unless FAILDIS or POWRDN is active, or INTEN is inactive. The STATUS[1:0] register is forced to be inactive when RST# is asserted. The STATUS[1:0] register may be read at any time in Configuration space.

STATUS1 R/W		STATUS0 R	
31	DPE	23	TFBTBC
30	SSE	22	RSVD
29	RMA	21	RSVD
28	RTA	20	RSVD
27	STA	19	RSVD
26	DST1	18	RSVD
25	DST0	17	RSVD
24	DPR	16	RSVD

Bit		Name	Definition
31	(0)	DPE	**Detected Parity Error.** Set active (=1) when a 36-bit even-parity error is detected by a target during an Address phase or a Write Data phase (except for Special Cycles) and by the transaction master during a Read Data phase. DPE is set inactive during and after assertion of RST# or by a write to the STATUS register with bit 31 (=1). When the AIC-7870 sets its DPE bit active with PERRESPEN active and FAILDIS inactive, it will cause an interrupt to be generated to the driver to handle the exception condition and assert PERR# for write accesses to the AIC-7870 as a target or for read transactions with the AIC-7870 as a master.
30	(0)	SSE	**Signal System Error.** Set active (=1) whenever an agent asserts SERR#. SSE is set inactive during and after assertion of RST# or by a write to the STATUS register with bit 30 (=1). The AIC-7870 sets its SSE bit active only when PERRESPEN and SERRESPEN are active for detected address parity errors. When SSE is active and FAILDIS is inactive, it will cause an interrupt to be generated to the driver to handle the exception condition.
29	(0)	RMA	**Received Master Abort.** Set active (=1) when an AIC-7870 bus master generated transaction is terminated by the AIC-7870 for no response from the intended target by the sixth (for SAC) or seventh (for DAC) PCLK after the AIC-7870 asserted FRAME#. The AIC-7870 will release the bus on the next PCLK and not retry the transaction. Software/firmware intervention is required for the AIC-7870 to continue master transactions. RMA is set inactive during and after assertion of RST# or by a write to the STATUS register with bit 29 (=1). The AIC-7870 will also set RMA active should the addressed target deassert DEVSEL# while the AIC-7870 is asserting FRAME#, a PCI protocol violation.

(Continued)

Bit		Name	Definition
			Note: Should RMA be cleared with the aborted master transaction still waiting to complete, the AIC-7870 will retry the transaction. To prevent this action, if desired, HDMAEN should be set inactive prior to clearing RMA. The interrupt will remain active till cleared with CLRPARERR.
28	(0)	RTA	**Received Target Abort.** Set active (=1) when the target of an AIC-7870 bus generated transaction is terminated by the target, with a target-abort indication. RTA is set inactive during and after assertion of RST# or by a write to the STATUS register with bit 28 (=1). When a target-abort indication is received, the AIC-7870 will not retry the transaction and software/firmware intervention is required for the AIC-7870 to continue master transactions. Note: Should RTA be cleared with the AIC-7870 still waiting to complete the aborted master transaction, the AIC-7870 will retry the transaction. To prevent this action, if desired, HDMAEN should be set inactive prior to clearing RTA. The interrupt will remain active till cleared with CLRPARERR.
27	(0)	STA	**Signal Target Abort.** Set active (=1) by the target of a PCI bus transaction unable to respond due to a fatal error condition. STA is set inactive during and after assertion of RST# or by a write to the STATUS register with bit 27 (=1). The AIC-7870 will indicate target-abort for: • Incorrect data width — must be 8 bits for Device space, except for SCI double word write — must be 8 bits for ROM space write • Value stored in base address registers for BASEADR1 and EXROMCTL are the same and EXROMEN is active. • SEEMS is active and an access is made to SCB or ROM space address. • Accesses with POWRDN active except for Host only registers and Configuration registers. • Address parity error detected with correct address compare (SERR# asserted) for current access. Note: No valid data (CBE[3:0] value of Fh for a Data phase) is not an error condition.
26-25	(1)	DST[1:0]	**Device Select Timing[1:0].** Value indicates the longest response time of a PCI device for assertion of DEVSEL# for any bus transaction with valid values of 0h for *fast* (1 PCLK), 1h for *medium* (2 PCLKs), 2h for *slow* (3 PCLKs) with value 3h *reserved*. Respond time for the AIC-7870 is *medium*. DST[1:0] are fixed value read only bits.
24	(0)	DPR	**Data Parity Reported.** When active (=1), indicates the master of a transaction, with its PERRESPEN bit active, has detected PERR# asserted or asserted PERR#. DPR is set inactive during and after assertion of RST# or by a write to the STATUS register with bit 24 (=1).
23	(1)	TFBTBC	**Target Fast Back-to-back Capable.** When active (=1), indicates that the target is capable of accepting Fast PCI Back-to-back transactions even when the transactions are not to the same target. The AIC-7870 as a target supports Fast Back-to-back transactions. TFBTBC is a read only bit.
22-16	(0)	RSVD	Always reads 0.

Device Revision ID (DEVREVID)

Type: R
Address: N08h

The Device Revision ID identifies the revision level of a PCI device. Part revision ID may be read at any time in Configuration space. Easy device revision value changes in metal only.

PARTREVID R	
7	DRID7
6	DRID6
5	DRID5
4	DRID4
3	DRID3
2	DRID2
1	DRID1
0	DRID0

Bit		Name	Definition
07-00	(01h)	DRID[7:0]	DRID[7:0] always reads 01h.

Programming Interface (PROGINFC)

Type: R
Address: N08h

The Programming Interface register value identifies the specific register-level programming interface the agent supports. The PROGINFC for the first version of the AIC-7870 will be identified as 00h (not VGA compatible). PROGINFC may be read at any time in Configuration space.

PROGINFC R	
15	PGFC7
14	PGFC6
13	PGFC5
12	PGFC4
11	PGFC3
10	PGFC2
09	PGFC1
08	PGFC0

Bit		Name	Definition
15-08	(0)	PGFC[7:0]	Always reads 0.

Sub Class (SUBCLASS)

Type: R
Address: N08h

The Sub Class register identifies the sub class the PCI device is assigned to. The SUBCLASS for the first version of the AIC-7870 will be identified as 00h (SCSI bus controller). SUBCLASS may be read at any time in Configuration space.

SUBCLASS R	
23	SCLASS7
22	SCLASS6
21	SCLASS5
20	SCLASS4
19	SCLASS3
18	SCLASS2
17	SCLASS1
16	SCLASS0

Bit		Name	Definition
23-16	(0)	SCLASS[7:0]	Always reads 0.

Base Class (BASECLASS)

Type: R
Address: N08h

The Base Class register identifies the base class the PCI device has been assigned to. The BASECLASS for the first version of the AIC-7870 will be identified as 01h (mass storage controller). BASECLASS may be read at any time in Configuration space.

BASECLASS R	
31	BCLASS7
30	BCLASS6
29	BCLASS5
28	BCLASS4
27	BCLASS3
26	BCLASS2
25	BCLASS1
24	BCLASS0

Bit		Name	Definition
31-24	(01h)	BCLASS[7:0]	Always reads 01h.

Cache Line Size (CACHESIZE)

Type: R/W
Address: N0Ch

The Cache Line Size register specifies the system cache line size in units of 32-bit DWDs. The value stored in the register defines the minimum data transfer size and associated cache starting boundary (and multiples thereof) that may be performed with cache line referenced PCI MWRIC, MRDLC or MRDMC commands.

The AIC-7870 initiated burst cycle transactions can last indefinitely as long as GNT# remains asserted, provided that data or space for data that is being transferred and its transfer byte count has not expired. However, if GNT# is deasserted after the burst cycle is initiated, the AIC-7870 further limits the duration of the burst cycle to the number of CLKs specified by the LATTIME register, plus completion of an in-process cache line transfer referenced command. When the stored value in the CASHESIZE register is 0h, the AIC-7870 will issue MWRC or MRDC instead of MWRIC, MRDLC or MRDMC for data transfer.

Note the effect of MRDCEN active state on command issued and stopping point. CACHESIZE register may be read at any time in Configuration space. CDWDSIZE[5:2] are reset to 0h during assertion of RST#.

CACHESIZE R-R/W	
07	RSVD
06	RSVD
05	CDWDSIZE5
04	CDWDSIZE4
03	CDWDSIZE3
02	CDWDSIZE2
01	CDWDSIZE1=0
00	CDWDSIZE0=0

Bit		Name	Definition
07-06	(0)	RSVD	Always reads 0.
05-00	(0)	CDWDSIZE[5:0]	**Cache Double Word Size [5:0]**. Define the cache line size that the AIC-7870 as a master supports. Note that CDWDSIZE[1:0] always reads 0.

Cachesize Value [7:0]	Action
0	Use of MWRIC, (MRDLC and MRDMC) are disabled and replaced with MWRC or MRDC respectively.
1-3	RSVD (default to value 0)
4	MWRIC, MRDLC and MRDMC use is enabled with a cache line size of 4 DWDs (16 Bytes).
5-7	RSVD (default to value 4)
8	MWRIC, MRDLC and MRDMC use is enabled with a cache line size of 8 DWDs (32 Bytes).
9-15	RSVD (default to value 8)
16	MWRIC, MRDLC and MRDMC use is enabled with a cache line size of 16 DWDs (64 Bytes).
17-31	RSVD (default to value 16)
32	MWRIC, MRDLC and MRDMC use is enabled with a cache line size of 32 DWDs (128 Bytes).
33-255	RSVD (default to value 32)

Latency Timer (LATTIME)

Type: R/W
Address: N0Ch (dual access, see Device Register space)

The AIC-7870's master latency timer is held initialized until the AIC-7870 asserts FRAME#, then it is enabled to count PCLKs. Whenever FRAME# is deasserted, the LATTIME timer is reinitialized. When the AIC-7870's latency timer expires with FRAME# still asserted, then the AIC-7870 will initiate transaction termination as soon as its GNT# is deasserted (unless a Cache Line Referenced command was issued and is in a process which must be completed before termination) and the target asserts TRDY# on the final Data phase. LATTIME register may be read/written at any time in Configuration space. The LATTIME register is cleared to 0h during RST# assertion.

When the LATTIME value is less than the CACHESIZE value and the GNT# assertion period is also less than the CACHESIZE value, each PREQ# period and data transferred will be the same as the cache size when Cache Line Referenced commands are issued. Providing that the data beginning and end addresses are cache line boundaries.

LATTIME R/W	
15	LATT7
14	LATT6
13	LATT5
12	LATT4
11	LATT3
10	LATT2
09	LATT1=0
08	LATT0=0

Bit		Name	Definition
15-10	(0)	LATT[7:2]	Latency Timer [7:2] are read/write and their value determines the AIC-7870's bus master latency timer period (in PCLK periods)
09-08	(0)	LATT[1:0]	Latency Timer [1:0] always read 0 (sets granularity at four CLKs).

Header Type (HDRTYPE)

Type: R
Address: N0Ch

The Header Type register specifies the PCI Configuration header type the device supports. HDRTYPE register may be read at any time in Configuration space. The AIC-7870 supports PCI Configuration header type 00h (with only one function).

HDRTYPE R	
23	MFDEV=0
22	HTYPE6
21	HTYPE5
20	HTYPE4
19	HTYPE3
18	HTYPE2
17	HTYPE1
16	HTYPE0

Bit		Name	Definition
23	(0)	MFDEV	**Multifunction Device.** When active (=1), indicates the device is a device containing multiple independent functions, each of which contains a Configuration space containing device data and address mapping for its function. When MFDEV is active, PCI POST software scan for multiple functions by scanning for the multiple Configuration spaces using AD[10:8] with the same IDSEL active. When MFDEV is inactive, no scanning is required. The AIC-7870 is a single function device and MFDEV always read 0.
22-16	(0)	HTYPE[6:0]	**Header Type.** Always reads 0.

Base Address 0 (BASEADR0)

Type: R/W
Address: N10h

Base Address register 0 enables the AIC-7870 Device register space and HIM RAM to be relocated (mapped) within system IO address space to enable the system board device independent POST software to build a consistent IO address map. BASEADR0 may be read at any time in Configuration space. BASEADR0 value is reset to 1h during RST# assertion.

BASEADR0 R/W		BASEADR0 R/W		BASEADR0 R/W		BASEADR0 R	
31	IBMADR29	23	IBMADR21	15	IBMADR13	07	IBMADR05=0
30	IBMADR28	22	IBMADR20	14	IBMADR12	06	IBMADR04=0
29	IBMADR27	21	IBMADR19	13	IBMADR11	05	IBMADR03=0
28	IBMADR26	20	IBMADR18	12	IBMADR10	04	IBMADR02=0
27	IBMADR25	19	IBMADR17	11	IBMADR09	03	IBMADR01=0
26	IBMADR24	18	IBMADR16	10	IBMADR18	02	IBMADR00=0
25	IBMADR23	17	IBMADR15	09	IBMADR07	01	RSVD
24	IBMADR22	16	IBMADR14	08	IBMADR06	00	ISPACEIND=1

Bit		Name	Definition
31-08	(0)	IBMADR[29:06]	**IO Base Map Address [31:08].** Bits are read/write capable to provide the ability for device independent software on the system board to relocate the AIC-7870 Device register space on 256 byte IO command boundaries within the low 32-bit address segment of the 64-bit address space.
07-02	(0)	IBMADR[05:00]	**IO Base Map Address [07:02].** Always reads 0.
01	(0)	RSVD	Always reads 0.
00	(1)	ISPACEIND	**IO Space Indicator.** Always reads 1. Note: Bit [00] =1 indicates that BASEADR0 register is used for mapping into system IO address space.

Base Address 1 (BASEADR1)

Type: R/W
Address: N14h

Base Address register 1, enables the AIC-7870 Device register space to be relocated (mapped) within system Memory Address space to enable the system board device independent POST software to build a consistent system memory address map. The AIC-7870 Device register space, located in Memory Address space, improves throughput of the AIC-7870 Device register transactions. BASEADR1 may be read at any time in Configuration space. BASEADR1 value is reset to 0h during RST# assertion.

> **Note:** Software when using BASEADR1 with MSPACEN active to access Device registers must ensure that instructions not allow data moves that bridge 32-bit boundaries to ensure that bytes are not transferred out of intended order. When an access is made to the AIC-7870 when BASEADR1 and EXROMCTL registers contain the same value and EXROMEN is active will result in a Target-Abort response.

BASEADR1 R/W		BASEADR1 R/W		BASEADR1 R/W		BASEADR1 R	
31	MBMADR27	23	MBMADR19	15	MBMADR11	07	MBMADR03=0
30	MBMADR26	22	MBMADR18	14	MBMADR10	06	MBMADR02=0
29	MBMADR25	21	MBMADR17	13	MBMADR09	05	MBMADR01=0
28	MBMADR24	20	MBMADR16	12	MBMADR08	04	MBMADR00=0
27	MBMADR23	19	MBMADR15	11	MBMADR07=0	03	PREFETCH=0
26	MBMADR22	18	MBMADR14	10	MBMADR06=0	02	MSPACTYP1=0
25	MBMADR21	17	MBMADR13	09	MBMADR05=0	01	MSPACTYP0=0
24	MBMADR20	16	MBMADR12	08	MBMADR04=0	00	MSPACEIND=0

Bit		Name	Definition
31-12	(0)	MEMADR[27:08]	**Memory Base Map Address[27:08].** Bits are read/write capable to indicate a mapping increment capability of 4096 bytes of system memory space with the AIC-7870's 256 byte memory command range rolling over within the selected 4096 byte increment within the low 32-bit address segment of the 64-bit address space.
11-04	(0)	MEMADR[07:00]	**Memory Base Map Address[07:00].** Indicates address space requirement. Always reads 0.
03	(0)	PREFETCH	**Prefetchable.** Always reads 0. The AIC-7870 does not support this feature.
02-01	(0)	MSPACTYP[1:0]	**Memory Space Access Type [1:0].** Always reads 0. The AIC-7870 as a target may be located anywhere in the low 32-bit address segment of the 64-bit address space. MSPACETYPE[1:0] 00 = locate anywhere in 32-bit address space (AIC-7870) 01 = locate below 1 Meg 10 = locate anywhere in 64-bit address space 11 = PCI RSVD
00	(0)	MSPACEIND	**Memory Space Indicator.** Always reads 0. Note: Bit [00] =0 indicates that BASEADR1 register is used for mapping into system Memory Address space.

External ROM Control (EXROMCTL)

Type: R/W
Address: N30h

The External ROM Control Base Address register is used to define the base address, maximum size and access enable control of a local external ROM which may be used with a PCI device. The external ROM's data and address must pass through an additional interface Memory Port of the device and use its normal address/data path to the PCI bus so that no additional loading is presented to the PCI bus other than the device's loading without the external ROM. EXROMCTL may be read at any time in Configuration space. EXROMCTL value is reset to 0h during RST# assertion. The AIC-7870 supports an external ROM/EEPROM of 64 KBytes and no slower than 150 nsec access. When an access is made to the AIC-7870 both BASEADR1 and EXROMCTL registers contain the same value and EXROMEN is active will result in a Target-Abort response.

EXROMCTL R/W		EXROMCTL R/W		EXROMCTL R/W		EXROMCTL R/W	
31	MBAXROM20	23	MBAXROM12	15	MBAXROM04=0	07	RSVD
30	MBAXROM19	22	MBAXROM11	14	MBAXROM03=0	06	RSVD
29	MBAXROM18	21	MBAXROM10	13	MBAXROM02=0	05	RSVD
28	MBAXROM17	20	MBAXROM09	12	MBAXROM01=0	04	RSVD
27	MBAXROM16	19	MBAXROM08	11	MBAXROM00=0	03	RSVD
26	MBAXROM15	18	MBAXROM07	10	RSVD	02	RSVD
25	MBAXROM14	17	MBAXROM06	09	RSVD	01	RSVD
24	MBAXROM13	16	MBAXROM05	08	RSVD	00	EXROMEN

Bit		Name	Definition
31-15	(0)	MBAXROM[20:04]	**Mapped Base Address External ROM[20:05].** Bits are read/write capable to indicate a mapping increment capability of 64 KBytes.
14-11	(0)	MBAXROM[04:00]	**Mapped Base Address External ROM[04:00].** Always reads 0 to set the maximum ROM size to 64 KBytes.
10-01	(0)	RSVD	Always reads 0.
00	(0)	EXROMEN	**External ROM Enable.** When active (=1) (and MSPACEEN in the Configuration Command register is active), enables the device to accept accesses to its expansion ROM. MSPACEEN in the COMMAND register should not be active when writing to set EXROMEN active. For reads, the AIC-7870 will use latched AD[31:02] and internally generated address [1:0] to access four bytes from the AIC-7870's external ROM/EEPROM for each PCI expansion ROM read access, with TRDY# deasserted until all bytes are assembled into 32-bits for the access regardless of the CBE[3:0]# value. The first byte read from the ROM/EEPROM will be stored in bits 7-0 and the fourth byte in bits 31-24. As the combined access exceeds the maximum PCI recommended data access period, all PCI burst attempts will be disconnected on each Data phase transfer of the transaction. For writes, the AIC-7870 will use latched AD[31:02] and the single asserted CBE[3:0]# to generate the internal address [1:0] to write a byte to the external EEPROM with TRDY# deasserted until the access is completed. PCI burst attempts will be disconnected after the first Data phase transfer. Accesses with more than one CBE bit asserted will result in a Target-Abort response. Note that accesses may also be extended due to arbitration when required. Unless both EXROMEN and MSPACEEN are active, access attempts to external ROM addresses will not return DEVSEL# and will be ignored.

Interrupt Line Select (INTLINSEL)

Type: R/W
Address: N3Ch

The Interrupt Line Select register provides the capability for the system to communicate to the device's software driver the system interrupt line that has been connected to the devices interrupt pin when one has been included in the devices design. When no interrupt pin is provided in the device's design, this register is RSVD and read only with a value of 0h.

INTLINSEL R/W	
07	INTLS7
06	INTLS6
05	INTLS5
04	INTLS4
03	INTLS3
02	INTLS2
01	INTLS1
00	INTLS0

Bit		Name	Definition
7-0	(0)	INTLS[7:0]	**Interrupt Line Select [7:0].** Bits are read-write register bits in which are stored the interrupt line to which the devices interrupt output IRQA[D:A]# has been connected in the system in which it is installed.

INTLS[7:0]	Assignment
0-15	Interrupt numbers (referenced to a standard dual 8259 configuration).
16-254	RSVD.
255	No connection or unknown. Note: The AIC-7870 driver does not operate with this setting.

Interrupt Pin Select (INTPINSEL)

Type: R
Address: N3Ch

The Interrupt Pin register specifies the PCI interrupt pin the device (or device function) uses. A separate Configuration space is required for each function in a device and only one pin may be identified in each space. INTPINSEL register may be read at any time in Configuration space. The AIC-7870 supports PCI Configuration header type 00h (only one function) with IRQA#.

INTPS[7:0]	
0	Device does not use an interrupt pin
1	Device uses interrupt pin IRQA#
2	Device uses interrupt pin IRQB#
3	Device uses interrupt pin IRQC#
4	Device uses interrupt pin IRQD#

INTPINSEL R	
15	INTPS7=0
14	INTPS6=0
13	INTPS5=0
12	INTPS4=0
11	INTPS3=0
10	INTPS2=0
09	INTPS1=0
08	INTPS0=1

Bit		Name	Definition
15-8	(1h)	INTPS[7:0]	Always reads 1h.

Min_Gnt Status (MINGNT)

Type: R
Address: N3Ch

The Minimum Grant register indicates the desired GNT# asserted burst period needed to complete transfer of a devices data buffer assuming that the intended target does not extend the transfer time by use of TRDY#. The value read from the register specifies a period of time in units of 0.25 microsecond. The AIC-7870's MINGNT register value is 8h which is the minimum time to burst out its 256-byte buffer. *The AIC-7870 is able to operate with any size GNT# period from one clock to constant park condition.* MINGNT register may be read at any time in Configuration space.

	MINGNT R
23	MINGNT7=0
22	MINGNT6=0
21	MINGNT5=0
20	MINGNT4=0
19	MINGNT3=1
18	MINGNT2=0
17	MINGNT1=0
16	MINGNT0=0

Bit		Name	Definition
23-16	(8h)	MINGNT[7:0]	Always reads 8h.

Max_Lat Status (MAXLAT)

Type: R
Address: N3Ch

The Maximum Latency register indicates the desired LATTIME register value needed to complete transfer of a devices data buffer assuming that the intended target does not extend the transfer time by use of TRDY#. The value read from the register specifies a period of time in units of 0.25 microsecond. The AIC-7870's MAXLAT register value is 8h which is the minimum time to burst out its 256-byte buffer. *The AIC-7870 is able to operate with any size LATTIME register value.* MAXLAT register may be read at any time in Configuration space.

MAXLAT R	
31	MAXLAT7=0
30	MAXLAT6=0
29	MAXLAT5=0
28	MAXLAT4=0
27	MAXLAT3=0
26	MAXLAT2=0
25	MAXLAT1=0
24	MAXLAT0=0

Bit		Name	Definition
31-24	(8h)	MAXLAT[7:0]	Always reads 8h.

Device Configuration (DEVCONFIG)

Type: R/W
Address: N40h

Device Configuration register provides the AIC-7870 with mode selection control of features in the AIC-7870. DEVCONFIG may be read/written at any time in Configuration space. However, changing values in this register must be done with care due to the functional changes they control. DEVCONFIG value is reset to 0h only during RST# assertion.

DEVCONFIG R/W	
07	DECONFIG7
06	MRDCEN
05	EXTSCBTIME
04	EXTSCBPEN
03	BERREN
02	DACEN
01	STPWLEVEL
00	DIFACTNEGEN

Bit		Name	Definition
07	(0)	DECONFIG	**Device Configuration.** A read/write bit with no current logic control assignment
06	(0)	MRDCEN	**Memory Read Command Enable.** When active (=1), enables the PCI master command out generator to generate the MRDC command value for its defined conditions. Enables master burst transfers being made with MRDLC or MRDMC commands to release the bus after the current and next data transfers are completed when the LATTIME register has expired and GNT# is not asserted. When inactive, causes the PCI master command out generator to convert all MRDC commands to MRDLC commands. Enables master burst transfers being made with MRDLC or MRDMC commands, when the LATTIME register has expired and GNT# is not asserted, to release the bus after completing the current cache line.
05	(0)	EXTSCBTIME	**External SCB Time.** When active (=1), selects the long (2X) external SCB cycle time for external SRAM connected to the Memory Port, and when inactive selects the normal (1X) external SCB cycle time.
04	(0)	EXTSCBPEN	**External SCB Parity Enable.** When active (=1) and RAMPSM is active, enables checking of odd-byte parity for SCB data read from an external 9-bit SRAM. Note that when a 8-bit SRAM is present, do not enable checking of external SCB parity.
03	(0)	BERREN	**Byte Parity Error Enable.** When active (=1), forces the internal byte parity generators to create even-parity for the associated data byte and when BERREN is inactive normal odd-parity is generated. BERREN provides the capability to test this logic and the parity checking logic in the data path (PCI to data FIFO, data FIFO and SCSI to data FIFO less the SCSI PIO odd-byte parity generator)

(Continued)

Bit		Name	Definition
			Note: Data bytes being written to the data FIFO have odd-parity bits attached to them when they are from the PCI bus with the AIC-7870 as a master, from Device register DFDAT access (sequencer or the AIC-7870 as a target) or from SCSI reads. Bytes written to scratch RAM, QINFIFO, QOUTFIFO or SCB RAM also have odd-parity bits attached to them. When reading bytes from the data FIFO, scratch RAM, QINFIFO, QOUTFIFO and the SCB RAM the odd-parity is tested and a error will be flagged when an improper value is accessed. After power-up, all RAM byte locations that are to be used must be written to initialize their parity bits.
02	(0)	DACEN	**Dual Address Cycle Enable.** When active, enables the AIC-7870 to issue Dual Address Cycle (DAC) master transactions of 32-bit range within a 32-bit page of a 64-bit range pointed to by the value stored in the HHADDR[3:0] registers. When HHADDR[3:0] stored value is zero, only Single Address Cycles (SAC) may be issued the same as when DACEN is not active.
01	(0)	STPWLEVEL	**SCSI Termination Power Level.** When inactive (=0), selects the high level for the active state of the STPWCTL output. When active, selects the low level for the active state of output STPWCTL. STPWLEVEL is cleared to the inactive state by RST# assertion. Writing to the CHIPRST bit in the Device HCNTRL register has no effect on STPWLEVEL.
00	(0)	DIFACTNEGEN	**Differential Active Negation Enable.** When active (=1), enables the chip SCSI interface to be connected directly to the external differential components. This reduces the power used in the differential configuration and requires only light pull-ups instead of the otherwise required standard single-ended SCSI bus terminators.

Device Status (DEVSTATUS)

Type: R
Address: N40h

Device Status register provides read capability for selected internal conditions in the AIC-7870. DEVSTATUS may be read at any time in Configuration space.

DEVSTATUS R	
15	
14	
13	
12	
11	
10	MPORTMODE
09	RAMPSM
08	VOLSENSE

Bit		Name	Definition
15-11	(0)	Not Used	Are unused bits that always read as 0.
10	(*)	MPORTMODE	**Memory Port Mode**. Provides the capability to determine when the AIC-7870's Memory Port is in Single-user (dedicated) mode (=1) or in Multiuser mode (=0). In Single-user mode, the Memory Port outputs are always driven to prevent floating signal pins. In Multiuser mode, external termination is required to prevent the floating condition. Note: (*) The state following reset is determined by the state of input EXARBACK# during reset. When EXARBACK# is asserted MPORTMODE will be (=1) and when not asserted will be (=0).
09	(*)	RAMPSM	**RAM Present Mode**. Provides the capability to determine when the AIC-7870's Memory Port has external RAM present (=1) that will be used for SCB data storage. When RAMPSM is (=0), SCB data is stored internally in the AIC-7870. Note: (*) The state following reset is determined by the state of input RAMPS# during reset. When RAMPS# is asserted RAMPSM will be (=1) and when not asserted will be (=0).
08	(*)	VOLSENSE	**Voltage Sense.** Provides the capability to determine which PCI bus voltage level ((=0) for 3.3V and (=1) for 5V) that the AIC-7870's PCI interface has been connected to. The state of VOLSENSE adjusts the operation of the AIC-7870's PCI interface pin cells to account for the difference in voltage. Note: (*) The reset state is determined by the external voltage present.

Host Device Space Register Definition

Registers in the host Device register space may be written and/or read through the AIC-7870 PCI bus interface, or through the host internal interface by the sequencer with time bases that may be asynchronous. For registers that are written by one source and read by another, the host register read data is latched at the beginning of the read to provide stable data for those reads. Users of direct register outputs need to consider their asynchronous nature. Also some registers in normal operation function as

counters and data memories that may be constantly changing their contents. Reading from these registers should be restricted to certain operational states where their activity will be idle. Writing to some of these registers is restricted when HDMAENACK is active. Also some registers when written may have to be read multiple times to verify the desired action has occurred.

Register addresses shown for host Device registers refer to internal chip address values which have extended external address prefix values which must be mapped to the desired value in the system address map via BASEADR0 or BASEADR1 registers in the Configuration register space. Writing to CHIPRST bit (=1) affects only registers in the AIC-7870 Device register space, while assertion of RST# affects both Configuration and Device registers.

Defined Device register unused bits read as 0h. Undefined Device register addresses when read, will return the last value asserted on the CSDAT[7:0] bus except for device address 8Fh which will return 0h. Data written to the AIC-7870 Device register unused Device register bits and unassigned Device register addresses is not stored. The following conventions are used throughout this section:

- **set** or **active:** Indicates that the bit was loaded with or is a 1
- **cleared** or **inactive:** Indicates that the bit was loaded with or is a 0
- **(0):** Indicates that the bit is cleared when the reset POR is active
- **(1):** Indicates that the bit is set when the reset POR is active
- **(x):** Indicates that the bit is in an unknown state both during and after the reset POR condition
- **Mxxh:** Indicates PCI address decode with address prefix stored in either BASEADR0 or BASEADR1 registers in the Configuration space.
- **DS-xxh:** Indicates Device space internal address decode. It represents Mxxh without the M prefix and is the value used by the internal sequencer block.

  Note: Only the AIC-7870 Device registers (DSVENDID[1:0], DSDEVID[1:0], HCNTRL, INTSTAT, CLRINT, ERROR, QOUTFIFO(R) and QOUTCNT(R)) may be accessed from the PCI bus without regard for the state of PAUSEACK when ISPACEEN or MSPACEEN is active in the Configuration Command register. These Device registers are never accessible by the internal sequencer. Device registers that are used by the sequencer, when it is not paused may be accessed from the PCI bus when PAUSACK is not active with a delay in the access while automatic access PAUSE (AAP) action causes PAUSEACK to become active. No access delay will occur for accesses performed after the PAUSE bit in the Device HCNTRL register is first set and polled until PAUSEACK is active 2. When both ISPACEEN and MSPACEEN are inactive in the Configuration Command register, none of the AIC-7870 Device registers are accessible from the PCI bus.

DSVendor Identification (DSVENDID[1:0])

Type: R
Address: M80h/81h Dual access DS-80h/81h

The PCI vendor identifier registers contain product information for use by the host system software when initializing and configuring the system. The Vendor ID contains two bytes of a compressed representation of the vendor code. The vendor characters for the AIC-7870, represented by the value 9004h, are

- the first ID character = A
- the second ID character = D
- the third ID character = P

DSVENDOR ID[1:0] may be read without consideration of the state of the PAUSE bit in the AIC-7870 Device register space, only when the ISPACEEN bit or the MSPACEEN bit is set in the Configuration Command register. For bit assignments see the VENDORID register in the Configuration register space.

DSVENDID1 R 81h		DSVENDID0 R 80h	
7	DSVID15=1	7	DSVID07=0
6	DSVID14=0	6	DSVID06=0
5	DSVID13=0	5	DSVID05=0
4	DSVID12=1	4	DSVID04=0
3	DSVID11=0	3	DSVID03=0
2	DSVID10=0	2	DSVID02=1
1	DSVID09=0	1	DSVID01=0
0	DSVID08=0	0	DSVID00=0

DSDevice Identification (DSDEVID[1:0])

Type: R
Address: M82h/83h Dual access DS-82h/83h

The two Device ID bytes contain a product code 7870, in big-endian format. DSDEVICEID[1:0] may be accessed at any time without consideration of the state of PAUSEACK in the AIC-7870 Device register space only when the ISPACEEN bit or the MSPACEEN bit is active in the Configuration Command register. For bit assignments, see the DEVICEID register in the Configuration register space.

Note: The DSDEVID1 default value may be changed. See *IDDAT* section.

DSDEVID1 R-		DSVEDID0 R-	
7	DSDID15=0	7	DSDID07=0 \
6	DSDID14=1	6	DSDID06=1 | MS
5	DSDID13=1	5	DSDID05=1 |
4	DSDID12=1	4	DSDID04=1 /
3	DSDID11=0 \	3	DSDID03=1
2	DSDID10=0 | LS	2	DSDID02=0
1	DSDID09=0 |	1	DSDID01=0
0	DSDID08=0 /	0	DSDID00=0

DSCommand (DSCOMMAND)

Type: R/W
Address: M84h D-84h Dual access for read only bits [5:0]

The DSCOMMAND register access of bits [3:0] are read only for PCI command enable selections. This allows monitoring the Configuration COMMAND[1:0] registers stored values from the AIC-7870 Device register space without requesting access to the Configuration address space. Bits [7:4] are read/write and assigned to new functions in the AIC-7870.

The DSCOMMAND register may be accessed by the sequencer when not paused and by the software driver by using AAP access or setting PAUSE bit active, only when the ISPACEEN bit or the MSPACEEN bit is set in the Configuration Command register. See *Configuration Register Space* section for register bit [3:0] usage. Writing to register bits [3:0] in the Device space will have no effect, only writing to the Configuration space or assertion of RST# will effect these bits. Bits [7:4] may be changed by writing to the Device space DSCOMMAND register, CHIPRST bit or the assertion of RST#.

DSCOMMAND R/W	
7	CACHETHEN
6	DPARCKEN
5	MPARCKEN
4	EXTREQLCK
3	DSSERRESPEN
2	DSPERRESPEN
1	DSMWRICEN
0	DSMASTEREN

Bit		Name	Definition
7	(0)	CACHETHEN	**Cache Threshold Enable.** When active (=1), will cause the PCI master to utilize signal DFCACHETH instead of DFTHRSH to determine when to request use of the PCI bus. Selecting the appropriate LATTIME and CACHESIZE register values when CACHETHEN is active will cause data transfers when the AIC-7870 is a master to match the selected cache size (deterministic transfer) using cache line referenced PCI commands. This use presumes that software has located buffers on cache line boundaries or a small transfer will be performed to reach the first cache line boundary. The final transfer indicated by FIFOFLUSH active, for system memory write direction, will be flushed even if not a full cache size.
6	(0)	DPARCKEN	**Data Parity Check Enable.** When inactive (=0), disables the AIC-7870 byte parity checking being performed as follows on internal data path byte accesses. For PIO read operations from the SCSI bus, the SCSI bus parity check control is controlled by the Device SIMODE1 register and is independent of DPARCKEN. 1 Odd byte parity generated internally in the AIC-7870 block at the PCI bus interface, passed through the data FIFO and SCSI FIFO then checked internally in the SCSI block at the SCSI bus interface, then sent to the SCSI bus, and again checked by the SCSI device connected on the SCSI bus. Data with detected internal parity errors will PAUSE the sequencer and generate an interrupt if enabled. 2 Odd byte parity generated by the active SCSI device connected to the SCSI bus, checked in the SCSI block internally at the SCSI bus interface, passed through the SCSI FIFO and the data FIFO, then checked in the AIC-7870 block internally at the PCI bus interface. Data with detected internal parity errors will PAUSE the sequencer and generate an interrupt if enabled. 3 Odd byte parity generated internally in the AIC-7870 block at the PCI bus interface, passed through the data FIFO, then checked internally in the data FIFO block at the CIOBUS-CSDAT interface for sequencer or software driver read accesses. These errors will cause the sequencer to be paused. 4 Odd byte parity generated by the active SCSI device connected to the SCSI bus, checked in the SCSI block internally at the SCSI bus interface, passed through the SCSI FIFO and the data FIFO then checked internally in the data FIFO block at the CIOBUS-CSDAT interface for sequencer or software driver read accesses. These errors will cause the sequencer to be paused. 5 Odd byte parity generated in the data FIFO block at the CIOBUS-CDDAT interface for sequencer or software driver write accesses, passed through the data FIFO, then checked as in 1 - 4 above. When DPARCKEN is active, byte parity checking is performed and parity errors will cause DPARERR to be active in the ERROR register, BRKADRINT to be active in the INTSTAT register and (if INTEN is active and POWRDN is inactive) in the HCNTRL register the IRQA# output will become active. Internal byte parity is always generated.
5	(0)	MPARCKEN	**Memory Parity Check Enable.** When inactive (=0), disables the AIC-7870 parity checking on scratch RAM, QINFIFO, QOUTFIFO and SCB Array (internal, or external when EXSCBPEN is active) byte read accesses. When MPARCKEN is active, byte odd-parity checking is performed and parity errors will cause MPARERR to be active in the ERROR register, BRKADRINT to be active in the INTSTAT register and (if INTEN is active and POWRDN is inactive) in the HCNTRL register the IRQA# output will become active. Memory byte odd-parity is always generated.

(Continued)

Bit		Name	Definition
4	(0)	EXTREQLCK	**External Request Lock.** When active (=1) causes the Memory Port output EXTARBREQ#, once it is asserted, to remain asserted until EXTREQLCK is inactive. This capability enables shared resource semaphore based accesses to be completed by a single user without being interrupted. EXTREQLCK active period should be limited as all other users are locked out from accessing the external devices at this time.
3-0	(0)		These bits are read only. See Configuration COMMAND[1:0] registers for use of these bits.

DSLatency Timer (DSLATTIME)

Type: R
Address: M85H, dual access DS-85h

The DSLATTIME register provides two functions:

- The ability to monitor the Configuration LATTIME[7:2] register stored values from the AIC-7870 Device register space without requesting access to the Configuration address space. See Configuration register space section for register bit usage.
- The ability to select host address Device register pages. Writing to DSLATTIME register will only affect HADDLDSEL[1:0] bits.

DSLATTIME R	
7	DSLATT7
6	DSLATT6
5	DSLATT5
4	DSLATT4
3	DSLATT3
2	DSLATT2
1	HADDLDSEL1
0	HADDLDSEL0

Bit		Name	Definition
7-2	(0)	DSLATT[7:2]	**Device Space Latency Timer.** (R). See Configuration register space section for register bit usage. LATT[1:0] are always 0.
1-0	(0)	HADDLDSEL[1:0]	**Host Address Load Select [1:0].** (R/W). Enable page expansion of the Host Address register addresses M/DS-88h - M/DS-8Bh as follows:

[1:0]	Bit Value Assignment
0	Enables the low four bytes of a 64-bit address to be loaded into LHADDR[3:0] and SHADDR[3:0] registers.
1	Enables the high four bytes of a 64-bit address (or alternately a general purpose storage register) to be loaded into HHADDR[3:0].
3, 2	Values are reserved. Note reading M88h - M8Bh with these values will access the last data value asserted on the CSDAT[7:0] bus.

DSPCI Status (DSPCISTATUS)

Type: R/W
Address: M86h, dual access DS-86h

The DS PCI STATUS register provides two functions:

- The ability to select the data FIFO data threshold level that will initiate a bus master cycle.
- The ability to monitor PCI exception status bits in the Configuration STATUS register. DFTHRSH[1:0] are R/W and are forced to the default value of 0 when POR is active. Writing to this register will have no effect for bits [5:0].

PCISTATUS R/W-R	
7	DFTHRSH1
6	DFTHRSH0
5	DSDPR
4	DSDPE
3	DSSSE
2	DSRMA
1	DSRTA
0	DSSTA

Bit		Name	Definition
7-6	(0)	DFTHRSH[1:0]	**Data FIFO Threshold Select.** (R/W) The value stored in the DFTHRSH[1:0] bits determines at what quantity of 64-bit data words stored in the data FIFO will activate the DFTHRSH status (bit 2 in the DFSTATUS register).

	Transfer Data FIFO to System		Transfer System to Data FIFO	
DFTHRSH [1:0]	Start	Stop	Start[1]	Stop
00	16 Bytes[2]	Empty[3]	Full-24 Bytes[2]	Full[4]
01	50% Full[2]	Empty[3]	50% Empty[2]	Full[4]
10	75% Full[2]	Empty[3]	75% Empty[2]	Full[4]
11	100% Full[2]	Empty[3]	100% Empty[2]	Full[4]

Example of DFTHRSH[1:0] selections for a 64-bit data FIFO containing 32 64-bit Quad-Words (QWD): Status DFTHRSH will be active at stored QWD levels of (3, 16 24 and 32) for data FIFO to system memory transfers and at stored QWD levels of (29, 16, 8 and 0) for system memory to data FIFO transfers. Note that status DFSDH (for this example) is active only when 30 QWD are stored.

5-0	(0)		Read only. See Configuration register space section for register bit usage. Also see Device ERROR register for software driver monitoring of error status. The leading active edge of each bit becoming active will be latched and will cause IRQA# to be asserted when enabled. These interrupts may be cleared by writing CLRPARERR=1 in the CLRINT register. Clearing the interrupt in this manner does not clear the Configuration STATUS register bits.

[1] Initial start is when HDMAEN becomes active and EMPTY is active.
[2] Status DFTHRSH active.
[3] Status FIFOEMP or HCNT[2:0] value =0h, control HDMAEN inactive or bus-on time expired.
[4] Status FIFOFULL, DFSDH or HCNT[2:0] value =0h, control HDMAEN inactive or bus-on time expired.

Host Control (HCNTRL)

Type: R/W
Address: M87H, DS-87h

The HCNTRL register provides the capability for the software driver to gain latched PAUSE access to registers located on the CIOBUS and normally used by the sequencer. HCNTRL also provides control features that only the software driver may access. HCNTRL may be written to at any time without consideration of the state of the PAUSEACK bit. The HCNTRL register is cleared to 05 when RST# is asserted or CHIPRST is written to (=1).

Writes to HCNTRL that change the state of PAUSE bit followed by a read of HCNTRL will extend the read access until the read value matches the written value (HCNTRL register for this condition is included in the AAP logic timing cycle).

Writes to HCNTRL will cause all active SLP[1:0] bits in the SLEEPCTL register to become inactive.

HCNTRL R/W	
7	
6	POWRDN
5	
4	SWINT
3	HCNTRL3
2	PAUSE[ACK]
1	INTEN
0	CHIPRST[ACK]

Bit		Name	Definition
7	(0)	Not Used	Always reads 0.
6	(0)	POWRDN	**Power Down.** When active (=1), delegates the use of clock input CLKIN and selected PCLK input tree elements. Disables the SCSI bus inputs from external input levels. POWRDN also disables IRQA# from being asserted and limits Device register access to PCI Host only registers. Prior to placing POWRDN in the active state, software must ensure that SCSIENACK, SDMAENACK and HDMAENACK are in the inactive state and PAUSEACK is in the active state from a latched PAUSE condition to prevent errors from being transferred to the SCSI bus, System board, or System memory. Additionally a hardware interlock prevents writing a 0 to the PAUSE bit while writing a 1 to the POWRDN bit.
5	(0)	Not Used	Always reads 0.
4	(0)	SWINT	**Software Interrupt.** When active (=1) will cause the selected IRQA# to be active independent of other interrupt conditions. providing INTEN, MASTEREN and (MSPACEEN and/or ISPACEEN) are active and POWRDN is inactive. SWINT when active does not cause any changes to the INTSTAT register and its state must be monitored by reading the HCNTRL register.
3	(0)	HCNTRL3	A spare R/W capable bit with no current hardware control assignment.
2	(1)	PAUSE[ACK]	**Pause** (located in the write portion of the PAUSE[ACK] bit). When active, requests the sequencer to be paused in an latched condition. When this bit is read, it gives the pause acknowledge status and should be polled to be sure that the sequencer is paused. The driver may start at the address that was paused or at this time change the sequencer program counter (SEQADDR). Clearing this bit will release the sequencer and it will continue at the current value of the program counter. When PAUSEACK is active, access is enabled for the host to Device registers normally restricted when it is not (also see POWRDN bit). This bit also becomes active due to certain hardware conditions, RST# assertion, or writing to CHIPRST. For other conditions, see INTSTAT register and the sequencer section. Note: Software accesses attempted to Device registers normally used by the sequencer or to the External ROM (if enabled) will cause an Automatic Access PAUSE (AAP) of the sequencer which is not latched, with the sequencer continuing its operation automatically after the access is completed. The duration of the AAP is affected by the asynchronous relationship of the software driver access to the internal sequencer cycle, and whether it is to a Device register or the external ROM. The external ROM is slower by design as it should only be read at boot time and it's contents relocated to system memory. For internal Device register access when multiple accesses are required latched PAUSE may be faster.
1	(0)	INTEN	**Interrupt Enable.** The interrupt enable bit when active will allow active interrupt conditions stored in the AIC-7870 to assert the IRQA# pin, providing that the MASTEREN bit in the Configuration Command register is active.
0	(1)	CHIPRST[ACK]	**Chip Reset** (located in the write portion of the CHIPRST[ACK]) This bit when written (=1) will produce the same condition as when RST# is asserted to cause the device (except for the Configuration register space) to enter an initialized state. Following a write to CHIPRST or the desertion of RST#, POR (the internal device reset condition) will remain active for 2-3 CLKIN periods before becoming inactive.
		CHIPRSTACK	**Chip Reset Acknowledge** (located in the read portion of the CHIPRST[ACK] bit).This status bit when active (=1) indicates a write to CHIPRST has occurred or that RST# has been asserted. CHIPRSTACK will remain active until explicitly writing to CHIPRST (=0) (provided that POR has clocked off).

Low Host Address (LHADDR[3:0])

Type: R/W
Address: (HADDLDSEL[1:0] = 0) and M88/89/8A/8Bh, DS-88/89/8A/8Bh

LHADDR[3:0]. The Low Host Address registers contain the low 32-bits of system memory address of the data that will transfer to or from the data FIFO as an active bus master. They perform as count up counters and count up by one for each byte transferred between the device and system memory. LHADDR[3:0] values are issued on the AD[31:00] lines during the PCLK cycle that FRAME# is asserted on for either SAC or DAC transactions. LHADDR[3:0] initialize to zero by CHIPRST or RST#.

Note, after LHADDR[3:0] and HCNT[2:0] have been loaded with the desired values, HDMAEN is placed in the active state allowing transfers to commence. Then, at a later time, should HDMAEN be placed in the inactive state prior to HCNT[2:0] reaching a count of zero, it is disallowed to reload LHADDR[3:0] with new values, placing HDMAEN back in the active state for the same disrupted transfer, unless the data FIFO is reinitialized.

LHADDR0 R/W		LHADDR1 R/W		LHADDR2 R/W		LHADDR3 R/W	
7	LHADDR07	7	LHADDR15	7	LHADDR23	7	LHADDR31
6	LHADDR06	6	LHADDR14	6	LHADDR22	6	LHADDR30
5	LHADDR05	5	LHADDR13	5	LHADDR21	5	LHADDR29
4	LHADDR04	4	LHADDR12	4	LHADDR20	4	LHADDR28
3	LHADDR03	3	LHADDR11	3	LHADDR19	3	LHADDR27
2	LHADDR02	2	LHADDR10	2	LHADDR18	2	LHADDR26
1	LHADDR01	1	LHADDR09	1	LHADDR17	1	LHADDR25
0	LHADDR00	0	LHADDR08	0	LHADDR16	0	LHADDR24

High Host Address (HHADDR[3:0])

Type: R/W
Address: (HADDLDSEL[1:0] = 1) and M88/89/8A/8Bh, DS-88/89/8A/8Bh

HHADDR[3:0]. The High Host Address registers contain the high 32-bits of a system memory 64-bit address of the data that will transfer to or from the data FIFO as an active bus master. They perform only as 32-bit page registers.

When HHADDR[3:0] contain only the value of zero, the AIC-7870 as a PCI master will only issue SACs. When HHADDR[3:0] contain some value other than zero and DACEN is active in the Configuration DEVCONFIG register, the AIC-7870 as a PCI master will issue DACs using the HHADDR[3:0] values in the PCLK cycle following the PCLK cycle that asserted FRAME#. The AIC-7870 as a target does not support DACs. HHADDR[3:0] initialize to zero by CHIPRST or RST#. When DACEN is not active HHADDR[3:0] registers may be used for general-purpose storage if desired.

HHADDR0 R/W		HHADDR1 R/W		HHADDR2 R/W		HHADDR3 R/W	
7	HHADDR07	7	HHADDR15	7	HHADDR23	7	HHADDR31
6	HHADDR06	6	HHADDR14	6	HHADDR22	6	HHADDR30
5	HHADDR05	5	HHADDR13	5	HHADDR21	5	HHADDR29
4	HHADDR04	4	HHADDR12	4	HHADDR20	4	HHADDR28
3	HHADDR03	3	HHADDR11	3	HHADDR19	3	HHADDR27
2	HHADDR02	2	HHADDR10	2	HHADDR18	2	HHADDR26
1	HHADDR01	1	HHADDR09	1	HHADDR17	1	HHADDR25
0	HHADDR00	0	HHADDR08	0	HHADDR16	0	HHADDR24

Host Count (HCNT(n))

Type: R/W
Address: M8C/8D/8E/8Fh, DS-8C/8D/8E/8Fh

HCNT[2:0]. The Host Count registers contain a count of the number of bytes to be transferred between system memory and the data FIFO when the AIC-7870 is an active bus master. HCNT[2:0] perform as count down counters and count down by one for each byte transferred between system memory and data FIFO. Transfers are inhibited when the count value of HCNT[2:0] is zero.

Note: Address M/DS-8Fh is reserved for future expansion. Always reads as 0h and writes are ignored.

HCNT0 R/W		HCNT1 R/W		HCNT2 R/W		RSVD R/W	
7	HCNT07	7	HCNT15	7	HCNT23	7	RSVD
6	HCNT06	6	HCNT14	6	HCNT22	6	RSVD
5	HCNT05	5	HCNT13	5	HCNT21	5	RSVD
4	HCNT04	4	HCNT12	4	HCNT20	4	RSVD
3	HCNT03	3	HCNT11	3	HCNT19	3	RSVD
2	HCNT02	2	HCNT10	2	HCNT18	2	RSVD
1	HCNT01	1	HCNT09	1	HCNT17	1	RSVD
0	HCNT00	0	HCNT08	0	HCNT16	0	RSVD

Interrupt Status (INTSTAT)

Type: R/W
Address: M91hDS-91h

INTSTAT register provides device interrupt status for the driver when an interrupt condition occurs. The INTSTAT register is written to by the sequencer, and may be read from the PCI bus without pausing the sequencer. The sequencer is automatically paused when the SEQINT, SCSIINT or BRKADRINT bit(s) are active. The INTCODE is only valid when SEQINT bit is active. Bits [3:0] may also be individually reset by use of the CLRINT register.

When IRQA# is asserted, the software driver must check both the INTSTAT register values and the PCIERRSTAT bit in the ERROR register to determine the cause(s) of the interrupt. Status bits (DPE, DPR) become active as a result of a PCI master or target transaction and cause an interrupt, provided that PERRESPEN is active and FAILDIS is inactive. Status bits (STA, RTA, RMA, SSI) become active as a result of a PCI master or target transaction and cause an interrupt, provided that FAILDIS is inactive.

INSTAT R/W	
7	INTCODE3
6	INTCODE2
5	INTCODE1
4	INTCODE0
3	BRKADRINT
2	SCSIINT
1	CMDCMPLT
0	SEQINT

Bit		Name	Definition
7-4	(0)	INTCODE(3:0)	**Interrupt Code.** These bits enable a code to be stored to identify the condition causing the SEQINT bit to be active. By convention the INTCODE[3:0] bits are only considered valid when the SEQINT bit is active and should be written in the same write operation that activates SEQINT. See the discussion on interrupts for a definition of this code.
3	(0)	BRKADRINT	**Break Address Interrupt.** This bit becomes active (=1) when using the sequencer firmware breakpoint feature (see registers BRKADDR[1:0] in the sequencer section). Note when the current sequencer instruction breakpoint is an access to an SCB Array address with RAMPSM active (external SRAM). In this case, the sequencer instruction may be stretched while arbitration is being performed for the external SRAM, and a PAUSE request due to an active BRKADRINT will be delayed until arbitration is completed. BRKADRINT is also used for selected error conditions as follows:

(Continued)

Bit		Name	Definition
			1 When the program counter of the sequencer and the break address are equal and the Breakpoint feature is enabled (BRKDIS=0). 2 When ILLOPCODE becomes active (FAILDIS=0). 3 When SQPARERR becomes active (FAILDIS and PERRORDIS=0). 4 When DPARERR becomes active (DPARCKEN=1 and FAILDIS=0). 5 When MPARERR becomes active (MPARCKEN=1 and FAILDIS=0). While the BRKADRINT bit is active it forces the PAUSE bit in HCNTRL register to be active and IRQA# to be asserted when INTEN and MASTEREN are active and POWRDN is inactive. When BRKADRINT is active due to source 1, it may be set inactive by a write to the CLRINT register with CLRBRKADRINT bit 3 (=1). When BRKADRINT is active due to source 2, it may be set inactive by a write to CHIPRST (=1) in the HCNTRL register. When BRKADRINT is active due to source 3-5, it may be set inactive by a write to CLRPARERR (=1) in the CLRINT register. This action will also clear any PCIERRSTAT latched interrupt conditions that may exist, but will not clear the Configuration STATUS1 register bits.
2	(0)	SCSIINT	**SCSI Interrupt.** This bit is latched in the INTSTAT register and is active when there is a catastrophic SCSI event. Causes are SCSI Reset, Parity Error, Selection Time-out, or Unexpected Bus Free. Any interrupt condition in the SCSI section may cause this interrupt if the corresponding interrupt is enabled in SIMODE0 or SIMODE1. When this bit is set, the sequencer is paused immediately. Except when the current sequencer instruction is a access to an SCB Array address with RAMPSM active (external SRAM). In this case the sequencer instruction may be stretched while arbitration is being performed for the external SRAM, and a PAUSE request due to an active SCSIINT will be delayed until arbitration is completed. IRQA# is also asserted when INTEN and MASTEREN are active and POWRDN is inactive. The cause of SCSIINT being active must be cleared, then a write to CLRSCSIINT =1 in the CLRINT register to cause SCSIINT to be read in the inactive state.
1	(0)	CMDCMPLT	**Command Complete Interrupt.** This bit is set active by the sequencer writing to the INTSTAT register with CMDCMPLT (=1) during normal operation after a SCSI command has been completed and the SCB pointer has been loaded on the QOUTFIFO. The sequencer will continue running while this bit is active. While the CMDCMPLT bit is active IRQA# is asserted when INTEN and MASTEREN is active. CMDCMPLT is inactive after a write to the CLRINT register with CLRCMDINT bit (=1).
0	(0)	SEQINT	**Sequencer Interrupt.** This bit is set active by the sequencer writing to the INTSTAT register with SEQINT (=1) when the sequencer requires driver intervention to complete a command or to handle an exception condition. The sequencer is paused by this interrupt immediately (no instruction is performed following the write to set this bit). While SEQINT is active IRQA# is also asserted when INTEN and MASTEREN is active and POWRDN is inactive. SEQINT is inactive after a write to the CLRINT register with CLRSEQINT bit (=1).

Clear Interrupt (CLRINT)

Type: W
Address: M92h, DS-92h

The CLRINT register allows the driver to clear the cause of the interrupt from the device. Selected interrupts are cleared by writing with the desired bit pattern =1. The bits in this register are self-clearing. The sequencer cannot write to this register and the driver may write to it without pausing the sequencer.

Note: When POR and/or RST# is active it also forces CLRPARERR, CLRBRKADRINT, CLRSCSIINT, CLRCMDINT and CLRSEQINT to also be active.

CLRINT W	
7	
6	
5	
4	CLRPARERR
3	CLRBRKADRINT
2	CLRSCSIINT
1	CLRCMDINT
0	CLRSEQINT

Bit		Name	Definition
7-5	(0)	Not Used	Always reads 0.
4	(0)	CLRPARERR	**Clear Parity Errors.** When this bit is written (=1), the SQPARERR, MPARERR and DPARERR bits are cleared if set in the ERROR register. Latched interrupts caused by active PCISTATUS register bits [5:0] will also be cleared, but the PCI Configuration STATUS register bits themselves will not be cleared. To clear these bits, see the *Configuration Status Register* section. This bit will self-clear.
3	(0)	CLRBRKADRINT	**Clear Break Address Interrupt.** When this bit is written (=1), the BRKADRINT bit is cleared in the INTSTAT register. CLRBRKADRINT bit will self-clear and always reads 0. See the INTSTAT register for causes of BRKADRINT being active which may have to be cleared prior to clearing the BRKADRINT bit. ILLOPCODE may only be cleared by writing CHIPRST (=1) or asserting RST#.
2	(0)	CLRSCSIINT	**Clear SCSI Interrupt.** When this bit is written (=1), the SCSIINT bit is cleared in the INTSTAT register. This bit will self-clear.
1	(0)	CLRCMDINT	**Clear Command Complete Interrupt.** When this bit is written (=1), the CMDCMPLT bit is cleared in the INTSTAT register. This bit will self-clear.
0	(0)	CLRSEQINT	**Clear Sequencer Interrupt.** When this bit is written (=1), the SEQINT bit is cleared in the INTSTAT register. This bit will self-clear.

Error (ERROR)

Type: R
Address: M92h, DS-92h

This register reports errors that are catastrophic in nature due to (software/firmware/hardware) error conditions that must be corrected for the device to operate properly. These errors will cause BRKADRINT to be active (except PCI errors) and the sequencer to be paused. Clearing of these bits (except for PCIERRSTAT and ILLOPCODE) requires writing to CLRPARERR for (SQPARERR, MPARERR, DPARERR). Writing to CHIPRST (=1) will clear ILLOPCODE, and asserting input RST# will clear all latched bits.

ERROR R	
7	
6	PCIERRSTAT
5	MPARERR
4	DPARERR
3	SQPARERR
2	ILLOPCODE
1	
0	

Bit		Name	Definition
7	(0)	Not Used	Always reads 0.
6	(0)	PCIERRSTAT	**PCI Error Status.** When active (=1), indicates a PCI error has been detected by the AIC-7870 and is stored in the Configuration STATUS1 register. PCIERRSTAT is the OR of any bit [5:0] active in the Device PCISTATUS register. PCIERRSTAT will become inactive when the PCI errors are inactive. This bit is read only, writing to this register or the CHIPRST bit has no effect. Writing to the Configuration STATUS registers or asserting RST# is required to clear it.
5	(0)	MPARERR	**Memory Parity Error.** When active (=1), indicates an odd-parity error has been detected in the QINFIFO, QOUTFIFO, SCRATCH or SCB Array (internal or external) RAM cells. The MPARCKEN bit in the Device space COMMAND register must be active to enable MPARERR to be become active.
4	(0)	DPARERR	**Data-path Parity Error.** When active (=1), indicates an odd-parity error has been detected in the device internal data path (check logic is located in the PHOST, data FIFO and the SCSI blocks) byte parity. The DPARCKEN bit in the Device space command register must be active to enable DPARERR to be become active.
3	(0)	SQPARERR	**Sequencer Parity Error.** When active (=1), indicates a parity error has been detected in the sequencer control store RAM. PERRORDIS in the Sequencer SEQCTL register must be inactive for SQPARERR to become active.
2	(0)	ILLOPCODE	**Illegal Opcode Error.** When active (=1), indicates a nondefined sequencer instruction has been detected in the sequencer's firmware fetch from its control store RAM.
1-0	(0)	Not Used	Always reads 0.

Data FIFO Control (DFCNTRL)

Type: R/W
Address: M93h, DS-93h

The DFCNTRL register provides data path hardware control. DIRECTIONACK, HDMAENACK, SDMAENACK, SCSIENACK and FIFOFLUSH bits have hardware enforced state changes to ensure proper control of the data path. This control allows several hardware functions to be combined into a single write to the DFCNTRL register. Some bits are self-clearing and some must be cleared by the driver. When RST# or CHIPRST is active all DFCNTRL register bits are forced to zero. data FIFO data path access is determined by the state selections of DIRECTIONACK, HDMAENACK and SDMAENACK bits in the DFCNTRL register.

DFCNTRL R/W	
7	
6	WIDEODD
5	SCSIEN[ACK]
4	SDMAEN[ACK]
3	HDMAEN[ACK]
2	DIRECTION[ACK]
1	FIFOFLUSH[ACK]
0	FIFORESET

Bit		Name	Definition
7	(0)	Not Used	Always reads 0.
6	(0)	WIDEODD	**Wide Odd control.** This bit has no effect except during Wide SCSI transfer with an odd byte count. When active (=1), the last byte (low byte of a 16-bit transfer) is held in the SCSI block until the first byte of the next transfer is received, and then the data transfer is continued as a 16-bit transfer. This maintains data continuity across Scatter/Gather boundaries. WIDEODD should always be active for Scatter/Gather transfers when 16-bit SCSI transfers are used, except after the last list element then it should be turned off, enabling possible last low byte being held, to be sent with a dummy high byte for the last 16-bit transfer.
5	(0)	SCSIEN[ACK]	**SCSI Transfer Enable/SCSI Transfer Enable Acknowledge.** When this bit is active (=1) it enables transfers between a SCSI bus and a SCSI FIFO. Clearing this bit will cleanly halt the transfer by preventing another ACK to the SCSI bus. Reading this bit (SCSIENACK) provides status which indicates the state of the hardware. When this bit is cleared, it must be read back as zero before the transfer is guaranteed to have halted. Synchronous data-in transfers to the SCSI FIFO will always be enabled when the synchronous offset value programmed in SCSIRATE is nonzero and the SCSI bus is in Data-in phase.
4	(0)	SDMAEN[ACK]	**SCSI DMA Enable/SCSI DMA Enable Acknowledge.** When this bit is active (=1) it enables transfers between the SCSI block and the data FIFO. Reading this bit (SDMAENACK) provides status which indicates the state of the hardware. When this bit is cleared it must be read back as zero before the transfer is guaranteed to have halted.

(Continued)

Bit		Name	Definition
3	(0)	HDMAEN[ACK]	**Data FIFO DMA Enable/Data FIFO DMA Enable Acknowledge.** When this bit is active it enables the PCI host interface to transfer data to or from system memory. The Host Address and Host Count registers must be set up and the data FIFO initialized prior to activating this bit. Clearing this bit will halt transfers without losing data, status, or byte count. Transfers may be continued after halting. Reading this bit (HDMAENACK) provides status which indicates the state of the hardware. When this bit is cleared it must be read back as zero before the transfer is guaranteed to have halted.
2	(0)	DIRECTION[ACK]	**Data Path Direction Acknowledge.** This bit when active (=1) will condition data transfers to be from the PCI bus (system memory) or from the (sequencer cell for DFDAT write), to the data FIFO and from the data FIFO to the (SCSI cell) or to the (sequencer cell for DFDAT read). The Direction bit when not active (=0) will condition data transfers to be from the SCSI cell or from the (sequencer cell for DFDAT write), to the data FIFO and from the data FIFO to the PCI bus (system memory) or to the (sequencer cell for DFDAT read). When both HDMAENACK and SDMAENACK are inactive, the sequencer cell may both read and write DFDAT without regard for the state of the DIRECTION bit. The state of DIRECTIONACK will not change unless the enable bits (bits 3, 4 and 5) are cleared.
1	(0)	FIFOFLUSH[ACK]	**Data FIFO Flush.** During a SCSI bus to PCI bus transfer (a read operation as initiator), FIFOFLUSH may be set (manual flush) to force the remaining bytes in the data FIFO to be sent to the PCI host memory. If FIFOEMP is active, attempts to set this bit active will have no effect. This bit is self-clearing and has no effect during a PCI bus to SCSI bus operation. An Autoflush to force the remaining bytes to be transferred will occur by the hardware when STCNT counts down to zero or a SCSI phase change occurs (providing AUTOFLUSHDIS=0 in the SBLKCTL register). When this bit is read as a one, it indicates a flush operation is pending or in operation due to either a firmware or hardware flush. It will read as a zero when the flush operation is completed. During sequencer writes to the data FIFO, setting this bit will force the host or SCSI blocks to recognize a partial quad word load made by the sequencer. This feature allows any number of bytes to be loaded by the sequencer without considering byte alignment in the data FIFO. Note, manual flush using the FIFOFLUSH write bit should not be used if WIDEODD bit is active.
0	(0)	FIFORESET	**Data FIFO Reset.** This bit when written (=1) will force the data FIFO status to reflect that the data FIFO is empty. This bit is self-clearing. FIFORESET must be written to after loading LHADDR0 register in order to load the data path byte offset pointers in the (data FIFO, PHOST and SCSI blocks) which are determined from LHADDR[02:00] in preparation for a new data transfer operation.

Data FIFO Status (DFSTATUS)

Type: R
Address: M94h, DS-94h

The DFSTATUS register contains data path status.

DFSTATUS R	
7	
6	DFCACHETH
5	FIFOQWDEMP
4	MREQPEND
3	HDONE
2	DFTHRSH
1	FIFOFULL
0	FIFOEMP

Bit		Name	Definition
7	(0)	Not Used	Always reads 0.
6	1	DFCACHETH	**Data FIFO Cache Threshold Status.** This is active (=1) to indicate that sufficient space exists in the data FIFO to store at least one more selected cache size transfer for data path direction from PCI bus to the data FIFO, or that sufficient data is available in the data FIFO for at least one more selected cache size transfer when the data path direction is from the data FIFO to the PCI bus. See the Configuration CACHESIZE register for additional information.

CACHESIZE Value	DFCACHETH Action
[7:0]	
0 - 7	= 1 DIRECTION bit =1 (space for 8 DWDs (32 Bytes) or greater exists) or DIRECTION bit =0 (4 DWDs (16 Bytes) or more are stored)
8 - 15	= 1 DIRECTION bit =1 (space for 8 DWDs (32 Bytes) or greater exists) or DIRECTION bit =0 (8 DWDs (32 Bytes) or more are stored)
16 - 31	= 1 DIRECTION bit =1 (space for 16 DWDs (64 Bytes) or greater exists) or DIRECTION bit =0 (16 DWDs (64 Bytes) or more are stored)
32 - 255	= 1 DIRECTION bit =1 (space for 32 DWDs (128 Bytes) or greater exists) or DIRECTION bit =0 (32 DWDs (128 Bytes) or more are stored)

5	(1)	FIFOQWDEMP	**Data FIFO Quad Word Empty.** This is active (=1) when the data FIFO does not contain a complete quad word of 64-bits (QWD). Partially stored QWD (one to seven bytes) will not cause FIFOQWDEMP to become inactive. FIFOQWDEMP status is used by the SCSI and host blocks to determine when to read or write data FIFO data. When the FIFORESET bit is active or CHIPRST is written (=1) or RST# is asserted, this bit is forced to be active.

(Continued)

Bit		Name	Definition
4	(0)	MREQPEND	**Memory Request Pending.** This bit is active when the host block has reached a condition (a master transfer has been set up and the proper data FIFO status becomes active, or FIFOFLUSHACK becomes active for SCSI reads) which requires a data transfer on the PCI bus. The actual request for the bus (PREQ# asserted) is still subject to other conditions and may not be asserted at the time MREQPEND becomes active. MREQPEND is cleared when there is no requirement for a host transfer, when HDONE is active, when FIFORESET is active, when CHIPRST is written (=1), or RST# is asserted. Status MREQPEND is set active by the DFTHRSH[1:0] selected start condition (DFTHRSH status) or DFSXDONE (SXFERDONE from the SCSI block, or FIFOFLUSH in DFCNTRL register) active with FIFOQWDEMP inactive. MREQPEND is set inactive by the selected stop condition (HCNT, data FIFO empty for system memory writes, data FIFO full for system memory read, HDMAEN set inactive).
3	(0)	HDONE	**Host Done** status bit. When active (=1), indicates that the count previously stored in HCNT[2:0] has expired (count=0) and the last transfer between system memory and the data FIFO has completed, including any temporary storage in the AIC-7870 host block.
2	(0)	DFTHRSH	**Data FIFO Threshold Status** bit. When active (=1), indicates that the quantity of quad data words stored in the data FIFO equals the value selected by DFTHRSH[1:0] located in the PCISTATUS register. When DFTHRSH becomes active, MREQPEND is set active to enable the device to request PCI bus master status (providing HDMAEN is active, HCNT is not zero, MASTEREN is active in the Configuration Command register and PCI status bits RMA and RTA are not active) to either empty the data FIFO by transfers to system memory or to fill the data FIFO by transfers from system memory.
1	(0)	FIFOFULL	**Data FIFO Full Status** bit. When active (=1) indicates that all byte locations in the data FIFO contain data. Data must not be written to the data FIFO when FIFOFULL is active. Note: The current full position for normal operation in the data FIFO will change depending on the number of data bytes read from the data FIFO prior to writing it. FIFOFULL will be forced to be inactive when FIFORESET is active, CHIPRST is written (=1), or RST# is asserted.
0	(1)	FIFOEMP	**Data FIFO Empty Status** bit. When active (=1) indicates that no data bytes are stored in the data FIFO. The data FIFO must not be read from when FIFOEMP is active. Note: The current empty position for normal operation in the data FIFO will change depending on the number of data bytes written to the data FIFO prior to reading it. FIFOEMP will be forced to be active when FIFORESET is active, CHIPRST is written (=1) or RST# is asserted.

[1] When RST# is asserted or CHIPRST is written (=1), DFCACHETH is forced to be inactive. When the FIFORESET bit is written to and DIRECTION is inactive, DFCACHETH will be forced to be inactive. When the FIFORESET bit is written to and DIRECTION is active, DFCACHETH will be forced to be active.

Queue Out FIFO (QOUTFIFO)

Type: R/
Address: M9Dh, DS-9Dh

This register is written by the sequencer with the SCB pointer value of the SCB which was just completed in the SCB Array. Writing to QOUTFIFO puts a value onto the queue. The QOUTFIFO can queue sixteen SCB command values when RAMPSM is inactive and 255 SCB command values when RAMPSM is active. The controlling host driver will read the QOUTFIFO which will remove one value from the queue. The QOUTFIFO may be read by the host driver without access or latch pausing of the sequencer. The status of the QOUTFIFO is given in QOUTCNT register. Writes when QOUTCNT is 255 and reads when QOUTCNT is 0 will be ignored. When RAMPSM is active, SRAM external to the AIC-7870 is used for SCB data storage and the SCB SRAM internal to the AIC-7870 is used for the QOUTFIFO data storage.

QOUTFIFO R/	
7	QOUT7
6	QOUT6
5	QOUT5
4	QOUT4
3	QOUT3
2	QOUT2
1	QOUT1
0	QOUT0

Bit		Name	Definition
7:4	(0)	QOUT[7:4]	**Queue Out.** When RAMPSM is inactive, always reads 0. When RAMPSM is active, these bits store the pointers to the SCB commands that have been completed and are only valid when QOUTCNT read value is not equal to zero.
3:0	(0)	QOUT[3:0]	**Queue Out.** These bits store the pointers to the SCB commands that have been completed and are only valid when QOUTCNT read value is not equal to zero.

Queue Out Count (QOUTCNT)

Type: R
Address: M9Eh, DS-9Eh

The QOUTCNT register contains the count of the number of entries stored into the QOUTFIFO. The QOUTCNT read value of 0 indicates no valid data is stored in or all entries have been read out of the QOUTFIFO. The QOUTCNT may be read by the driver without latch pausing the sequencer. Valid QOUTCNT range is always 1 to 255.

QOUTCNT R	
7	QOUTCNT7
6	QOUTCNT6
5	QOUTCNT5
4	QOUTCNT4
3	QOUTCNT3
2	QOUTCNT2
1	QOUTCNT1
0	QOUTCNT0

Bit		Name	Definition
7:5	(0)	QOUTCNT[7:5]	**Queue Out Count.** When RAMPSM is inactive, QOUTCNT[7:5] only indicate rollover status of bits QOUTCNT[4:0]. When RAMPSM is active these bits identify the number of QOUTFIFO locations that contain valid data.
4:0	(0)	QOUTCNT[4:0]	**Queue Out Count.** These bits identify the number of QOUTFIFO locations that contain valid data.

Special Function (SFUNCT)

Type: R/W
Address: M9Fh, DS-9Fh

This register provides selection of certain sections of the chip for test purposes. The pins in some cases are redefined to provide ample I/O for chip test. Identified selections in this register may also be used for diagnostics prior to starting normal operation. Redefining of pins that would affect system operation is prevented by nonnormal use of certain pins.

SFUNCT R/W	
7	SFUNCT2
6	SFUNCT1
5	SFUNCT0
4	TESTRAM
3	TESTHOST
2	TESTSEQ
1	TESTFIFO
0	TESTSCSI

Bit		Name	Definition
7:5	(0)	SFUNCT[2:0]	**Special Function.** Used to select hardware test features. See SFUNCT register bits 4:0 for usage. Currently SFUNCT[2:0] states only have meaning when defined with a SFUNCT register bit 4:0 in the active state. These bits may be used for software flags when bits [4:0] are equal to zero.
4	(0)	TESTRAM	When active (=1) enables stress testing of the sequencer RAM.
3	(0)	TESTHOST	When active (=1) selects the host block for testing.
2	(0)	TESTSEQ	When active (=1) selects the sequencer block for testing.
1	(0)	TESTFIFO	When active (=1) selects the data FIFO block for testing.
0	(0)	TESTSCSI	When active (=1) selects the SCSI block for testing.

Sleep Control (SLEEPCTL)

Type: R/W
Address: M1Ch DS-1Ch

The SLEEPCTL register provides the control for the Host driver to disable the SLEEP mode from becoming active. It also provides control for the sequencer to activate the SLEEP mode when it desires to wait for one of the selected wake up status to become active. The SLEEP mode is similar to the PAUSE mode and reduces the power utilized by the AIC-7870 when the sequencer is running (i.e., sequencer logic and RAM is not being clocked and the CIOBUS is idle).

The SLP[1:0] bits are enable bits that allow selection of wake up status that are desired for the current sleep period. As a result of the sequencer writing to make an SLP[1:0] bit active, it will cause the sequencer to go to sleep, provided that the SLEEP mode has not been disabled by SLEEPDIS. When one of the selected statuses becomes active, the sequencer will wake up and automatically begin to process the next instruction. When SLEEPDIS is active, the sequencer will not go to sleep and will continue with the next instruction without delay. When the selected status becomes active, the SLP[1:0] bits will be set inactive. SLP[1:0] will also be set inactive whenever the Host driver writes to set the PAUSE bit inactive. Thus, no additional Host driver access is required to clear possible sleep conditions of the sequencer when a new SCB has been loaded in the QINFIFO. This action will occur even when the PAUSE bit is not active.

SLEEPCTL R/W	
7	SLEEPDIS
6	
5	
4	
3	
2	
1	SLP1
0	SLP0

Bit		Name	Definition
7	(0)	SLEEPDIS	**Sleep Disable.** When active (=1), prevents the SLP[1:0] bits from being set active by the sequencer. In this case, the sequencer will continue the next instruction without delay, the same as if it was just woken up.
6-2	(0)	Not Used	Always reads 0.
1	(0)	SLP1	When active (=1), enables an active DMADONE or PHASEMIS status to wake up the sequencer.
0	(0)	SLP0	When active (=1), enables an active SELDO or SELDI status to wake up the sequencer.

SCB Array Device Space Register Definition

SCB Pointer (SCBPTR)

Type: R/W
Address: M90h, DS-90h

The SCB Pointer register provides the page address to the SCB Array. The data value loaded in this register selects a page of 32 registers, within the SCB address range, which contain a description of an executable command. Changing this value during execution will not alter any data, but will address a different page of the array. When FFh is written to the SCBPTR register the value returned upon reading it will be FFh (255 SCB pages) whether or not RAMPSM is active.

SCBPTR R/W	
7	SCBVAL7
6	SCBVAL6
5	SCBVAL5
4	SCBVAL4
3	SCBVAL3
2	SCBVAL2
1	SCBVAL1
0	SCBVAL0

Bit		Name	Definition
7-0	(0)	SCBVAL[7:0]	These read/write bits are used for SCB (32-byte) page selection.

SCB Counter (SCBCNT)

Type: R/W
Address: M9Ah, DS-9Ah

The SCBCNT register provides two modes for SCB Array page byte address access.

- Direct SCB Array page byte addressing for random access within the page.
- Indirect SCB Array page byte addressing by use of SCBCNT[4:0] where each access causes the value in SCBCNT[4:0] to increment for linear access within the selected page. When SCBCNT[4:0] are being used both the read and write addresses of the SCB Array are the same address and when not used (for sequencer access only) may be different addresses in the page.

SCBCNT R/W	
7	SCBAUTO
6	RSVD
5	RSVD
4	SCBCNT4
3	SCBCNT3
2	SCBCNT2
1	SCBCNT1
0	SCBCNT0

Bit		Name	Definition
7	(0)	SCBAUTO	**SCB Auto Address.** When active (=1), enables SCBCNT[4:0] to supply the offset address into the selected SCB Array page in place of the address supplied by the host or sequencer. When SCBAUTO is inactive, the value stored in SCBCNT[4:0] is not used. When SCBAUTO is active, accesses to the SCB Array may be made repeatedly to the same address, which must be within the normal SCB Array address range (M/DS-A0 - M/DS-BFh). When SCB access is from the sequencer, both read and write accesses are bytes without regard for the state of SCBAUTO. When SCB access is from the PCI bus and SCBAUTO is inactive, the SCB access is byte access only for both read or writing. When SCB access is from the PCI bus and SCBAUTO is active, SCB read access is byte access only and SCB write access is by byte or double word. When only one CBE[3:0]# is asserted, the write will be a byte access. When all CBE[3:0]# are asserted, the write will be a double word access. Each double word transferred will be byte unpacked and sent to the SCB Array address selected by the SCBCNT[1:0] value. TRDY# will remain deasserted until all four bytes are written to the SCB Array. SCBCNT[1:0] will increment with each byte transferred. When RAMPSM is inactive, SCBs are being stored internally in the AIC-7870 and the transaction may be a burst without disconnection's between Data phases. When RAMPSM is active, SCBs are being stored externally to the AIC-7870 and the transaction may be a burst with disconnection's between Data phases due to the longer access time of the external SRAM.
6-5	(0)	RSVD	Always reads 0.
4-0	(0)	SCBCNT[4:0]	**SCB Counter.** Stores the address of the byte location that will be accessed in the selected SCB Array page. The value in SCBCNT[4:0] is only used when SCBAUTO is active. Note, SBCNT[4:2] are used to select the SCB array double word and SCBCNT[1:0] are used to select the double word byte to be written in the SCB Array. After the data byte is written when SCBCNT[1:0] (=3), TRDY# will be asserted to complete transfer of the current PCI transaction Data phase.

Queue In FIFO (QINFIFO)

Type: R/W
Address: M9Bh, DS-9Bh

This register is written by the controlling host driver with the SCB pointer value of the SCB which was just loaded in the SCB Array SRAM. Writing to QINFIFO puts a value onto the queue. The QINFIFO can queue 16 SCB commands when RAMPSM is inactive and 255 SCB commands when RAMPSM is active. The sequencer will read the QINFIFO which will remove one value from the queue. The status of the QINFIFO is given in QINCNT register. Writes when QINCNT is maximum 255 and reads when QINCNT is 0 will be ignored. When RAMPSM is inactive, SCB SRAM internal to the AIC-7870 is used for SCB data storage. When RAMPSM is active, SRAM external to the AIC-7870 is used for the queue data storage and the SCB SRAM internal to the AIC-7870 is used for the QINFIFO data storage.

QINFIFO R/W	
7	QIN7
6	QIN6
5	QIN5
4	QIN4
3	QIN3
2	QIN2
1	QIN1
0	QIN0

Bit		Name	Definition
7:4	(x)	QIN[7:4]	**Queue In FIFO.** When RAMPSM is inactive, always read 0, When RAMPSM is active, these bits store the pointers to the SCBs which have been loaded in the SCB Array and are only valid when QINCNT read value is not equal to zero.
3-0	(x)	QIN[3:0]	**Queue In FIFO.** These bits store the pointers to the SCBs which have been loaded in the SCB Array and are only valid when QINCNT read value is not equal to zero.

Queue In Count (QINCNT)

Type: R
Address: M9Ch, DS-9Ch

The QINCNT register contains the count of the number of entries stored in the QINFIFO. The QINCNT read value of 0 indicates no valid data is stored in or all entries have been read out of the QINFIFO. Valid QINCNT count range is always 1 to 255.

QINCNT R/W	
7	QINCNT7
6	QINCNT6
5	QINCNT5
4	QINCNT4
3	QINCNT3
2	QINCNT2
1	QINCNT1
0	QINCNT0

Bit		Name	Definition
7:5	(0)	QINCNT[7:5]	**Queue In Count.** When RAMPSM is inactive, these bits only indicate rollover status of bits QINCNT[4:0]. When RAMPSM is active, these bits identify the number of QINFIFO locations that contain valid data.
4:0	(0)	QINCNT[4:0]	**Queue In Count.** These bits identify the number of QINFIFO locations that contain valid data.

Data FIFO Device Space Register Definition

Data FIFO Write Address (DFWADDR)

Type: R/W
Address: M95h, DS-95h

This register contains the address (write pointer) of the current data FIFO location to be written to. Each value points to a 64-bit Quad Word (QWD) location in the data FIFO. The DFWADDR register is automatically incremented when the high byte in a QWD is written through the DFDAT register or data path transfers between the PCI and SCSI buses. The starting byte location (byte 0, 1, 2 . . .7) of a QWD is determined by the starting value of LHADDR[02:00] and activation of FIFORESET. Note, for DFWADDR increment with each write from the DFDAT register, see SFUNCT register. See the section on reading and writing the data FIFO for more information.

DFWADDR R/W	
7	CACHETHLA
6	
5	DFWADDR5
4	DFWADDR4
3	DFWADDR3
2	DFWADDR2
1	DFWADDR1
0	DFWADDR0

Bit		Name	Definition
7	(0)	CACHETHLA	**Cache Threshold Look Ahead** (read only). Provides capability to test this data FIFO status bit. CACHETHLA is active when either data or space for the selected cache size is available in the data FIFO plus 16 bytes of look ahead.
6	(0)	Not Used	Always reads 0.
5	(0)	DFWADDR5	**Data FIFO Write Address 5.** Provides access to the data FIFO write address roll-over status bit which is used to determine data FIFO full and empty status.
4:0	(0)	DFWADDR[4:0]	**Data FIFO Write Address [4:0].** The QWD address lines to the data write port of the data FIFO.

Reserved Address

Type: R/W
Address: M96h, DS-96h

RESERVED R	
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	RSVD
1	RSVD
0	RSVD

Bit		Name	Definition
7:0	(0)	RSVD	Always reads 0 after RST# assertion or a write to CHIPRST=1, otherwise will read the previous read access value from the CSDAT bus.

Data FIFO Read Address (DFRADDR)

Type: R/W
Address: M97h, DS-97h

This register contains the address (read pointer) of the current data FIFO location to be read from. Each value points a 64-bit Quad Word (QWD) location in the data FIFO. The DFRADDR register is automatically incremented when the high byte in a QWD is read through the DFDAT register or data path transfers between the PCI and SCSI buses. The starting byte location (byte 0, 1, 2 . . .7) of a QWD is determined by the starting value of LHADDR[02:00] and activation of FIFORESET. See the section on reading and writing the data FIFO for more information.

DFRADDR R/W	
7	DFDSH
6	
5	DFRADDR5
4	DFRADDR4
3	DFRADDR3
2	DFRADDR2
1	DFRADDR1
0	DFRADDR0

Bit			Definition
7	(0)	DFSDH	**Data FIFO Stored Data High** (read only). Provides capability to test this data FIFO status bit. DFSDH is only active when the stored data level in the data FIFO is full less 16 bytes.
6	(0)	Not Used	Always reads 0.
5	(0)	DFRADDR5	**Data FIFO Read Address 5**. Provides access to the data FIFO read address roll-over status bit which is used to determine data FIFO full and empty status.
4:0	(0)	DFRADDR[4:0]	**Data FIFO Read Address [4:0].** The QWD address lines to the data read port of the data FIFO

Reserved Address

Type: R/W
Address: M98h, DS-98h

RESERVED R	
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	RSVD
1	RSVD
0	RSVD

Bit		Name	Definition
7:0	(0)	RSVD	Always reads 0 after RST# assertion or a write to CHIPRST=1, otherwise will read the previous read access value.

Data FIFO Data (DFDAT)

Type: R/W
Address: M99h, DS-99h

This register stores data into the data FIFO using register DFWADDR0 when written and reads data from the data FIFO using register DFRADDR0 when read. Before writing or reading the data FIFO with DFDAT register or data path operations from the PCI and/or SCSI buses, the host address registers must be loaded and FIFORESET (bit 0, DFCNTRL) should be written to initialize the correct byte offset in the data FIFO starting QWD. DFWADDR0 and DFRADDR0 registers may be adjusted after the FIFORESET if a specific data FIFO QWD location is desired. For more information, see reading and writing the data FIFO. For starting location, if not at QWD boundary, see DFWADDR0 and DFRADDR0.

DFDAT R/W	
7	DFDAT7
6	DFDAT6
5	DFDAT5
4	DFDAT4
3	DFDAT3
2	DFDAT2
1	DFDAT1
0	DFDAT0

Memory Port Device Space Register Definition

Serial EEPROM Control (SEECTL)

Type: R/W
Address: M1Eh, DS-1Eh

The Serial EEPROM Control register provides the capability to control reading and writing an external serial 1-bit EEPROM device that contains a 4-pin interface (typical devices are NM93C06/C46/C56/C66). The SEEPROM is interconnected to the AIC-7870's Memory Port which may be shared with an SRAM, a ROM/EEPROM and board logic devices. The timing required by the external SEEPROM is a function of a software routine that matches the device's timing. Due to the slow clock rate, typically 1MHz maximum, a hardware timer is provided that may be accessed to ease software development and provide portability.

	SEECTL R/W
7	EXTARBACK
6	EXTARBREQ
5	SEEMS
4	SEERDY
3	SEECS
2	SEECK
1	SEEDO
0	SEEDI

Bit		Name	Definition
7	[1]	EXTARBACK	**External Arbitration Acknowledge.** This read only bit enables the sequencer or host driver to determine the external level of input pin EXTARBACK#. See the Memory Port interface section for additional information. EXTARBACK# contains a weak internal pull-up and depending on the application may require an external pull-up resistor.
6	(1)	EXTARBREQ	**External Arbitration Request.** This read only bit enables the sequencer or host driver to determine the level of output pin EXTARBREQ#. See the *Memory Port Interface* section for additional information. EXTARBREQ# requires an external pull-up resistor for multiuser applications.
5	(0)	SEEMS	**Serial EEPROM Mode Select.** This read/write bit when active (=1) generates a request to the Memory Port for access to the external SEEPROM/board logic devices. When the Memory Port is not busy servicing another request, sequencer/host to SRAM or host to ROM/EEPROM, then SEEMS active will cause EXTARBREQ# to be asserted. When EXTARBACK# becomes asserted in response to EXTARBREQ# assertion, then the Memory Port is reconfigured to allow SEECTL bits to control MD[2:0] and SEECS for the SEEPROM/board logic device access. Note: Once EXTARBACK# is asserted so that the AIC-7870 drives the Memory Port, no other device may gain access to the external devices until SEEMS is inactive. Prior to beginning the external access process verify that the Memory Port access has been granted by sampling SEERDY active after a write to set SEEMS active.
4	(0)	SEERDY	**Serial EEPROM Ready.** This read only bit provides a hardware timer that may be used instead of a software timer when accessing the external SEEPROM/board logic devices. Each time the SEECTL register is written to, SEERDY will become inactive and after an 800 nsec delay will become active (=1). In addition, SEERDY provides the initial indication that the Memory Port arbitration has been completed following setting SEEMS active. In this case SEERDY will not become active after the normal 800 nsec time-out, but will stay inactive until EXTARBACK# is sampled asserted in response to EXTARBREQ# being asserted as a result of SEEMS being active. Note: The Memory Port outputs will not become driven until EXARBACK# is asserted after SEEMS is set active. The access acknowledge and independent timing are the only hardware function performed by SEERDY.
3	(0)	SEECS	**Serial EEPROM Chip Select.** This read/write bit is used to control the AIC-7870 output SEECS. The SEECS bit being active (=1) can only assert the AIC-7870 output SEECS when SEEMS is active and has completed arbitration for Memory Port use. SEECS is also used with the BRDCTL register when both SEEPROM and board logic devices are present. SEECS requires an external pull-down resistor.
2	(0)	SEECK	**Serial EEPROM Clock.** This read/write bit controls the state of MD2 which is connected to the shift clock input of the SEEPROM. When SEECK and SEEMS are active (=1), MD2 will be at a high level. This bit may also be utilized for board logic devices as an output.
1	(0)	SEEDO	**Serial EEPROM Data Out.** This read/write bit controls the state of MD1 which is connected to the data input of the SEEPROM. When SEEDO and SEEMS are active (+1), MD1 will be at a high level for writing a 1 bit into the EEPROM. This bit may also be utilized for board logic devices as an output.
0	[2]	SEEDI	**Serial EEPROM Data In.** This read only bit is used to access data from the SEEPROM. Its value reflects the value of MD0 which is connected to the data output of the SEEPROM and SEEMS is active. This bit may also be utilized for board logic devices as an input.

[1] Level is sampled by reset, EXTARBACK# is expected to be high for multiuser configurations and low for single user configuration.
[2] Level following reset is expected to be high when an external pull-up is provided otherwise unknown.

Board Control (BRDCTL)

Type: R/W
Address: M1Dh, DS-1Dh

The Board Control register provides the capability to control reading and writing of external device(s) interconnected to the AIC-7870's Memory Port which may be shared with an SRAM, ROM/EEPROM and SEEPROM. The timing provided to the external devices is a function of a software routine that matches the device's timing. The SEECTL bits as indicated may also be used for board logic control if desired.

BRDCTL R/W	
7	BRDDAT7
6	BRDDAT6
5	BRDDAT5
4	BRDSTB -
3	BRDCS -
2	BRDRW
1	BRDCTL1
0	BRDCTL0

- Note the pin output state is the inversion of the register state.

Bit		Name	Definition
7-5	(0)	BRDDAT[7:5]	**Board Data [7:5].** Bits are read/write data bits which are only connected to MD[7:5] when SEEMS is active and EXTARBACK# is asserted. These bits are write bits when BRDRW is inactive (=0) and their written value will be asserted on MD[7:5] pins. These bits are read bits when BRDRW is active and the values read will be those present on MD[7:5] pins.
4	(0)	BRDSTB	**Board Strobe.** Connected to MD4 when SEEMS is active and EXTARBACK# is asserted. When BRDSTB is active (=1), MD4 will be asserted low (=0). The normal use is to store the desired write data in BRDDAT[7:5] and a one in BRDCS. Then store a one in BRDSTB to activate the external MD4 at a low level (starts the strobe). Then store a zero in BRDSTB (ends the strobe) to write the data into the external logic
3	(0)	BRDCS	**Board Chip Select.** This read/write bit is connected to MD3 when SEEMS and EXTARBACK# is asserted. When BRDCS bit is active (=1) output MD3 is asserted low (=0).
2	(0)	BRDRW	**Board Read/Write.** This read/write bit controls the output state of MDP when SEEMS is active and EXTARBACK# is asserted. The state of BRDRW also controls the data direction of BRDDAT[7:5], when BRDRW is inactive (=0) BRDDAT[7:5] are outputs and MDP is asserted low (=0).
1-0	(0)	BRDCTL[1:0]	Always read as 0 and writes are ignored.

5 Functional Description

About This Chapter

Read this chapter to find out

- A description of SCSI transfer control logic
- Features of the sequencer (SCSI PhaseEngine)
- Descriptions of the PCI Bus Commands
- Information about Memory Port Arbitration

▼▼▼▼▼ 5

SCSI

Wide (16-bits)/Narrow (8-bits) Bus

The AIC-7870 contains one SCSI bus, SCSI channel A, which may perform 8-bit (narrow) or 16-bit (wide) data transfers in Single-ended or Differential (with external components) mode at rates up to 10 Mtransfers/sec. When channel A is in Wide mode the state of WIDEXFER in SCSIRATE determines whether 8-bit or 16-bit SCSI transfers will take place on the Wide bus. Note that REQB and ACKB are not supported, so a P-type cable must be used to implement a Wide bus. When Narrow mode is selected, data bits SCD[15:8] and SCDPH are never asserted. Internal operation of the AIC-7870 is the same for either Single-ended or Differential mode as the outputs for control of the external differential components operate all the time even if the components are not present.

During the reset state, there is a mechanism to indicate whether the device is connected to a Wide bus. This is accomplished by grounding pin WIDEPS#.

The data interface to the SCSI control logic within the device is 16 bits. Sixteen-bit Wide transfers that are odd length and/or start on odd segment boundaries are handled automatically by the SCSI transfer control logic. Since the first word from the host of an odd boundary write will have a *dummy* low byte, the SCSI transfer logic reads only the high byte of the first word from the data FIFO and subsequently packs the remaining bytes to be sent out to the SCSI bus. On an odd boundary read, the SCSI transfer logic will write the first (low) byte of the first word transferred from the SCSI bus to the high byte of the data interface. This will cause the first word to the data FIFO to have a *dummy* low byte. For odd length writes, the last byte will be placed in the low byte position of the last word and a dummy byte will be placed in the high byte position. When a Scatter/Gather segment is an odd number of bytes, the last byte of the transfer may be held off by setting the WIDEODD control bit. This will prevent the sending of the last byte to the wrong segment, but it will be held until the device is set up for the next segment and data transfers have started. Transfers that are odd length reads to even boundaries or even length reads to odd boundaries leave a residual byte at the end of the transfer. This last byte is transferred to the data FIFO in the low byte position of the last word with a *dummy* byte in the high position.

The SCSI transfer control logic automatically handles the conditions described above by examining the state of DIRECTION, STCNT, and SHADDR at the beginning of the transfer. Table 5-1 illustrates how the decisions are made: (DIR 1 = write, 0 = read; CNT = bit 0 of STCNT; ADDR = bit 0 of SHADDR)

Table 5-1. SCSI Transfer Control Logic

DIR W=To SCSI R= To Data FIFO	CNT	ADDR	First Byte Alignment	Last Byte Alignment
1 (W)	0	0	Low Byte	High Byte
1 (W)	0	1	High Byte	Low Byte, dummy High
1 (W)	1	0	Low Byte	Low Byte, dummy High
1 (W)	1	1	High Byte	High Byte
0 (R)	0	0	Low Byte	High Byte
0 (R)	0	1	High Byte	Low Byte, dummy High
0 (R)	1	0	Low Byte	Low Byte, dummy High
0 (R)	1	1	High Byte	High Byte

Arbitration on a Wide bus is handled in a manner in accordance with the SCSI-2 specification. In order to accommodate 8-bit devices on a 16-bit Wide bus, the priority scheme is as follows:

	Highest ID	Lowest ID
P- Cable	7 ---> 0	15 ---> 8

This scheme insures that 8-bit devices are given fair consideration during arbitration.

SCSI Termination Power Control

This feature provides the capability to enable or disable the external SCSI bus termination power source by use of the STPWCTL output pin.

The STPWLEVEL bit (bit 1, DEVCONFIG register) selects the active/inactive polarity of the STPWCTL output pin.

- If STPWLEVEL=0, then a high-level is selected for the active state of output pin STPWCTL.
- If STPWLEVEL=1, then a low-level is selected for the active state of output pin STPWCTL. If a low-level active state is desired for output pin STPWCTL, then STPWLEVEL should always be written before writing to STPWEN bit after a RST# assertion.

The STPWEN bit (bit 0, SXFRCTL1 register) selects the actual active/inactive state of the STPWCTL output pin. If STPWEN=1, then the STPWCTL output pin will be asserted to the level selected by STPWLEVEL.

Note: Additional control may be provided by use of the BRDCTL register feature.

Table 5-2. SCSI Termination Power Control

RST3 Input Pin	CHIPRST Bit	STPWLEVEL Bit	STPWEN Bit	STPWCTL Output Pin	Comments
0 note[1]	0	0 (cleared)	0 cleared	Z note[2]	Output is tri-stated
1	0	0	0	0	SCSI term power source disabled
1	0	0	1	1	SCSI term power source enabled
1	1 note[3]	0 (no effect)	0 (cleared)	0	SCSI term power source disabled
1	0	1	0	1	SCSI term power source disabled
1	0	1	1	0	SCSI term power source enabled
1	1 note[3]	1 (no effect)	0 (cleared)	1	SCSI term power source disabled

[1] When RST# is asserted (low-level), then both bits (STPWLEVEL and STPWEN) are forced to be inactive and STPWCTL output pin is tri-stated.

[2] The STPWCTL output pin will remain tri-stated until the STPWEN bit is first written to the active state. Then output pin STPWCTL will remain driven at the selected level controlled by STPWEN until the next RST# assertion.

[3] When the software driver writes a one to the CHIPRST bit in HCNTRL register, the STPWEN bit is forced to be inactive. Writing to CHIPRST has no effect on the state of STPWLEVEL. Thus, CHIPRST forces STPWCTL to the selected inactive level.

Initializing the SCSI Section

When the device is reset at power on, the sequencer (SCSI PhaseEngine) is held in the paused state. This allows the driver to access the SCSI registers directly. The SCSI channel A may be initialized by writing the OID. All interrupts are disabled, and all automatic functions are turned off. The Bus Free status bit will be set after reset if the SCSI bus is in the Bus Free state.

Manual Mode Data Transfer

In Manual PIO mode, the SCSI block is used essentially as a bus buffer having no control functions. The host transfers data directly to and from the SCSI bus via the SCSI data latch registers SCSIDATL and SCSIDATH, and processes the SCSI control signals via the SCSI signal registers SCSISIGI and SCSISIGO. This mode only supports asynchronous transfers and is usually used during the Message and Status phases. Care should be taken to ensure that data is stable while ACK or REQ is asserted.

Automatic Mode Data Transfer

Automatic PIO transfers on the SCSI bus are enabled by setting SPIOEN (bit 3, SXFRCTL0). In Automatic PIO mode, the sequencer transfers data directly to and from the SCSI bus via the SCSI data latch registers SCSIDATL and SCSIDATH, while the hardware performs SCSI bus control automatically. Transfer complete can be signaled by an interrupt or by polling the status bit SPIORDY. This mode only supports asynchronous transfers and is usually used during the Message and Status phases. SCSI data may be read directly using SCSIBUSL and SCSIBUSH. The following Initiator and Target discussions assume an 8-bit transfer.

In Initiator mode, when the SCSI I/O signal indicates the Out direction with REQ active, SPIORDY (bit 1, SSTAT0) is a one and data may be written to SCSIDATL. Writing the data to SCSIDATL clears SPIORDY, the written data is presented on the SCSI bus, then ACK is driven active. REQ will be driven inactive by the Target, which will clear the ACK. When the SCSI I/O signal indicates the In direction and REQ is active, then valid data has been latched in the SCSIDATL register and SPIORDY (bit 1, SSTAT0) will be a one. When SCSIDATL is read, ACK is driven active on the SCSI bus and SPIORDY is cleared. Automatic mode may be left on during normal transfers without adverse effect. This allows handshake of Message In bytes with no additional bit manipulation.

In Target mode, when SCSI I/O indicates an Out direction and data is read from SCSIDATL, REQ will be driven. The Initiator will drive data onto the SCSI data lines and drive ACK active. The data will be latched on the leading edge of ACK in SCSIDATL and SPIORDY (bit 1, SSTAT0) will be set. Reading this byte with SPIOEN set will cause another REQ to be driven on the bus and will clear SPIORDY. When SCSI I/O indicates the In direction and data is written to SCSIDATL, the data is driven onto the SCSI bus, REQ is driven active and SPIORDY is cleared. When the Initiator reads the data it will drive ACK active. This will cause REQ to go inactive and will set SPIORDY.

Automatic PIO mode may be used with 16-bit SCSI data transfers, provided asynchronous timing is used. In this case, SCSIDATH should be written to or read from first, since the SCSI handshake signals will be triggered with an access to SCSIDATL.

Normal (DMA) Mode Data Transfer

In Normal (DMA) mode, the hardware performs the SCSI transfers and bus control automatically. Data is transferred automatically between the SCSI bus and the data FIFO through the SCSI FIFO. This transfer can be monitored via interrupts or by polling status bits. Wide, DMA transfers which are of the odd length and/or odd boundary type are handled automatically. Normal mode supports asynchronous transfers for Command and Data phase, and synchronous transfers which may be Wide and which are used during Data phase only. A DMA data transfer is enabled by setting up the SCSI and host sections with regard to direction, pointers and count values, then setting the appropriate enable bits in DFCNTRL. The data transfer rate for the Data phase is set up in the SCSIRATE register. This register chooses asynchronous or synchronous transfers, and may be set up beforehand. It has no effect on the Command phase.

Differential

Differential drivers may be added externally to the device transparently to software. The differential interface consists of four data bits (DIFFDAT), one strobe (DIFFSTRB) and two address bits (DIFFADR). The hardware external to the device is assumed to have the following address map and bit definition.

Table 5-3. Address Definition

Data bits	Adr=0	Adr=1	Adr=2
DIFFDAT3	SCSI ID 3	ENRST	
DIFFDAT2	SCSI ID 2	ENARB	ENDRV
DIFFDAT1	SCSI ID 1	ENSEL	ENTARG
DIFFDAT0	SCSI ID 0	ENBSY	ENINIT

Table 5-4. Bit Definition

Data Bits	Definition
SCSI ID 0-3	Own ID from SCSIID register.
ENRST	Enables SCSI Reset onto the SCSI bus.
ENARB	Enables OID onto SCSI data bus.
ENSEL	Enables SEL, data drivers, and ATN onto SCSI bus.
ENBSY	Enables BSY onto SCSI bus.
ENDRV	Enables data drivers onto SCSI bus.
ENTARG	Enables C/D, I/O, MSG, REQ, BSY onto SCSI bus.
ENINIT	Enables ACK, ATN onto SCSI bus.

Address 0 is written when OID (bits 0-3) in the SCSIID register is written to. The SCSI ID is latched externally and used during Arbitration. Address 1 is written at the appropriate time with the appropriate values to enable BSY, SEL, or the SCSI ID for Arbitration and Selection. Address 2 is written after a successful select or reselect and identifies the device as an Initiator or a Target and enables the drivers for the appropriate direction.

After the detection of a Bus Free condition, ENBSY and ENARB are set by the device. ENBSY drives the BSY line onto the SCSI bus. ENARB drives only the SCSI ID bit onto the SCSI bus. The device will monitor the SCSI data lines and if it is determined that Arbitration is won, will set the ENSEL bit along with ENARB cleared and ENBSY set. The external logic will turn on the data drivers to allow the single-ended device and target IDs to be driven, as well as the ATN driver. After the appropriate time, ENBSY will be cleared with ENSEL set. When the Target responds with BSY, then the selection is complete. The device will set ENDRV, and ENINIT, and will clear ENSEL and ENBSY in that order.

Differential Schematics

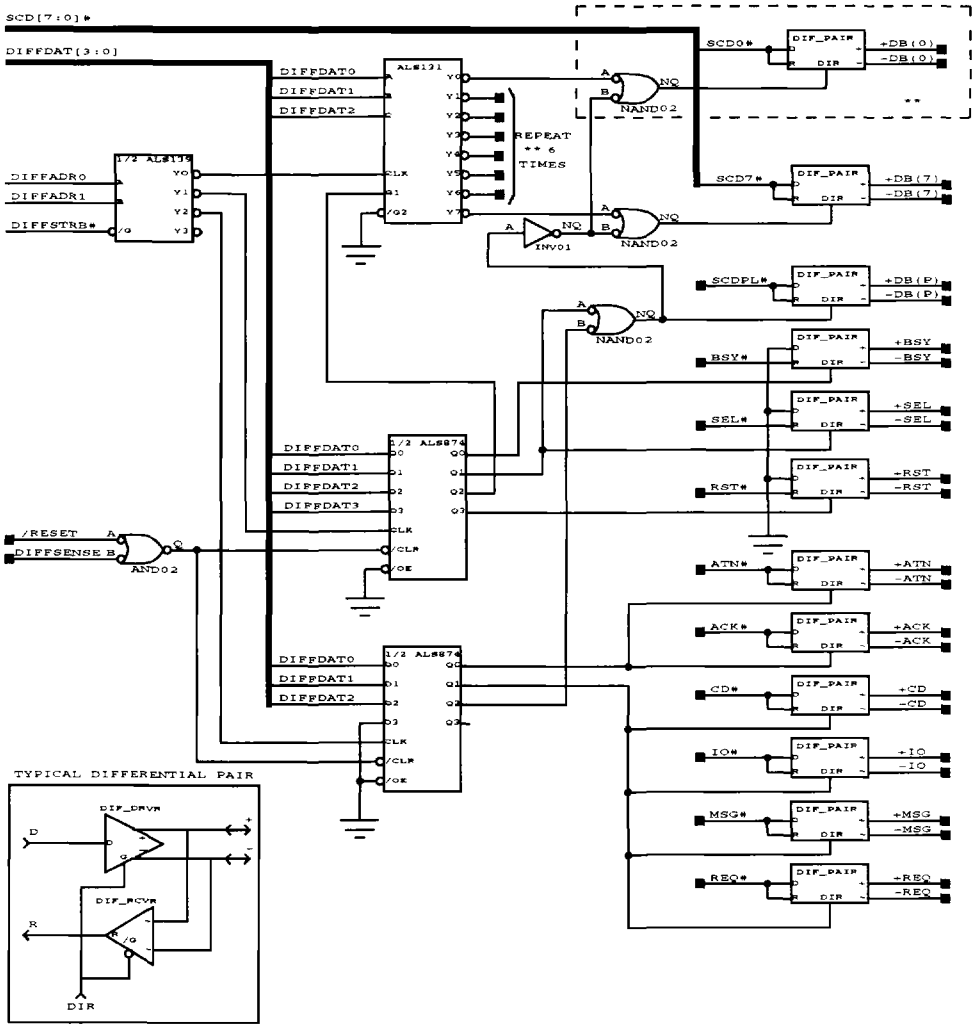


Figure 5-1. Differential Schematic

SCSI Interrupts

The SCSI module has one interrupt signal, SCSIINT, which is always routed to the active channel. The SCSI interrupt logic provides for the masking, generation, and clearing of all interrupts. This logic includes the interrupt mask (SIMODE), interrupt clear (CLRSINT) and interrupt status (SSTAT) registers. A SCSI interrupt is caused by some significant event occurring such as Selection/Reselection Successful, SCSI Reset, Transfer Done, Unexpected Bus Free, or Selection Time-out. SCSIINT is generated only when an interrupt condition occurs and the bit associated with the condition is set in the mask register SIMODE0 or SIMODE1. When an interrupt is generated, the status registers SSTAT0 and SSTAT1 will contain the cause of the interrupt. SCSIINT is cleared by writing to the associated bit in the appropriate clear register CLRSINT0 or CLRSINT1, or by the condition that caused the interrupt going away. Upon receiving an interrupt, the host may want to examine all bits in the status registers since the occurrence of another interrupt causing event before the host services the original interrupt will cause another bit to be set in the status register but will not cause another SCSIINT.

Counters

SHADDR(0-3) and STCNT(0-2) are the two counters used in Normal (DMA) mode and Automatic PIO transfers to regulate the flow of data and provide status information regarding the current transfer in progress.

The address pointer which is contained in counter SHADDR0-3 is loaded when the host address pointer LHADDR0-3 (088-08Bh) is loaded. The current value of the address pointer can be read by reading SHADDR0-3. This counter will contain the next starting address when a disconnect occurs. In DMA mode, both SDMAEN and SCSIEN must be disabled prior to reading or loading these counters. The counters are enabled when either SDMAEN, SPIOEN or SCSIEN is set and there is either a nonzero transfer count or a zero transfer count with SWRAPEN set. The event that both counters rely on is a SCSI byte being successfully transferred. A SCSI byte is considered transferred when writing to the SCSI bus when the handshake associated with this byte has occurred on the SCSI Bus (REQ/ACK). A SCSI byte is considered transferred when reading from the SCSI bus when the byte has been written to the data FIFO. Two counters are maintained for this purpose, and the sense of DIRECTION (bit 2, DFCNTRL) determines which one is used. This definition applies also to Wide transfers.

STCNT0-3 is the 24-bit counter that contains the DMA SCSI transfer count, which is the number of bytes remaining to be transferred. STCNT decremented by SCSI FIFO reads in the same manner that SHADDR is incremented. This counter can be made to wrap past 0 by setting SWRAPEN in SXFRCTL1. If SWRAPEN is 0 and STCNT counts from 1 to 0, transfers will stop and SDONE will be set.

SCSI FIFO

The SCSI FIFO exists to catch data during the Synchronous Data In phase as an Initiator. The FIFO may be reset under several conditions listed below after which no data will be in the FIFO.

- CHIPRST
- Setting CLRCHN (bit 1, SXFRCTL0)
- Clearing SDMAEN and DIRECTION=1 and STCNT=0 and Initiator mode

SCSI Reset

The SCSI bus may be reset by setting SCSIRSTO (bit 0, SCSISEQ), waiting for the reset time and then clearing SCSIRSTO. SCSIRSTI (bit 5, SSTAT1) will be set only when receiving a reset from some other device on the SCSI bus. If the reset originates from SCSIRSTO only, then SCSIRSTI will not be set.

SHADDR Address Pointer

The address pointer SHADDR(3:0) is loaded when the host low address pointer LHADDR(3:0) is loaded. LHADDR counts the number of bytes transferred on the Host bus and SHADDR counts the number of bytes that have been transferred on the SCSI bus. SHADDR will contain the next starting address when a disconnect occurs.

Sequencer

Loading

The sequencer is ready for loading after being reset or paused. The sequencer is loaded by first setting the LOADRAM bit in SEQCTL. The starting sequencer address should then be loaded in SEQADDR, with the low order address written first. The sequencer map should then be loaded sequentially into SEQRAM. The bytes are loaded into the RAM starting with the least significant byte at the address in SEQADDR. Subsequent bytes will load in the same word until the word is complete, and then SEQADDR is incremented and the next word is loaded. Parity should be disabled when loaded.

Pause

The sequencer may be paused at anytime without adverse effect by setting PAUSE in HCNTRL, by Automatic Access Pause (AAP) action from a host request without setting the PAUSE bit, or by the sequencer entering the SLEEP mode. When the access that caused the AAP is complete, or the status that has been enabled for sleep wake up is active, the sequencer will continue the same as when the PAUSE bit is reset. The sequencer will hold at the current address and all internal address and data paths will be gated to the host interface. The sequencer logic will set PAUSEACK in HCNTRL when the hardware is in this state. This state is used by the driver to gain access to any of the internal registers or RAM. When the driver is finished, the PAUSE bit is cleared and the sequencer will continue with its program. When PAUSE is cleared, the sequencer will always execute at least one instruction, even if some other event is active to pause the sequencer. When changing the address of the sequencer to start execution at a different location, SEQADDR0 should be written first, followed by SEQADDR1.

> **Note:** Sequencer reaction to a pause request when the current instruction is to SCB address will be delayed when RAMPSM is asserted and arbitration is required to gain access to the external SRAM as the sequencer instruction time is being stretched until the arbitration is completed.

Pause must be set before setting POWRDN (bit 6, HCNTRL).

Breakpoint

The sequencer has a diagnostic feature which allows a driver to stop the sequencer at a predetermined address. The address is loaded in BRKADDR0 and BRKADDR1 with BRKDIS (bit 7, BRKADDR1) cleared. When the program counter of the sequencer equals the value loaded in BRKADDR then the sequencer will be paused, and BRKADRINT (bit 3, INTSTAT) will be set. If BRKADRINTEN (bit 3, SEQCTL) is set, the IRQA# pin will also be driven active. BRKADRINT and the interrupt may be cleared by setting CLRBRKADRINT (bit 3, CLRINT). A driver may do any of the following:

- The driver may start execution from the current address and break on the next occurrence by clearing PAUSE.
- The driver may change the break address and clear PAUSE. This will start execution from the current address and break on the new one.
- The driver may single-step the sequencer.
- The driver may change the break address and the program counter, and clear PAUSE. The sequencer will start at a new address and break on a new address.

Single Step

The sequencer may be single-stepped after PAUSE (bit 2, HCNTRL) is set or a breakpoint has been reached. This is done by setting STEP in SEQCTL. The software driver should then clear PAUSE. The sequencer will execute one cycle and set PAUSE again. For consecutive single steps, PAUSE should be cleared consecutive times. To continue executing from the current location, clear STEP and then clear PAUSE.

Reset

The sequencer may be reset by writing to SEQRESET in SEQCTL. Setting this bit will cause the sequencer to start executing at address zero.

Restart

The sequencer may be restarted at any location by first setting PAUSE (bit 2, HCNTRL) and then loading SEQADDRL and SEQADDRH with the starting address. When the sequencer is unpaused by clearing PAUSE, the sequencer will start executing at the address that was loaded.

Indirect Jump

The sequencer may jump indirect to any location within the same 256 instruction page by writing the new address to SEQADDR0. The new address is moved from some general RAM location. A bank switch may be performed by setting SEQADDR1.

Hardware Failure Detect

The device has hardware failure detection mechanisms. MPARERR, DPARERR and sequencer RAM parity errors will be detected and causes a BRKADRINT interrupt which pauses the sequencer and drives the IRQA# pin when enabled. The cause of the interrupt may be read from the ERROR register. If this condition occurs, BRKADRINT may be cleared by setting CLRPARERR or CHIPRST (bit 0, HCNTRL). This feature may be disabled by setting FAILDIS (bit 5, SEQCTL).

Writing Hardware Control Bits

Due to the single-cycle operation of the sequencer, it is possible to set a hardware control bit and read a status bit which is affected by the control bit with the next instruction and not see the effect of the control bit. In these cases, one or more NOPS should be executed between the control write and the status read of the bit in interest. Table 5-5 is a list of status bits that fall into this category and the number of NOPs required.

Table 5-5. Writing Hardware Control Bits

Status Bit	Control Causing Change
SDONE	Goes active after loading a nonzero value into STCNT, or changing SWRAPEN.
DMADONE	Goes active after loading a nonzero into STCNT, or changing SWRAPEN, or after loading a nonzero value into HCNT.
SCSIPERR	Goes active after setting ENSPCHK (bit 5, SXFRCTL1). Goes inactive after setting CLRSCSIPERR (bit 2, CLRSINT1).
SCSIENACK	Goes active after setting SCSIEN (bit 5, DFCNTRL).
SDMAENACK	Goes active after setting SDMAEN (bit 4, DFCNTRL).
HDMAENACK	Goes active after setting HDMAEN (bit 3, DFCNTRL).
DIRECTIONACK	Goes active or inactive after setting or resetting DIRECTION (bit 2 DFCNTRL), if SCSIEN, SDMAEN, and HDMAEN are inactive.
FIFOFLUSHACK	Goes active after setting (bit 1, DFCNTRL), unless or until FIFOQWDEMP is active.
DWORDEMP	Goes active after setting FIFORESET, or reading DFDAT and less than 8 bytes remain in the data FIFO.
MREQPEND	Goes active when HDMAEN is active (with DIRECTION active and FIFOQWDEMP goes active), (with DIRECTION inactive and FIFOFULL or FIFOFLUSH goes active), or (CACHETHEN inactive and DFTHRSH goes active), or (CACHETHEN active and DFCACHETH goes active). Goes inactive when HDMAEN is active (with DIRECTION active and FIFOFULL or DFSDH goes active), (with DIRECTION inactive and FIFOQWDEMP or goes active), or HCNT value is zero, FIFORESET is written =1.
HDONE	Goes active when HCNT value is zero and final master data transfer is completed. Goes inactive after loading a nonzero value in HCNT.
DFTHRSH	Goes active or inactive with a write to or a read from DFDAT that causes the data FIFO stores data level to be at the selected DFTHRSH level.
FIFOFULL	Goes active when a write to DFDAT causes the data level stored in the data FIFO to be at maximum capacity (256 bytes), or writing to FIFORESET=1.
FIFOEMP	Goes active when a read from DFDAT causes the data level stored in the data FIFO to be at minimum capacity. (0 bytes) or writing to FIFORESET=1.
DFCACHETH	Goes active when DIRECTION is active and writing to FIFORESET=1, or when DFDAT is read so that space is available to store data to match or exceed the selected cache line size, or when DIRECTION is inactive and DFDAT is written to so that available stored data in the data FIFO match or exceed the selected cache line size. Goes inactive when DIRECTION is active and data is written to DFDAT so that space available is less than the selected cache line size, or when DIRECTION is inactive and DFDAT is read so that available stored data is less than the selected cache line size.

Host PCI Interface

Configuration

The configuration VENDERID will be hard-wired registers with a value of (9004h) representing ADP for Adaptec. The configuration DEVICEID will be hard-wired register with a value of 7870 following RST# assertion, however the value may be changed, see IDDAT. All chip setup will take place by the controlling BIOS or driver at initialization time.

Data Transfer

Data transfer is enabled by setting up the SCSI and host sections with regard to direction, pointers and count values. The data FIFO should be cleared, and then the HDMAEN, SDMAEN, and SCSIEN bits in DFCNTRL should be set to one. Transfers may be disabled by clearing any of these bits, but they should be polled for zero before the transfers are guaranteed to have stopped. In addition to these bits, HDONE and SDONE have been implemented to indicate the end of the transfer. DMADONE is also implemented and is the logical AND of HDONE and SDONE. DMADONE is intended to be one bit which will determine the end of transfer in either direction.

PCI Bus Commands

PCI bus commands indicate to the target the type of transaction the master is requesting. Bus commands are encoded on signals CBE[3:0]# during the Address phase of the transaction. The sense of values present on CBE[3:0]# for Bus commands during the Address phase is the same as that used on AD[31:00] (i.e., one = high level and zero = low level). This is the reverse of when CBE[3:0]# are used for valid byte indicators (=0) during Data phases the AIC-7870 support of PCI Bus commands follows:

Interrupt Acknowledge Command (IAC) CBE[3:0]#=0000: is a Read command implicitly addressed to the system Interrupt controller. The command is defined only by the CBE[3:0]# value. The ADn value during Address phase value is not used and the CBE[3:0] value in the Data phase determines the requested valid data width response expected on ADn.

- **The AIC-7870 as Target:** ignored after checking the address parity.
- **The AIC-7870 as Master:** not generated.

Special Cycle Command (SCC) CBE[3:0]#=0001: is a Message Broadcast command to pass status to all PCI agents on the PCI bus or for logical side-band signaling between PCI agents on the bus that recognize the passed message. The SCC contains no explicit destination address. Each agent on the PCI bus that accepts an SCC must determine whether the message is applicable to it. DEVSEL# and TRDY# are never asserted to an SCC. Command timing is controlled by FRAME# and IRDY# during the Address phase ADn are stable, with don't care value, with correct parity. During the single Data phase AD[07:00] encode 64 specified *fixed* messages, AD[23:08] are 128 reserved message values (reserved value is 00h and *must not be aliased or used*) and AD[31:24] encode 64 optional *soft* messages that are agent dependent (=00h when not used).

The current specified messages are:

AD[07:00]	Message
00h	Shutdown
01h	Halt
02h	RSVD
03h - 3Fh	RSVD

- **The AIC-7870 as Target:** ignored after checking the address parity.
- **The AIC-7870 as Master:** not generated.

IO Read Command (IORDC) CBE[3:0]#=0010: is a command to read data from an addressed Target's Device register space, which has been mapped into system IO Address space and enabled for access with ISPACEEN active.

- **The AIC-7870 as Target:** supports IORDC only for 8-bit transfers for all registers in its Device register space. Note, Disconnect will be returned when a data burst is indicated for all registers. When more than one CBE[3:0]# is asserted, a target-abort condition will be returned. When no CBE[3:0]# signal is asserted, the data cycle will be treated as a NOP. DEVSEL# is asserted using medium speed target response timing. TRDY# will not be asserted until the addressed register access and data is valid. The period before TRDY# is asserted will vary depending on the address (internal/external) and state of PAUSEACK and/or EXTARBACK#.
- **The AIC-7870 as Master:** not generated.

IO Write Command (IOWRC) CBE[3:0]#=0011: is a command to write data to a Device register space, which has been mapped into system IO address space and enabled for access with ISPACEEN active.

- **The AIC-7870 as Target:** supports IOWRC only for 8-bit transfers for all registers (except for SCB Array which may be either 8-bit or 32-bit) in its Device register space. Note, Disconnect will be returned when a data burst is indicated for all registers (except for linear burst order, 32-bit transfers to the internal SCB Array). When more than one CBE[3:0]# is asserted (except for the SCB Array where they must be one or all), a target-abort condition will be returned. When no CBE[3:0]# signal is asserted the data cycle will be treated as a NOP. DEVSEL# is asserted using medium speed target response timing. TRDY# will not be asserted until the addressed register access is valid. The period before TRDY# is asserted and will vary depending on the address (internal/external) and state of PAUSEACK and/or EXTARBACK#.
- **The AIC-7870 as Master:** not generated.

RSVD CBE[3:0]#=0100

- **The AIC-7870 as Target:** ignored after checking the address parity.
- **The AIC-7870 as Master:** not generated.

RSVD CBE[3:0]#=0101

- **The AIC-7870 as Target:** ignored after checking the address parity.
- **The AIC-7870 as Master:** not generated.

Memory Read Command (MRDC) CBE[3:0]#=0110: is a command used to read data from an addressed target mapped in the system Memory Address space with its MSPACEEN active in the Configuration Command register. MRDC is not cache line referenced and may contain any length of Data phases. MRDC may be used for transfers not starting on cache line boundaries to reach a boundary after which the MRDC command could be continued or either a MRDLC or MRDMC could be used to improve system memory performance.

- **The AIC-7870 as Target:** supports MRDC only for 8-bit transfers for all registers in its Device register space. The AIC-7870 supports 32-bit MRDC transfers only from the external ROM/EEPROM. Note, Disconnect will be returned when a data burst is indicated for all registers. When more than one CBE[3:0]# signal is asserted (except for ROM/EEPROM read where any value is acceptable), a target-abort condition will be returned. When no CBE[3:0]# signal is asserted, the data cycle will be treated as a NOP. DEVSEL# is asserted using medium speed target response timing. TRDY# will not be asserted until the addressed register access and data is valid. The period before TRDY# is asserted will vary depending on the address (internal/external) and state of PAUSEACK and/or EXTARBACK#.
- **The AIC-7870 as Master:** supports MRDC for transfers from system memory. It will be issued whenever the starting address is not on the selected cache line boundary, when DFCACHETH is not active, a byte offset condition exists, or remaining count is less than the cache line size. Also see MRDCEN.

Memory Write Command (MWRC) CBE[3:0]#=0111: is a command used to write data to a target mapped in the system Memory Address space with its MSPACEEN active in the Configuration Command register. MWRC is not cache line referenced and may contain any length of Data phases. MWRC must be used for transfers not starting on cache line boundaries to reach a boundary after which the MWRC command could be continued or a MWRIC could be used to improve system memory performance.

- **The AIC-7870 as Target:** supports MWRC only for 8-bit transfers for all registers (except for SCB Array which may be 8-bit or 32-bit, see SCBAUTO) in its Device register and external ROM/EEPROM spaces. Note, Disconnect will be returned when a data burst is indicated for all registers (except for linear burst order, 32-bit transfers to the internal SCB Array). When more than one CBE[3:0]# is asserted (except for the SCB Array where they must be one or all), a target-abort condition will be returned. When no CBE[3:0]# signal is asserted, the data cycle will be treated as a NOP. DEVSEL# is asserted using medium speed target response timing. TRDY# will not be asserted until the addressed register access is valid. The period before TRDY# is asserted and will vary depending on the address (internal/external) and state of PAUSEACK and/or EXTARBACK#.
- **The AIC-7870 as Master:** supports MWRC for transfers to system memory. It will be issued whenever the starting address is not on the selected cache line boundary, when DFCACHETH is not active, a byte offset condition exists, remaining count is less than the cache line size, or FIFOFLUSH is active.

RSVD CBE[3:0]#=1000

- **The AIC-7870 as Target:** is ignored after checking the address parity.
- **The AIC-7870 as Master:** not generated.

RSVD CBE[3:0]#=1001

- **The AIC-7870 as Target:** ignored after checking the address parity.
- **The AIC-7870 as Master:** not generated.

Configuration Read Command (CRDC) CBE[3:0]#=1010: is a command used to read data from a device's Configuration space or spaces mapped into the system Memory Address space by connecting one of AD[31:11] address lines to the device IDSEL input. All devices are required to support this command.

- **The AIC-7870 as Target:** supports CRDC access for all registers in its single function Configuration register space. Note, Disconnect will be returned when a data burst is indicated for all registers. All 32-bits are always provided without regard for the CBE[3:0]# value. When no CBE[3:0]# signal is asserted, the data cycle will be treated as a NOP. DEVSEL# is asserted using medium speed target response timing. For valid accesses TRDY# is asserted one PCLK following DEVSEL# assertion.
- **The AIC-7870 as Master:** not generated.

Configuration Write Command (CWRC) CBE[3:0]#=1011: is a command used to write data to a device's Configuration space or spaces mapped into the system Memory Address space by connecting one of AD[31:11] address lines to the device IDSEL input. All devices are required to support this command.

- **The AIC-7870 as Target:**supports CWRC for all registers in its single function Configuration register space. Note, Disconnect will be returned when a data burst is indicated for all registers. Any combination of CBE[3:0]# value assertion is acceptable for writing bytes and when none is asserted, the data cycle will be treated as a NOP. DEVSEL# is asserted using medium speed target response timing. For valid accesses TRDY# is asserted one PCLK following DEVSEL# assertion.
- **The AIC-7870 as Master:** not generated.

Memory Read Multiple (MRDMC) CBE[3:0]#=1100: is a command used to read data from a target mapped in the system Memory Address space with its MSPACEEN active in the Configuration COMMAND register. MRDMC is cache line referenced and when used indicates that multiple cache lines are expected to be required for the transaction, but do not have to be used.

- **The AIC-7870 as Target:** defaults to MRDC.
- **The AIC-7870 as Master:** supports MRDMC for transfers from system memory. It will be issued whenever the starting address is on the selected cache line boundary, no byte offset condition exists, remaining count is greater than/or equal to the cache line size, and DFCACHETH is active, with CACHETHEN not active. Cache line streaming is supported.

Dual Address Cycle (DAC) CBE[3:0]#=1101: is a command used to transfer 32-bit data anywhere in a 32-bit address space segment of a 64-bit address space. The 64-bit address is indicated on AD[31:00] with two address phases of one PCLK each. The low 32- bits are transferred in the first PCLK and the high 32-bits in the second PCLK with the data on the following PCLKs. The CBE[3:0]# value supplied in the second address PCLK period identifies the type of data command the same as when only a single address cycle occurs.

- **The AIC-7870 as Target:** is ignored after checking address parity on both address phases.
- **The AIC-7870 as Master:** supports DAC for transfers to and from system memory. It will be issued in the first address phase whenever DACEN is active in the Configuration DEVCONFIG register and HHADDR[3:0] register contents are not zero. In the second address phase, the AIC-7870 will issue a MRDC, MRDLC, MRDMC, MWRC or MWRIC command followed by Data phases of the transaction. Transfers across 4 GByte boundaries are not allowed.

Memory Read Line (MRDLC) CBE[3:0]#=1110: is a command used to read data from a target mapped in the system memory address space with MSPACEEN active in the Configuration Command register. MRDLC is cache line referenced and when used indicates that a single cache line is expected to be required for the transaction, but does not have to be used.

- **The AIC-7870 as Target:** defaults to MRDC.
- **The AIC-7870 as Master:** supports MRDLC for transfers from system memory. It will be issued whenever the starting address is on the selected cache line boundary, no byte offset condition exists, remaining count is greater than/or equal to the cache line size, and DFCACHETH and CACHETHEN are active (note DFTHRSH is not used). Control of data transfer size to only cache line size may be obtained when the CACHSIZE, LATTIME registers and the GNT# assertion time have appropriate values, except for the last transfer which may be less. Also see MRDCEN. Cache line streaming is supported.

Memory Write and Invalidate (MWRIC) CBE[3:0]#=1111: is a command used to write data to a target mapped in the system Memory Address space with MSPACEEN active in the Configuration Command register. MWRIC is cache line size referenced and when used to improve system memory performance indicates that complete cache lines are to be transferred in the transaction.

- **The AIC-7870 as Target:** defaults to MWRC.
- **The AIC-7870 as Master:** supports MWRIC for transfers to and from system memory. It will be issued whenever the starting address is on the selected cache line boundary, no byte offset condition exists, remaining count is greater than/ or equal to the cache line size, and DFCACHETH and DFTHRSH (DFTHRSH is not used when CACHETHEN is active) are active. When CACHETHEN is active, control of data transfers to only cache line size when the CACHESIZE, LATTIME registers and GNT# assertion time have appropriate values. When FIFOFLUSH is active, a MWRC command will be used instead of MWRIC. Should the Target signal Disconnect in the middle of a MWRIC cache line command, the AIC-7870 will terminate the MWRIC command and release the bus. The AIC-7870 will then request the bus to complete the cache line using MWRC. Cache line streaming is supported.

Memory Port Interface

The AIC-7870 has an external Memory Port interface, which optionally may be used to extend its internal capabilities and control external (to the AIC-7870) functionality. In addition, the AIC-7870 may share these external resources with other AIC-7870s or devices that are compatible with the AIC-7870's Memory Port protocol. When sharing is to be performed, an external Arbitration unit is required.

SRAM may be either an 8-bit or 9-bit device. When an 8-bit device is used, the MDP input must be pulled up to prevent float conditions on read cycles and the EXTSCBPEN bit must be inactive to prevent false parity errors from being reported. A 9-bit device is used when SRAM parity operation is desired. Notice that in this case, the EXTSCBPEN bit must be active to allow checking of SRAM parity. SRAM parity will always be generated on writes even if not used. SRAM access cycle time will be doubled when EXTSCBTIME bit is active. The maximum SRAM size for direct addressability is 8 KBytes. The size of the SRAM present will be determined by the controlling driver software. SRAM paging may be performed by use of the BRDCTL register function and external board logic.

When the host writes to the external SRAM, it only performs a write-only cycle that follows the PCI write access timing. MRW will be asserted low due to a CBE[3:0]# decoded write command, then RAMCS# will be asserted (low pulse, TRDY# assertion), with the data stored on the rising edge.

When the sequencer writes to the external RAM, one of two cycles will be performed, a write-only cycle or a read-modify-write cycle with timing from the CLKIN pin. Detection of the sequencer cycle type is determined by comparing the sequencer read and write address buses to the SCB address range. When only the sequencer write address bus is within the SCB address range then it is a write-only cycle, and when both sequencer address buses are within the SCB address range it is a read-modify-write cycle. For the write-only cycle, MRW is asserted low after detection of the cycle type. Then, near the end of the sequencer instruction cycle, RAMCS# is asserted (low pulse) with the data stored on the rising edge. For the read-modify-write cycle, MRW is asserted high after detection of the cycle type and RAMCS# is asserted (pulsed low) to access the read data from the SRAM for the sequencer. After the RAMCS# read pulse is high, MRW is asserted low to start the write portion of the cycle. Then, near the end of the sequencer instruction cycle, RAMCS# is again asserted (low) for a write pulse, with the modified data stored on the rising edge.

When the host reads from the external SRAM it only performs a read-only cycle that follows the PCI read access timing. MRW will be asserted high due to a CBE[3:0]# decoded Read command, then RAMCS# will be asserted (low pulse, so that the SRAM read data will become valid before asserting TRDY#).

When the sequencer reads from the external SRAM it only performs a read-only cycle, with timing from the CLKIN pin. Detection of the sequencer cycle type is determined by comparing the sequencer read and write address buses to the SCB address range. When only the sequencer read address bus is within the SCB address range, then it is a read-only cycle. For the read-only cycle, MRW is asserted high after detection of the cycle type, then RAMCS# is asserted (pulsed low) to access the read data from the SRAM for the sequencer.

When the host writes to the external 8-bit EEPROM, it performs a write-only cycle that follows the PCI write access timing with extended timing for an 150 nsec access device. MRW will be asserted low due to a CBE[3;0]# decoded Write command with EXROMEN

active then RAMCS# will be asserted (low pulse, with the data stored on the rising edge) after which TRDY# will be asserted to end the cycle.

When the host reads from the external 8-bit ROM/EEPROM (up to 64 KByte), four read-only cycles per host access will be performed with the PCI read access extended for the four read cycles from the 150 nsec access device. MRW will be asserted high due to a CBE[3;0]# decoded Read command with EXROMEN active then RAMCS# will be asserted (low pulse, with the read data stored internally on each rising edge to assemble 32-bits) after which TRDY# will be asserted to end the cycle.

When the host or sequencer accesses SEEPROM or board logic devices, the timing of MD[7:0] and MDP are controlled by the states of the bits in SEECTL and BRDCTL registers while SEEMS is active and EXTARBACK# is asserted.

The external Memory Port consists of the following signal lines:

Table 5-6. Memory Port Interface

15	Address Out (MA[14:0])
8	Bidirectional Data (8-bit data) (MD[7:0])
1	Bidirectional Memory Data Parity (MDP/ BRDRW/MA15)
1	Arbitration Request (EXTARBREQ#)
1	Arbitration Acknowledge (EXTARBACK#)
1	RAM Chip Select (RAMCS#)
1	ROM Chip Select (ROMCS#)
1	Read /Write (MRW)
1	Serial EEPROM/Board Logic Select (SEECS)
1	External SRAM Present (RAMPS#)

Memory Port Operation

Not Used

When the Memory Port is not used (no external devices), the EXTARBACK# input must be grounded to enable the AIC-7870 to self-terminate its Memory Port inputs. Otherwise, individual pull-ups must be provided to MD[7:0] and MDP to prevent floating conditions and high power dissipation.

Stand-alone Operation

The AIC-7870 may be used in a stand-alone Memory Port mode by grounding EXTARBACK#. Stand-alone mode indicates that the AIC-7870 will be the only active user of the external resources (SRAM/ROM/EEPROM/SEEPROM/board logic devices). This mode may be verified by reading the state of MPORTMODE bit in the Configuration register. For stand-alone type operation, no external arbitrator is required.

Shared Operation

The AIC-7870 may be used in a shared Memory Port mode by adding an external Memory Port arbitrator. An EXTARBREQ# and EXTARBACK# pair from each device is connected to the arbitrator. The Arbitrator uses the same clock signal as connected to the

AIC-7870's CLKIN pin to sample the EXTARBREQ# signals and control the EXTARBACK# signals in order to facilitate orderly use of the resources. (For a description of the protocol, see signal pin list for EXTARBREQ# and EXTARBACK#.)

SEEPROM/Board Control Logic Operation

The external Memory Port may be used to control custom logic. The Board Control mode is controlled by two registers which are either controlled by the host (when pause is active) or the sequencer. To enable board control, the SEEMS bit in the SEECTL (1E) register must be set. Setting the SEEMS bit will assert EXTARBREQ#. When the EXTARBACK# is asserted (indicated by SEERDY, bit 7 SEECTL register), the board control operations may start. Actual board control is now passed to the SEECTL and BRDCTL registers. The external bus is owned as long as the SEEMS bit remains active. Care must be taken not to lock out any other devices (shared operation) for extended periods of time.

BRDCTL to external memory data port signal correlation is shown in Table 5-7.

Table 5-7. BRDCTL Signal Correlation

BRDCTL Register	Register Bit Name	MD Pin Name	Function
7	BRDDAT7	MD7	I/O
6	BRDDAT6	MD6	I/O
5	BRDDAT5	MD5	I/O
4	BRDSTB	MD4	O
3	BRDCS	MD3	O
2	BRDRW	MDP	O
1			
0			

* External pull-up resistors to be provided to prevent float conditions when pins are configured as inputs. Since this condition is register based it could exist for long periods of time.

The BRDRW signal determines if data is written or read on the MD[7:5] pins. BRDRW when active, indicates that the value read is from the external MD[7:5] lines (by reading the BRDCTL register). BRDRW when inactive, indicates that the value written to BRDDAT[7:5] will be asserted on the MD[7:0] pins and may be written into external logic by use of BRDSTB.

It should be noted that MD[2:0] are connected to the SEECTL register when the SEEMS bit is set. The correlation is shown in Table 5-8.

Table 5-8. SEECTL Signal Correlation

SEECTL Register	Register Bit Name	MD Pin Name	Function
2	SEECK	MD2	O
1	SEEDO	MD1	O
0	SEEDI	MD0[1]	I

[1] External pull-up resistor to be provided to prevent float conditions when pin is configured as an input. Since this condition is register based it could exist for long periods of time.

The following register bits control timing and status of the Memory Port functions.

Table 5-9. SEECTL Support Functions

SEECTL Register	Register Bit Name	Pin Name	Function
7	EXTARBACK#	EXTARBACK#	[1]
6	EXTARBREQ#	EXTARBREQ#	[1]
5	SEEMS	none	[2]
4	SEERDY	none	[3]

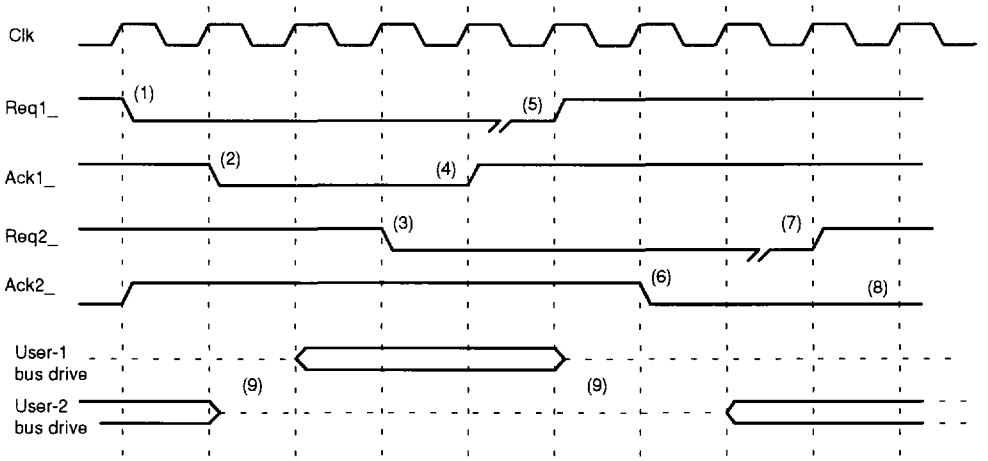
[1] Read access of arbitration control pins.
[2] SEEPROM/Board Control logic mode selection.
[3] Internal timing function for software usage and mode activation arbitration completed.

Memory Port Arbitration

Arbitration for the Memory Port is accomplished by external logic. The exact functionality is largely dependent upon the needs of the application. The following Memory Port bus Arbitration protocol must adhered to, see the Arbitration diagram. The protocol is characterized by the ability for a device to park itself on the bus if no other requests are outstanding. This feature reduces bus access time overhead. Another important feature is that successive bus grants will always be separated by one cycle, thus eliminating potential bus contention. The maximum number of EXTARBREQ#/ EXTARBACK# pair channels is a function of the arbiter design and bus loading. A maximum of 100pf per signal is allowed.

The Arbitration protocol is fully interlocked and synchronous (40 MHz clock). This implies that any arbiter design must use the same clock source as the AIC-7870 CLKIN pin. To minimize the effect of skew, only the positive edge of the clock should be used. Typically a device requesting the bus asserts (drives low) its EXTARBREQ# line. In response, the arbiter asserts (drives low) its corresponding EXTARBACK# line. The arbiter asserts one and only one EXTARBACK# signal at any time. The AIC-7870 may *park* itself on the bus (EXTREQLCK in DSCOMMAND register), by keeping its EXTARBREQ# active (even if no external cycles are run). The arbiter cannot grant the bus to any other devices, until a granted requester deasserts its EXTARBREQ# line. If another device requests the bus, the present owner of the bus is signalled to remove itself from the bus. This signalling is done by the arbiter by deasserting the EXTARBACK# signal to the current owner. The current owner either relinquishes the bus immediately, or finishes up the cycle(s) in progress by deasserting its EXTARBREQ# signal. The actual arbiter design may rely on customer requirements (round robin, priority etc.) as long as the Memory Port bus Arbitration protocol is maintained.

Arbitration Protocol



All timing on the positive edge of the clock.

(1) User-1 requests bus by asserting Req1_ .

(2) Arbitor grants bus to User-1 by asserting Ack1_ (no other Requests are active).

(3) User-2 requests bus by asserting Req2_.

(4) Arbitor recognizes Req2_, a new bus request, and removes Ack1_.

(5) User-1 recognizes Ack1_ has been de-asserted and completes the current access then de-asserts Req1_.

(6) Arbitor recognizes Req1_ has been de-asserted and asserts Ack2_.

(7) User-2 completes it's access and de-asserts Req2_.

(8) Arbitor has no active requests and continues to assert Ack2_ to maintain a bus parking condition for the last user.

(9) Bus exchange clock.

Figure 5-2. Arbitration Protocol

Reading and Writing the Data FIFO

The data FIFO may be read at any time or written when no other DMA activity is writing to the FIFO. Any attempt to enable two sources to write to the FIFO will result in a BRKADRINT interrupt. There are three sources which may read or write the FIFO; the SCSI Data Transfer port, the Host Data Transfer port, and the I/O port DFDAT. Read data is pointed to by DFRADDR0 and Write data is pointed to by DFWADDR0. These pointers point to 64-bit quad-words. Data is properly aligned in conjunction with the state of bits LHADDR(02:00). When set, FIFORESET (bit 0, DFCNTRL) clears the FIFO address counters (DFRADDR0, and DFWADDR0) and loads the byte offset pointers. The byte offset is decoded from LHADDR(02:00). Normal data transfer does not require any intervention from the sequencer. The correct offset is set up automatically when the Low Host Address is loaded in LHADDR and the FIFORESET bit is set. DFWADDR is incremented by writes to the FIFO from whatever source is active, and DFRADDR is incremented by reads from any source.

Data may be written to any location in the FIFO by first setting up LHADDR, then setting FIFORESET. The byte offset pointers will be pointing to the correct offset for the first byte transferred. DFWADDR may be changed to point to any starting location in the FIFO while maintaining the same byte offset. Consecutive writes will load consecutive

FIFO locations. Data may be sent to system memory by setting up LHADDR and HCNT, resetting the FIFO, writing the data to DFDAT, and setting DIRECTION with HDMAEN in DFCNTRL. When HDONE is set, the contents of the FIFO have been written to system memory.

Data may be read from any location in the FIFO by first setting up LHADDR, then setting FIFORESET. The byte offset pointers will be pointing to the correct offset for the first byte transferred. DFRADDR may be changed to point to any starting location in the FIFO while maintaining the same byte offset. Consecutive reads will return consecutive FIFO locations. Data may be read from system memory by setting up LHADDR and HCNT, resetting the FIFO, and clearing DIRECTION and setting HDMAEN in DFCNTRL. When HDONE is set, the contents of the FIFO have been read from system memory, and the data may be read from DFDAT.

Table 5-10 shows which port is active and in which direction with respect to the data FIFO.

Table 5-10. Data Flow Control

Direction	HDMAEN	SDMAEN	Read FIFO	Write FIFO
0	1	1	Host	SCSI
0	0	1	Sequencer	SCSI
0	1	0	Host	Sequencer
X	0	0	Sequencer	Sequencer
1	1	1	SCSI	Host
1	0	1	SCSI	Sequencer
1	1	0	Sequencer	Host

Data is aligned in the data FIFO and the SCSI FIFO according to the offset of the starting address from a quad-word boundary. The alignments in the data FIFO and SCSI FIFO are summarized in Table 5-11.

Table 5-11. Data FIFO and SCSI FIFO Alignments

LHADDR			PCI Bus	Host Block	Data FIFO Block Byte Offset			SCSI Block
(02)	(01)	(00)	Byte Offset	Byte Offset	SEQ Port	SCSI Port	HOST Port	Byte Offset
0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0	1
0	1	0	2	0	2	2	0	0
0	1	1	3	0	3	2	0	1
1	0	0	0	4	4	4	0	0
1	0	1	1	4	5	4	0	1
1	1	0	2	4	6	6	0	0
1	1	1	3	4	7	6	0	1

Writing to the SCB Array

The SCB Array may be written to by the driver after the sequencer is paused. The SCB Array is divided up into sections which are addressed by the value in SCBPTR. Only one 32-byte area of the array may be accessed at a time. Each area is mapped to the same address location no matter which area is selected. The device contains an Auto Increment feature where many or all the locations may be loaded with a minimum of overhead. The starting address is loaded in SCBCNT with SCBAUTO set. Each write to or read from any SCB Array address increments the address to the next location. This feature may be used with the REP OUTSB instruction or one of the memory MOV instructions to quickly load an SCB in the Array. The REP INSB instruction may be used to read the contents of an SCB. The contents of the SCB are application specific. SCBAUTO must be cleared to allow random access to the SCB Array. See *SCB Definition* on page 6-5 for a detailed description of the SCB contents.

Executing a Command

An operation is started by first pausing the sequencer by setting PAUSE in HCNTRL. This prevents the sequencer and controlling driver from colliding on the internal control bus. The driver should then wait for PAUSEACK (bit 2, HCNTRL) to become one. The driver should then save SCBVAL in SCBPTR so it may be restored later. The driver loads SCBPTR with the value of the empty SCB that it wishes to load. The SCB information is then loaded and the value that was written in SCBPTR is written to the QINFIFO. SCBPTR is then restored to the value that was previously set and the sequencer is then unpaused, and allowed to resume its program. The sequencer will initially be scanning the QINFIFO and if something is there, will read the pointer and attempt to execute that SCB if it does not conflict with an already open SCB. If it does conflict with an open command, then the SCB ID will be written back to the QINFIFO.

Once a command is started, the Target may disconnect. The sequencer will then save data pointers and mark a command as disconnected. The sequencer will then enter an idle loop and look for the next command to execute from the QINFIFO. If the sequencer needs driver assistance to execute a SCB, it will interrupt with the appropriate code in the INTSTAT with SEQINT set.

When the sequencer is finished with the command, it will write the SCB pointer value in the QOUTFIFO and will interrupt the driver with CMDCMPLT (bit 1, INTSTAT) set. The driver will then read the QOUTFIFO to get the value of the SCB that has just finished. If an error occurred, the driver should then save the SCBPTR value and load the SCB pointer of the finished SCB and read the SCB information. All status to report will be in the SCB area. The driver should then restore the SCB Pointer and clear PAUSE (bit 2, HCNTRL) to continue processing.

Interrupts

Interrupts fall into four basic classes: normal operation, driver intervention, error, and diagnostic. Interrupt status is given in INTSTAT. The sequencer does not have to be paused to read INTSTAT. The CMDCMPLT bit is set by the sequencer to indicate that a command has been completed and its location has been written to the QOUTFIFO. The sequencer will still be running and executing any other commands that have been loaded. Sequencer interrupts are interrupts that require the driver to intervene in the normal operation in order to provide a lengthy or difficult calculation. Sequencer interrupts are caused by the sequencer setting the SEQINT bit in INTSTAT along with the

INTCODE. Setting the SEQINT bit will cause the sequencer to self-pause. The sequencer may be restarted by clearing the SEQINT bit and writing a zero to the PAUSE bit in HCNTRL. The sequencer code will be structured to continue after the driver is finished handling the particular situation. A SCSI interrupt is caused by some catastrophic event such as a SCSI Reset, SCSI Parity Error, Unexpected Bus Free, or Selection Time-out. This interrupt is generated by hardware according to any SCSI event that is enabled in the SIMODE0 or SIMODE1. The sequencer is also paused by this interrupt. The BRKADRINT interrupt is used with special diagnostic code for the purpose of device debug, or for the detection of a hardware failure. The sequencer is paused by this
interrupt.

SCSI Interrupts

SCSI interrupts occur when the appropriate bit in SIMODE0 or SIMODE1 is set and the corresponding condition comes true. This will set the system interrupt pin if INTEN (bit 1, HCNTRL) is set and also the SCSIINT bit in INTSTAT. If the sequencer is executing a SCSI command, these conditions are error conditions and will pause the sequencer.

If the driver is executing the SCSI command, this is the normal way to respond to a SCSI interrupt.

Command Complete Interrupts

A Command Complete interrupt happens when the sequencer writes to the INTSTAT register with that bit set. It signifies that a command is finished and the ID has been loaded in the Queue Out register. The driver may read the Queue Out register and Queue Out count until the Queue Out is empty without pausing the sequencer. In this way the driver may service commands that have completed without error without interrupting the sequencer.

Breakpoint Interrupts

The sequencer has a diagnostic feature which allows a driver to stop the sequencer at a predetermined address. The address is loaded in BRKADDR0 and BRKADDR1 with BRKDIS (bit 7, BRKADDR1) cleared. When the program counter of the sequencer equals the value loaded in BRKADDR then the sequencer will be paused. BRKADRINT (bit 3, INTSTAT) will be set at this time. If BRKADRINTEN (bit 3, SEQCTL) is set, the IRQA# pin will also be driven active. BRKADRINT and the interrupt may be cleared by setting CLRBRKADRINT (bit 3, CLRINT).

This interrupt is also set upon detection of an illegal opcode, or sequencer RAM parity error. This feature may be disabled by setting FAILDIS (bit 5, SEQCTL).

Software Interrupt

The interrupt line IRQA# may be set by the software driver by setting SWINT (bit 4, HCNTRL). IRQA# will remain active until SWINT is cleared. INTEN will override SWINT, and must be set in order to see the IRQA#.

Interrupt Summary

Table 5-12. Interrupt Summary

Description	Enable Conditions	Pause	INSTAT bit	ERROR bit
Sequencer Parity Error	PERRORDIS=0 and parity error detected during opcode read	Yes	BRKADRINT	PARERR
Memory Parity Error	FAILDIS=0 and MPARERR detected	Yes	BRKADRINT	MPARERR
Data Parity Error	FAILDIS=0 and DPARERR detected	Yes	BRKADRINT	DPARERR
Opcode Error	FAILDIS=0 and ILLOPCODE detected	Yes	BRKADRINT	ILLOPCODE
PCI Status Error Detected	FAILDIS=0 and PCI status error detected	Yes	NONE	PCIERRSTAT
Sequencer Break Address Accessed	BRKDIS=0 and BRKADRINTEN=1 and BRKADDR compares with sequencer address	Yes	BRKADRINT	NONE
SCSI Event	Set in SIMODE0 and SIMODE1	Yes	SCSIINT	NONE
Sequencer Event	Always enabled	Yes	SEQINT	NONE
Command Complete	Always enabled	No	CMDCMPLT	NONE
Software	Always enabled	No	NONE	NONE

Power-down

Power may be conserved by degating the clocks to most of the chip. Setting POWRDN (bit 6, HCNTRL) will cause the chip to minimize the use of CLKIN and PCLK signals. Limit the Device space register access and disable any interrupts that may be generated independent from the clock. Interrupts pending in this case will drive IRQA# as soon as POWRDN is cleared. The sequencer must be paused before setting POWRDN.

Diagnostics

Upon power-on, the driver or BIOS will perform a series of diagnostics to the chip to ensure proper operation. This involves a series of register and RAM verifications as well as diagnostic code which will verify the sequencer and internal data path. Table 5-13 lists the series of tests.

Table 5-13. Diagnostics

Test Performed	Action Taken
Sequencer RAM Check by Driver	Driver verifies sequencer RAM through Host interface
Scratch and SCB RAM Check by Driver	Driver verifies scratch and SCB RAM through Host interface
Data FIFO Check by Driver	Driver verifies Host / FIFO interface
Register Check by Driver	Driver verifies Write/Read registers where possible
Sequencer Instructions	Diagnostic code is loaded to verify sequencer operation
Register Check by Sequencer	Sequencer verifies Write/Read registers where possible
Scratch and SCB RAM check by Sequencer	Sequencer verifies scratch and SCB RAM
Data FIFO Check by Sequencer	Sequencer verifies Write/Read ability
Data Path	Sequencer transfers data through data FIFO/SCSI interface
Interrupts	Driver verifies proper interrupt operation
Queue In/Queue Out	Driver verifies proper operation
Power Down Mode	Driver verifies power down nonoperation

6 Application Notes

About This Chapter

Read this chapter to find out

- The phases that occur on the SCSI bus
- The purpose of the data FIFO
- The instruction set for programming the sequencer (SCSI PhaseEngine)
- Command line definitions and formats

▼▼▼▼▼ 6

Chip Initialization

Certain hardware level features must be initialized before the device can be used. Following is a summary of those features. The actual value loaded in the registers depends on the application.

- PCI standard Configuration registers (COMMAND0, COMMAND1, CACHESIZE, LATTIME, BASEADR0, BASEADR1, EXROMCTL, INTLINSEL)
- PCI Device register (DEVCONFIG)
- Device space registers (DSCOMMAND, HCNTRL)
- Data FIFO Thresholds
- SCSI ID

SCSI Phases

Arbitration/Selection

Arbitration and Selection are automatic hardware sequences which are started by the sequencer (SCSI PhaseEngine). Arbitration will retry after a failed arbitration until complete. When a Bus Free condition is detected, the SCSI BSY signal is asserted along with the SCSI ID of the device. If no other higher priority IDs are on the SCSI bus and SEL is not active, the device will assert SEL with the device and target IDs, and drop BSY. After arbitration, the Selection phase will be entered using the target ID which is loaded in SCSIID. The attention bit will be driven during the Selection phase if ENAUTOATNO (bit 3, SCSISEQ) is set.

ID Message

The next usual event on the SCSI bus is the Message Out phase. If ATN is active, the ID message is sent by the device. The driver has control over the content of the message and may disable disconnection by setting a bit in the SCB. If the Attention signal is not driven during the Selection phase, then this phase is not entered by the Target.

The driver may also execute synchronous negotiation or wide negotiation with this Target after the ID message is sent by setting the appropriate bit in the synchronous or wide control byte in scratch RAM. The sequencer will interrupt with the proper interrupt status. The sequencer will also interrupt if it needs assistance to execute an Extended message. The driver will handle the Extended message and then release the sequencer to complete the command.

Command

After the Message Out phase or Selection phase, the Command phase is usually entered. The sequencer gets the command pointer from the SCB area which was loaded by the driver, and the number of bytes are sent that the Target asks for up to the limit in the command byte count in the SCB. The command is sent to the SCSI bus by using the Bus Master DMA mechanism.

Data Phase

The phase should be Data In or Data Out at this time. The driver will set up the data path for a write or read operation and the SCSI sequencer will send data when it arrives or receive data when it comes if the phase matches with the expected phase. The number of bytes is loaded into a 24-bit counter which is counted down during the Data phase. This counter is examined by the sequencer after the Data phase for correctness. SDONE in SSTAT2 will be set if the counter has a zero value. An interrupt will be generated by the sequencer to indicate an underrun or overrun condition. The sequencer will calculate the residual count on an underrun.

Note: The transfer speed and control signal widths are dependent on the input clock frequency.

Disconnection

The sequencer will handle all disconnections. The SCB contains the address pointer and byte counter for the particular transfer. If the Save Data Pointers message is received, the current value is saved in the SCB area. If the Disconnect message is received without the Save Data Pointers message, the value in the SCB area is not changed. The sequencer will mark the SCB as a disconnected command so it may be found at reconnect time.

Reconnection

Reselection should always be enabled when there is an outstanding command. When a Target reselects the device, the sequencer will get the LUN from the ID message and attempt to match the target ID, channel and LUN to a disconnected SCB. If one is found the tag enable bit in the control byte is checked, and if enabled, the tag value is received and the correct SCB is continued. The sequencer will then follow the target's phase and if the Data phase is entered the address pointer and byte counter will be loaded and the transfer continued from where it was left off. If a match is not found, then the host will be interrupted.

Modify Data Pointers

The AIC-7870 will support the Modify Data Pointers message if the Scatter/Gather list count is equal to one. The sequencer will accept the 2's complement value from the Target and add it to the current host address pointer.

If the Scatter/Gather list count is greater than one for the present command, then the sequencer will send a Message Reject message when the Modify Data Pointers message is received.

Status

The Status phase is handled by the sequencer, and the status byte is saved in a SCB location for examination later by the driver. If the status value or the Command Complete message is nonzero, the driver will be interrupted after the command complete message is received.

Command Complete Message

The Command Complete message is sent to the Initiator after the Status phase. This is handled by the sequencer and causes a command complete status bit to be set for the

firmware along with a CMDCMPLT interrupt. A Linked Command Complete message will cause a SEQINT with the appropriate code.

Scratch RAM Definition

The scratch RAM area contains general-purpose RAM area used during the execution of commands and to store configuration data which describes the system setup.

Multithreaded Operation

More than one target device may have commands open but disconnected. The sixteen SCBs are general purpose and may be used in any combination. The sequencer will match the Target/Channel/LUN when a new SCB ID is received from the QINFIFO. In order to preserve the order of execution for any Target/LUN combination, the restriction is made that no more than two SCBs with the same Target/Channel/LUN identification be loaded in the device. This restriction does not apply to tagged commands. Before any commands are executed or after a disconnection, the sequencer will look to see if there is another command to execute on the QINFIFO. If there is and the Target/Channel/LUN matches an open command, the ID will be pushed back on the QINFIFO. If there are more commands ready to be executed, they will be started. When reselection occurs a search for a disconnected command with the same Target/Channel/LUN is made and when found, the command is continued. In the case of tagged commands, the number of commands to the same Target/Channel/LUN may equal the space in the SCB Array. The commands will be sent with the tag value generated by the sequencer. Upon reselection, the sequencer will match Target/Channel/LUN/tag before completing the command.

Scatter/Gather

Scatter/Gather will be implemented as a part of the normal sequencer program. A Scatter/Gather transfer is characterized by using a list of data segments which the device uses to transfer data to or from the SCSI bus. The list is composed of 1 to 255 elements. Each element consists of a segment data pointer (4 bytes) and a segment byte count (4 bytes). All data transfers will be Scatter/Gather transfers. The Scatter/Gather list pointer is always valid and will be used to obtain the elements of the list. Each segment will be transferred as a stand-alone entity until the number of segments transferred is equal to the Scatter/Gather segment count. The segment byte count will be loaded into STCNT and HCNT, and the segment data pointer will be loaded into HADDR and SHADDR and the transfer will be started. When the SCSI counter is zero, the next segment data pointer and segment byte count will be read from host memory using the list pointer. The sequencer will then load the new values in the hardware and start the transfer in the normal manner. After a write operation, when STCNT is zero, and SDONE is set, SCSI FIFO is reset when SDMAEN is cleared. This is to clear any residual read ahead data in preparation for the next segment transfer. A read operation does not affect the SCSI FIFO when SDMAEN is cleared.

The working values of the list pointer and segment count value are stored in temporary scratch RAM area. The current value of the segment data pointer is gotten from SHADDR and the value of the segment byte count is gotten from SCNT. If a Save Data Pointers message is received before the Disconnect message, the working values will be saved in the SCB area. If a Disconnect message is received without a Save Data Pointers message, then the current value in the SCB area is not modified.

Tagged Queuing

In order to execute a tagged command, the tag enable bit in the control byte of the SCB must be set. The type of tag is also indicated by coding bits 0 and 1 of the same byte. A 00 means a simple queue is intended, a 01 means a Head of Queue message will be sent, and a 10 means an Ordered Queue message will be sent. The tag value will be the ID of the SCB. The Tag message will be sent after the ID message with the tag value if the tag enable bit is set. On reconnection, the search will be made for the disconnected SCB for the Target/Channel/LUN, and if the tag enable bit is set, a Tagged Queue message will be expected. Once the tag value is received, the correct SCB is chosen and the command is resumed.

Using the FIFO Threshold Control

The purpose of the data FIFO is to buffer the data in such a way to keep data streaming from one bus to the other. The rate of transfer of the SCSI and host buses will generally be different, and so the data FIFO is also providing the additional functions of speed matching and minimal host bus time usage by bursting data at the host bus maximum rate. Another independent variable is the latency of the host bus. It may take more or less time to gain control of the bus, depending on how busy other devices on the bus may be. In general one may include the host latency and transfer rate and generally set up the device for one of three situations of various degrees; a slow host and fast SCSI, slow SCSI and fast host, or equal-speed host and SCSI. The rates will be determined for each device by the driver and the information will be passed to the sequencer firmware.

DFTHRSH0/1 are defined (bits 6 & 7, PCISTATUS) to set the transfer threshold at different places. The possible settings are outlined below with suggestions on their usage.

- DFTHRSH1, DFTHRSH0: 0,0—SCSI is much faster than the host.

  Write operation - In this case, one would like to keep the host on the bus as much as possible, since it will be the limiting factor. In this case, the host transfer logic will start transmitting as soon as there is room in the FIFO, since we know the SCSI device will empty it faster than the host can fill it.

  Read operation - In this case, one would like to read data from the FIFO as soon as there is something in it, since we know the SCSI device will fill it up faster than the host can empty it.

- DFTHRSH1, DFTHRSH0: 0,1—Nearly equal speeds, SCSI is faster than or equal to host.

  Write operation - When the FIFO empties to 50% full, the host transfer logic will request the bus and transfer till the FIFO is full.

  Read operation - When the FIFO fills to 50% full, the host transfer logic will request the bus and transfer till the FIFO is empty.

- DFTHRSH1, DFTHRSH0: 1,0—Nearly equal speeds, SCSI is slower than or equal to host

  Write operation - When the FIFO empties to 75% empty, the host transfer logic will request the bus and transfer till the FIFO is full.

  Read operation - When the FIFO fills to 75% full, the host transfer logic will request the bus and transfer till the FIFO is empty.

- DFTHRSH1, DFTHRSH0: 1,1—Host is much faster than SCSI.

  Write operation - In this case, the FIFO is empty before the host transfer logic will request the bus in order to minimize the host bus activity, since we know it will be some time before the SCSI device can empty it.

  Read operation - In this case, the FIFO is full before the host transfer logic will request the bus, since we can empty it out long before the SCSI device can fill it up.

If the device is set up in XXXX mode, a value of 1,1 will be chosen as the default. In XXX mode, a value of 1,0 will be chosen as a default. During synchronous negotiation, the negotiated speed will be used to decide whether to change the default to 0,0 if in XXX mode.

Contingent Allegiance

In the event that an error occurs on the Target, a check condition will be sent to the Initiator in the status byte. In this case, sense information will be kept by the Target pertaining to the command which was in error for the Initiator which sent the command. This information will be kept until the next command is sent. The sequencer will interrupt the driver and pause upon receipt of any nonzero status from the Target after the command completes. The driver will get all information from the SCB and then reload the SCB area with a SCSI Sense command. The driver will then restart the sequencer at the point where it will execute the Sense command.

Abort

When a driver receives an abort request, the command could be in several states of execution. It may be in the driver's own queue, in the QINFIFO, in the SCB Array but disconnected, or active on the SCSI bus. If the command is in the driver's own queue, it need only remove it and report completion. If it is not there, the driver will pause the sequencer and search the QINFIFO first. If the command is there, the driver need only remove that entry from the queue and unpause the sequencer. If the command is in the SCB Array and either waiting for selection or disconnected, the driver need only clear those status bits. When the sequencer responds to the selection or reselection, it will discover that there is no command available and will issue the Abort message on the SCSI bus. If the command is active at the time, the driver will need to recover by sending an Abort message, completing the command, or resetting the SCSI bus.

Retry on Busy

There are occasions when a SCSI command will terminate with a busy bit set in the status byte. The sequencer will interrupt with nonzero status. The driver will handle the option of retrying the command or reporting the error to the original caller.

Command Linking

SCSI Command Linking may be implemented by the driver. The sequencer will respond with an Unknown Message In interrupt. The driver will reload the SCB area with the new SCB and restart the sequencer at the entry which will execute the new command.

Target Mode

SCSI Target mode is supported by the SCSI block in the AIC-7870. The initial firmware/driver does not support Target mode, but may be implemented with some code to handle the Select-In sequence on the SCSI bus. The sequencer would respond to selection, accept the ID and SCSI Command, and then disconnect. Detection of a Select-In would interrupt the driver to pass the Initiator/LUN information. The driver would prepare a Target command to pass data and complete the handshaking of the command. This requires additional sequencer code.

▼▼▼▼▼ 7

Wide selection

There are several problems that arise when implementing 16-bit selection, and mixing 8 and 16-bit devices. The problem is how to validate the selection before responding to it under all the conditions that may occur. Consider a 16-bit Initiator and 8-bit Target on a bus. When the Target reselects, the upper 8 bits are not driven which causes a parity error to occur on the upper byte. In the case of a selection with the Initiator having an ID in the upper byte, the Target will be selected in Single-initiator mode and will not disconnect. An 8-bit Initiator selecting a 16-bit Target has similar problems. One solution is to decode the data bits to allow only certain combinations before responding to the selection. Following is a table of allowable low and high byte combinations that may occur during valid selections. The selections include 16-to-16, 8-to-16, 16-to-8, and single bit. The logic is as follows as shown in Table 7-1.

Table 7-1. Valid Low Byte and High Byte Combinations

Valid Combinations	Low Byte Data / Parity	High Byte Data / Parity
16-to -16, OR 16 single bit	0 bits/1	2 bits/1
16-to-16, 16-to-8, OR 8 single bit	1 bit/0	1 bit/0 or 0 bits/x
16-to-16, OR 8-to-16	2 bits/1	0 bits/x

Differential Controls

The differential controls have been defined for the purpose of eliminating any external uP address decoding, and to make the controls transparent to the driver. Three groups of registers have been defined which are loaded by the device during various phases of operation. These groups are the ID, Arbitration/Selection, and Operation. The ID group is loaded when the target ID is written to the internal register SCSIID. The Arbitration/Selection group is written at various times during that operation. ENARB drives the device ID on the SCSI bus during Arbitration. ENBSY drives BSY on the bus directly. ENSEL drives SEL on the bus directly. The third group contains signals which determine whether a successful Selection or Reselection Out or In was done. ENINIT is set when a successful Selection Out or Reselection In was accomplished. ENTARG is set when a successful Selection In or Reselection Out was accomplished. These signals are used to gate REQ, ACK, C/D, I/O, and MSG at the proper time. I/O is used to determine the direction but does not drive the SCSI bus unless validated by ENINIT or ENTARG.

Note: Use of DIFACTNEGEN in the Configuration DEVCONFIG register will delete the requirement to provide standard SCSI termination between the AIC-7870 and the differential devices, only a light pull-up is needed. Chip reset will cause all values to be loaded with zero.

I/O Decodes

Most registers will be accessible to both the sequencer (SCSI PhaseEngine) and the driver. There are some exceptions, however, where some registers will be accessible to the driver but not the sequencer and vice versa. Also there are some registers which the driver should be allowed to read or write without disturbing the sequencer (no pause). Below is a list of the exceptions.

- Host Control (HCNTRL), read or write by host only without pause or AAP
- Device Space Vendor ID (DSVENDID)[1:0], read by host only without pause or AAP
- Device Space Device ID (DSDEVID)[1:0], read by host only without pause or AAP
- Device Space Clear Interrupt (CLRINT), write by host only without pause or AAP
- Device Space Interrupt Status (INTSTAT), read by host only without pause or AAP, concurrent write by sequencer
- Device Space Error Status (ERROR), read by host only without pause or AAP
- Device Space Queue Out FIFO (QOUTFIFO), read by host only without pause or AAP, concurrent write by sequencer
- Device Space Queue Out Count (QOUTCNT), read by host only without pause or AAP, concurrent write by sequencer
- Device Space SINDIR is not usable by the host
- Device Space DINDIR is not usable by the host
- All Configuration space registers, read/write by host only without pause or AAP
- External ROM space, read/write by host only with pause or AAP

8 Test Features

About This Chapter

Read this chapter to find out

- The hardware test modes and features used in the AIC-7870 to facilitate production testing
- A detailed explanation of each of the test groups and configurations

▼▼▼▼▼ 8

Overview

This section is a description of the Hardware Test modes and features used in the AIC-7870 to facilitate production testing. It contains a description of the special hardware configurations designed into the AIC-7870. A complete list of the functional and other tests written for this purpose is contained in the *AIC-7870 Test Definition Document*.

The AIC-7870 consists of six basic sections: SCSI, Sequencer (SCSI PhaseEngine), Host, Data FIFO, SCB RAM, and Scratch RAM. All registers are available to the host computer (except 2) and to the sequencer (except for 11), but not at the same time. Most of the chip may be tested through the host interface. The rest can be tested by loading a Sequencer Test program and running it.

The sequencer RAM has a special test mode which allows the detection of missing or weak transistors in the RAM cell which otherwise may not be discovered. These tests are described in a later section.

The AIC-7870 contains special circuitry to help test input levels and output current levels.

Most of the signals connect to buses and should contain additional capacitive and resistive loading. These are detailed in a later section.

Some of the pins will change definition under certain test conditions. This is to allow the tester to see internal signals to shorten test time.

Test Register Description

This section summarizes the registers which have been implemented specifically to enhance the testability of the device. The usage of these registers is detailed in the following sections.

The following conventions are used throughout this section:

- **set:** Indicates that the bit was loaded with a 1
- **cleared:** Indicates that the bit was loaded with a 0
- **(0):** Indicates that the bit is cleared when the reset pin is active
- **(1):** Indicates that the bit is set when the reset pin is active
- **(x):** Indicates that the bit is in an unknown state after the reset condition

SCSI Test Control (SCSITEST)

Type: R/W
Address: I-034Fh, E-zC0Fh, C-0Fh

This register is used to force test modes in the SCSI Module Logic.

0Eh SCSITEST W	
7	
6	
5	
4	DATALOOP
3	
2	RQAKCNT
1	CNTRTEST
0	CMODE

Bit		Name	Definition
7-5	(0)	Not Used	
4	(0)	DATALOOP	When set to one, the SCSI data transfer may be controlled to transfer from the data FIFO through the SCSI block to the sequencer or in the reverse direction to test the data path and controls. The sequencer, in effect, acts like a Target to control the transfer.
3	(0)	Not Used	
2	(0)	RQAKCNT	This bit is used to select STCNT or RQAKCNT values to test counter activity.
1	(0)	CNTRTEST	When set to one, the SCSI transfer counter STCNT and the selection time-out counter SELTIMER are put into a mode where they count down at the input clock rate, and the SCSI host address counter SHADDR is put into a mode where it counts up at the input clock rate.
0	(0)	CMODE	When set to one, forces a stage-to-stage carry true in STCNT, SHADDR, and SELTIMER. During the Transfer Count test, the counter contents can be monitored by reading the desired stage.

Selection Time-out Timer (SELTIMER)

Type: R
Address: I0358h, E-zC18h, C-18h

This register is used to monitor the state of the hardware selection time-out timer.

0358 SELTIMER	
7	CLKOUT
6	
5	STAGE 6
4	STAGE 5
3	STAGE 4
2	STAGE 3
1	STAGE 2
0	STAGE 1

Bit		Name	Definition
7	(0)	CLKOUT	See register bit definition section for details.
6	(0)	Not Used	Always reads 0.
5	(0)	STAGE 6	(/2, output)
4	(0)	STAGE 5	(/2, output)
3	(0)	STAGE 4	(/2, output)
2	(0)	STAGE 3	(/10, output)
1	(0)	STAGE 2	(/256, output)
0	(0)	STAGE 1	(/256, output)

Special Function (SFUNC)

Type: R/W (CIOBUS)
Address: I-135Fh, E-zC9Fh, C-9Fh

SFUNCT bits [4:0] select certain sections of the chip for test purposes. SFUNCT register bits 0 and 3 can only be set (=1) in a component test environment as test logic will be activated that redefines output pad functions. Writing to the SFUNCT register to set bits 0 and/or 3 (=1) requires DIFFSTRB# to be pulled low during the previous RST# assertion to activate an internal TESTBITEN status. This status will remain active until the next RST# assertion with DIFFSTRB# is allowed to float, the normal condition has an internal pull-up. When TESTBITEN status is active, the value written to SFUNCT register will not take effect until 2 PCLKs after the PCLK that deasserts TRDY#. When TESTBITEN is not active, the written value will take effect on the PCLK that deasserts TRDY#. This delay, when present, must be taken into account when reading SFUNCT register or performing the selected test. This delay is provided so that the component tester that is performing the Write to Select tests, that will redefine the pin functions, will have an opportunity to switch from driving the pins that will be redefined to monitoring those pins without contention. SFUNCT register is cleared to 00h when RST# is asserted except when either Input Pad or Process Check OSC test is selected. To clear Input Pad or Process Check OSC tests, perform a write with value 00h or the next test value to the SFUNCT register,

or write to CHIPRST=1. PCI pins are always high impedance when RST# is asserted for normal operation (except when component tests Input Pad or Process Check OSC test is selected).

SFUNCT R/W	
7	SFUNCT2
6	SFUNCT1
5	SFUNCT0
4	TESTRAM
3	TESTHOST
2	TESTSEQ
1	TESTFIFO
0	TESTSCSI

Test Group Description

The following sections explain in greater detail each of the test groups and configurations. Table 8-1 explains the defined values which are loaded into SFUNCT and the tests they represent. CSDAT(7:0) is an internal bus which is made available to external pins in order to facilitate the test process.

Table 8-1. Test Group Description

Data (7:0)	HTEST	REF	Test Monitor	Definition
00	-	---	N	Normal Operation
02	-	abe	N	Enable data FIFO Quad-word Write
04	-	ae	N	Sequencer ALU Output Monitor Test
14	-	ae	N	Sequencer Single Port RAM Stress Test
84 *	-	abde	M	CSDAT[7:0] Monitor Test (with normal operation)
D4	-	ae	N	Sequencer RAM Parity Test
F4	-	ae	N	Sequencer RAM Data Test
0C	4	ab	M	Sequencer ALU Output Monitor Test
1C	0	a	M	Sequencer Single Port RAM Stress Test
29	1	acd	I	Input Pad Test
2B	1	a	I	Process Check Oscillator Test
49	2	acd	N	Output Pad High Voltage Test
69	3	acd	N	Output Pad Low Voltage Test
88	4	a	M	HCNT[2:0] and Decode Test
A8	5	a	M	LHADDR[3:0] and Decode Test
DC	6	a	M	Sequencer RAM Parity Test
FC	7	a	M	Sequencer RAM Data Test

a = PHOST b = Sequencer c = SCSI d = Memory Port
e = Tests modes that are also usable in normal system environment
N = None M = MDA[7:0] I = IRQA#
* = Usable only with internal SCBs and no access of the Memory Port

Input Pad Testing

Input Pad Test

When a write is performed to place the AIC-7870 in this test mode, all input pads are connected in an ANDed string which is connected to output pad IRQA# for input threshold testing. This allows one input to be tested at a time. When all inputs in the string are at the input high level, IRQA# output will also be at an output high level. Placing any one input at a input low level will cause IRQA# output to also go to an output low level.

> **Note:** PCLK input must be held at a low level to allow changes on the level of other inputs to be seen on IRQA# output pin. PCLK input itself may be tested by monitoring RAMPS# output when it is selected to output buffered PCLK when sampled at a high level by IDDAT load clock (LED#) number 4, following RST# assertion.
>
> While in the AIC-7870 Input Pad test selection, the reset of the SFUNCT register caused by RST# has a modified action to allow testing of the RST# input and at the same time prevent clearing of the Input Pad test selection. A write to SFUNCT register or CHIPRST may be used to exit from this test mode.

ANDed Input Pad Strings List

> **IRQA#**:GNT#, PCLK, CBE3#, AD31 - AD24, IDSEL, CBE2#, AD23 - AD16, FRAME#,IRDY#, TRDY#, DEVSEL#, STOP#, PERR#, PAR, CBE1#, AD15 - AD08, CBE0#, AD07 - AD00, RST#, PREQ#, IDDAT, IO#, REQ#, CD#, SEL#, MSG#, RESET#, ACK#, BSY#, ATN#, SCDPL#, SCD7# - SCD0#, SCDPL#, SCD15# - SCD9#,SCDPH#, WIDEPS#, DIFFSTRB#, MD0 - MD7, MDP, RAMPS#, EXTARBACK#, EXTARBREQ#

Process Check Oscillator Testing

Process Check Oscillator Test

When a write is performed to place the AIC-7870 in this test selection, the AIC-7870 located die will configure the input pad ANDed input strings (reference the AIC-7870 Input Pad test) into one long string containing one inversion with the ends of the string connected together to form a ring oscillator. The current ring oscillator contains cells located all over the die along with long metal interconnects. An output from the ring (reference to AD00 in the ANDed string) is muxed into the LHADDR[3:0] 32 bit counter clock input when this test is selected to allow monitoring the rate at which the ring oscillates. The resulting count in LHADDR[3:0] will determine the processed die speed in relation to our standard cell library values of fast, typical or slow. The processed die speed value will determine the clock rate selection of 8 or 10 MIPS for the sequencer block. The tester will start the test period timing from the PCLK that deasserts TRDY# following the write that entered this test mode with all other inputs at a high level as in Input Pad test.

After a suitable delay, change input EXARBACK# to a low level to pause the oscillation and read the LHADDR[3:0] value to determine the die speed value. The output is available on IRQA#, the same as in the Input Pad test.

> **Note:** While the Process Check Oscillator test is selected, inputs RST#, PCLK and CLKIN in the assigned input strings are forced to appear at a high level internally to enable the ring oscillator to free run.

Output Pad Voltage High Test

When a write is performed to place the AIC-7870 in this test selection, all output pads are enabled and controlled to place the outputs at an output high level two PCLKs after the PCLK that deasserts TRDY#.

Note: Care should be taken to not exceed the total expected current for the normal quantity of driven outputs when applying output test load current.

Read or write cycles are prohibited after the write that placed the AIC-7870 in this selection and assertion of RST# is required to exit from this test selection.

The SCSI bus must be placed into Wide mode before entering this test selection so that the high data byte and its parity bit will also be tested.

Output Pad Voltage Low Test

When a write is performed to place the AIC-7870 in this test selection, all output pads are enabled and controlled to place the outputs at an output low level two PCLKs after the PCLK that deasserts TRDY#.

Note: Care should be taken to not exceed the total expected current for the normal quantity of driven outputs when applying output test load current.

Read or write cycles are prohibited after the write that placed the AIC-7870 in this selection and assertion of RST# is required to exit from this test selection.

The SCSI bus must be placed into Wide mode before entering this test selection so that the high data byte and its parity bit will also be tested.

Host LHADDR Testing

Host LHADDR(3:0) And Decode Test

When this test is selected the LHADDR address counter is segmented into bytes which are incremented with each rising edge of PCLK.

Note: The carry-in logic between bytes and the logic which normally controls counting by 1, 2, 3, 4 in master operation are not tested in this test selection.

Performing reads from LHADDR[3:0] will allow access to the counter values. Timer LATT is clocked by PCLK. A write to SFUNCT register or RST# may be used to exit from this test mode. The LHADDR counter byte carry-outs are observable on the MA[7:0].

MA0 (CSDAT0) = LHADDRCOA
MA1 (CSDAT1) = LHADDRCOB
MA2 (CSDAT2) = LHADDRCOC
MA[7:3] (CSDAT[7:3]) = 0h

Host HCNT Testing

Host HCNT(2:0) And Decode Test

When this test is selected the HCNT byte counter is segmented into bytes which are decremented with each rising edge of PCLK.

> **Note:** The carry-in logic between bytes and the logic which normally controls counting by 1, 2, 3, 4 in master operation are not tested in this test selection.

Performing reads from HCNT[2:0] will allow access to the counter values. A write to SFUNCT register or RST# may be used to exit from this test mode. The HCNT counter byte carry-outs are observable on the MA[7:0].

MA0 (CSDAT0) = HCNTCOA
MA1 (CSDAT1) = HCNTCOB
MA2 (CSDAT2) = HCOTTH
MA3 (CSDAT3) = HCOTF
MA4 (CSDAT4) = HCOTE
MA5 (CSDAT5) = HCLTCL
MA6 (CSDAT6) = SCL
MA7 (CSDAT7) = ECL

Memory Port Testing

No special test logic is provided for the Memory Port. Exercise each of the functions supported by the Memory Port.

ROM/EEPROM Read Test

Perform ROM/EEPROM byte and DWD reads from a ROM/EEPROM address. Four Memory Port read cycles should occur for each read. The EXROMCTL register must be loaded with a ROM address with EXROMEN active, MSPACEEN active in the COMMAND register, and EXTARBACK# asserted.

Next, perform a read cycle with EXROMEN inactive, then with MSPACEEN inactive. These accesses should result in a master-abort termination, with no Memory Port read cycles performed.

EEPROM Write Test

Perform EEPROM byte write accesses to a ROM address. One Memory write cycle should occur for each write. The EXROMCTL register must be loaded with a ROM address with EXROMEN active, MSPACEEN active in the COMMAND register, and EXTARBACK# asserted.

> **Note:** DWD writes will result in a target-abort termination.

SEERDY Timer Test

Write a 0h value to the SEECTL register. SEERDY will now be inactive (=0) when reading the SEECTL register for 800 nsec then will become active. Write to SEECTL to activate SEEMS with EXTARBACK# deasserted. SEERDY will now be inactive (=0) when reading the SEECTL register. Wait for greater than 800 nsec SEERDY will still be inactive. Assert

EXTARBACK# then ready SEECTL register, SEERDY should now be active (this indicates that the Memory Port has been granted to the AIC-7870 for SEECTL and/or BRDCTL register activity).

Memory Data Bit Control Test

After SEERDY is active following setting SEEMS active, exercise the Memory Port data and parity bits by use of the SEECTL and BDRCTL registers bits.

EXTREQLCK Test

Activating (=1) EXTREQLCK then making an access of the Memory Port will cause EXTARBREQ# to remain asserted even when no accesses to the Memory Port are being made. Only after EXTREQLCK is set inactive will EXTARBREQ# become deasserted.

CSDAT Monitoring Test

CSDAT(7:0) Monitoring Test

Outputs MA[7:0] normal output replaced with CSDAT(7:0) with the rest of the chip operation continuing in the normal state when the SFunction register value is 84h. CSDAT is an internal source data bus. Data appears on this bus when a read from some I/O decoded address is done. This is provided to facilitate part testing by reducing the number of cycles run on failing parts and to provide internal signals to the outside world. The bit relationships are as follows:

Pin Name	Signal Name
MA7	CSDAT7
MA6	CSDAT6
MA5	CSDAT5
MA4	CSDAT4
MA3	CSDAT3
MA2	CSDAT2
MA1	CSDAT1
MA0	CSDAT0

Sequencer Testing

This capability is also provided for certain test modes when the SFunction register contains values of 0ch, 1ch, 88h, A8h, Dch, and Fch.

Sequencer ALU Test

The sequencer is loaded with code to test the internal logic of the block. These tests check the operation of the AND, OR, ADD, and JUMP instructions. CSDAT[7:0] is monitored to detect any unexpected results from the self-check.

Sequencer RAM Parity Test

Data is written to the sequencer RAM and read back. The parity bit is available on CSDAT0.

Sequencer RAM Data Test

Various patterns are written to the sequencer RAM and read back for verification.

Sequencer RAM Leakage Test

A special test sequence is performed to verify proper operation of RAM cell data storage capability.

The AIC-7870 TESTRAM procedure is run at wafer/package level and may also be run in demand type diagnostic level in the system. Its purpose is to write a test data value to a block RAM location. Place TESTRAM bit in the active state. Then write to the same block RAM location the second time with the same data. This will cause the CIOBUS logic in the block containing the RAM to enter a TESTLOCK state, (which latches the supplied CDDAT[7:0] data, the CDADR[7:0]- address, the RAM WE- input active and places the RAM cell TEST- input in the active state). The RAM when accessed in this manner will stress the addressed RAM location storage capability. The recommended minimum stress time period is 500ns. The stressed RAM location is then read to verify the stored data is still the same as originally stored. While in the TESTLOCK state the block containing the RAM is disconnected from the destination side of the CIOBUS with normal operation continuing on the source side of the CIOBUS. Placing TESTRAM in the inactive state will clear the latched states and return the block's destination side of the CIOBUS back to normal. Multiple blocks containing RAM may have a location stressed at the same time to shorten the overall test time. Block RAM that may be read 8-bits at a time may be verified directly with the CSDAT bus. RAM that is wider than 8bits or not dual-ported will need additional logic to provide the compare function, with the compare output status accessed by the CSDAT bus.

SCSI Testing

- When TESTSCSI is active, the external SCSI bus pins may be redefined for test purposes (reference SFUNCT register values 29, 2B, 49, 69). This redefinition may only be performed in a test environment where the redefinition will not affect system operation. TESTSCSI is placed in the active state (=1) when TESTBITEN is active by performing a write cycle to the SFUNCT register with D0 asserted).
- The SCSI block also has a test register within its own block that operates independently of SFUNCT register.

FIFO Testing

When TESTFIFO is active, CIOBUS writes to DFDAT will cause 64-bit writes to occur in the data FIFO with the 8-bit data value on the CSDAT bus written in parallel to each byte of the 64-bit data FIFO location pointed to by DFWADDR value. The DFWADDR value will increment following each write. This test feature shortens the data FIFO RAM test data load time for initializing data FIFO RAM parity and doing RAM data tests or for writing a constant data value to all RAM locations.

- DFDAT must be enabled for writing from the CIOBUS (reference DFCNTRL register).
- Reading DFDAT by the sequencer or the host is always 8 bits per read.

9 Electrical Information

About This Chapter

Read this chapter to find out

- Electrical information about the AIC-7870
- PCI bus timing diagrams for the AIC-7870

▼▼▼▼▼ 9

Absolute Maximum Ratings

Storage Temperature: -65°C to 150°C
Power Supply Voltage: 0 to 7 V
Voltage on any Pin: -0.5 to VCC+0.5 V

Operating/Test Conditions

Ambient Temperature: 0°C to 70°C
Supply Voltage: 4.5 to 5.5 V
Supply Current:

Active	50 mA
Paused	36 mA
Power Down	5 mA
t_f	< 5 ns
t_r	< 5 ns
C_L	50 pf unless otherwise noted

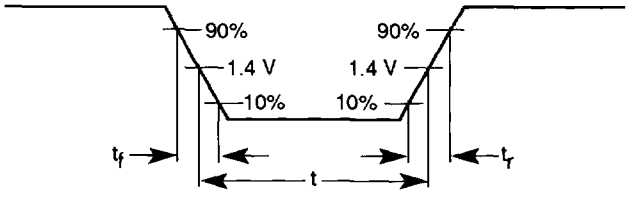
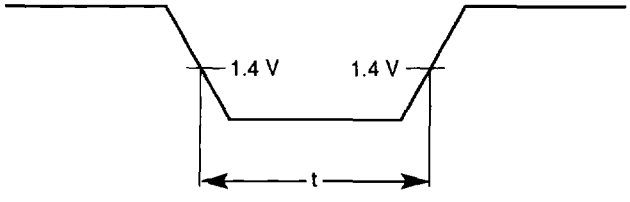


Figure 9-1. A.C. Input Conditions



C = 50 pf unless otherwise noted

Figure 9-2. A.C. Output Conditions

DC Parameters

PCI

Ta = 0°C to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Test Condition	Notes
V_{cc}	Supply Voltage	4.75	5.25	V		
V_{ih}	Input High Voltage	2.0	Vcc+0.5	V		
V_{il}	Input Low Voltage	-0.5	0.8	V		
I_{ih}	Input High Leakage Current		70	µA	Vin = 2.7	1
I_{il}	Input Low Leakage Current		-70	µA	Vin = 0.5	1
V_{oh}	Output High Voltage	2.4		V	Iout = -2 MA	
V_{ol}	Output Low Voltage		0.55	V	Iout = 3 MA	2
V_{ol}	Output Low Voltage		0.55	V	Iout = 6 MA	3
C_{in}	Input Pin Capacitance		10	pf		4
C_{clk}	PCLK Pin Capacitance	5	12	pf		4
C_{IDSEL}	IDSEL Pin Capacitance		8	pf		4
L_{pin}	Pin Inductance		20	nH		

[1] Input leakage include hi-Z output leakage for bidirectional buffers with tri-state outputs.
[2] Signals without pull-up resistors (AD[31:00], CBE[3:0]#, PAR).
[3] Signals with pull-up resistors (FRAME#, IRDY#, TRDY#, DEVSEL#, STOP#, PERR#, SERR#).
[4] At 1 MHz.

Signal trace length on printed circuit board upon which the AIC-7870 is installed is 1.5-inches maximum from the package pin to the PCI bus (speedway) connection.

SCSI

Ta = 0°C to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Test Condition	Notes
V_{cc}	Supply Voltage	4.75	5.25	V		
I_{il}	Input Low Leakage Current		+/-50	µA	Vin = 0.5 to VCC	1, 2
V_{ih}	Input High Voltage	2.0		V		
V_{il}	Input Low Voltage		0.8	V		
V_{ihys}	Input Hysteresis	0.2		V		
V_{oh1}	Output High Voltage	2.4		V	I_{o1} = -400 µA	3
V_{oh2}	Output High Voltage	2.4		V	I_{o2}= -2MA	
V_{ol}	Output Low Voltage		.5	V	I_{o3}= 48MA	

[1] Input leakage include hi-Z output leakage for bidirectional buffers with tri-state output (SCD[15:0]#, SCDPL#, SCDPH#, CD#, IO#, MSG#, REQ#, ACK#, RESET#, SEL#, BSY#, ATN#).
[2] Inputs are controlled to limit input current, see STPWEN.
[3] Outputs (BSY#, SEL#, RESET#).

SCSI Support Pins

Ta = 0°C to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Test Condition	Notes
Vcc	Supply Voltage	4.75	5.25	V		
Vih	Input High Voltage	2.0	Vcc+0.5	V		
Vil	Input Low Voltage	-0.5	0.8	V		
Iil1	Input Leakage Current		10	µA	Vin = 2.7	1
Iil2	Input Leakage Current		-50	µA	Vin = 0.5	2
Iol	Output Leakage Current		+/-10	µA	Vout = 0.5 to Vcc	3
Voh1	Output High Voltage	2.4		V	Iout = -4 MA	3
Vol1	Output Low Voltage		0.5	V	Iout = 4 MA	4
Voh2	Output High Voltage	2.4		V	Iout = -24 MA	5
Vol2	Output Low Voltage		0.5	V	Iout = 24 MA	5

[1] Input leakage include hi-Z output leakage for bidirectional buffer with tri-state output (DIFFSTRB#) and input WIDEPS#.
[2] Input leakage includes internal pull-up resistor (DIFFSTRB#).
[3] Output (STPWCTL).
[4] Outputs (DIFFSTRB#, DIFFADR[1:0], DIFFDAT[3:0]).
[5] Output (LED#).

Memory Port

Ta = 0°C to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Test Condition	Notes
Vcc	Supply Voltage	4.75	5.25	V		
Iil1	Input Current Leakage		+/-10	µA	Vin - .4 to VCC	1
Iil2	Input Current Leakage		-50	µA	Vin - .4	2
Iol	Output Current Leakage		+/-10	µA	Vout = .5 to VCC	
Vih	Input High Voltage	2.0		V		
Vil	Input Low Voltage		.8	V		
Voh1	Output High Voltage	2.4			Io = -4mA	3
Voh2	Output High Voltage	2.4			Io = -8mA	4
Voh3	Output High Voltage	2.4		V	Io = -24mA	5
Vol1	Output Low Voltage		.5	V	Io = 4mA	3
Vol2	Output Low Voltage		.5	V	Io = 8mA	4
Vol3	Output Low Voltage		.5	V	Io = 24mA	5

[1] Input leakage include hi-Z output leakage for bidirectional buffers with tri-state outputs (MD[7:0], MDP).
[2] Input leakage includes internal pull-up resistor (EXTARBACK#, RAMPS#).
[3] Outputs (SEECS, RAMPS#, ROMCS#).
[4] Outputs (MD[7:0], MDP, MA[14:0], EXTARBREQ#).
[5] Outputs (RAMCS#, MRW).

Other Pins

Ta = 0°C to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Test Condition	Notes
Vcc	Supply Voltage	4.75	5.25	V		
I_{li1}	Input Current Leakage		+/-10	µA	Vin - .4 to VCC	1, 2
I_{li2}	Input Current Leakage		-50	µA	Vin - .4	2
Vih	Input High Voltage	2.0		V		
Vil	Input Low Voltage		.8	V		

[1] Input leakage CLKIN.
[2] Input leakage includes internal pull-up resistor IDDAT.

Signal Test Loads

Signals	Circuit Values		
SCD[15:0]#, SCDPH#, SCDPL#, RESET#, BSY#, SEL#, REQ#, MSG#, IO#, CD#, ATN#, ACK#	Capacitance Pull-up Resistor Pull-down Resistor	300 110 165	pf Ohm Ohm
FRAME#, TRDY#, IRDY#, STOP#, DEVSEL#, PERR#, SEER#	Capacitance Pull-up Resistor Pull-up Resistor Pull-down Resistor	50 936 2.7 —	pf max Ohm min KOhm typ —
AD[31:00], CBE[3:0]#, PAR, PREQ#, WIDEPS#	Capacitance Pull-up Resistor Pull-down Resistor	50 — —	pf max — —
IRQA#	Capacitance Pull-up Resistor Pull-down Resistor	50 1.5 —	pf max KOhm min —
ADIFFDAT(3:0), ADIFFADR(1:0), ADIFFSTRB, STPWCTL#, LED#, MA[14:0], MRW, MD[7:0], MDP, EXTARBREQ#, RAMPS#	Capacitance Pull-up Resistor Pull-down Resistor	50 — —	pf max — —
ROMCS#, RAMCS#	Capacitance Pull-up Resistor Pull-down Resistor	50 1 —	pf KOhm —
SEECS	Capacitance Pull-up Resistor Pull-down Resistor	50 — 1	pf max — KOhm

AC Parameters

PCI Pin V/I Curves

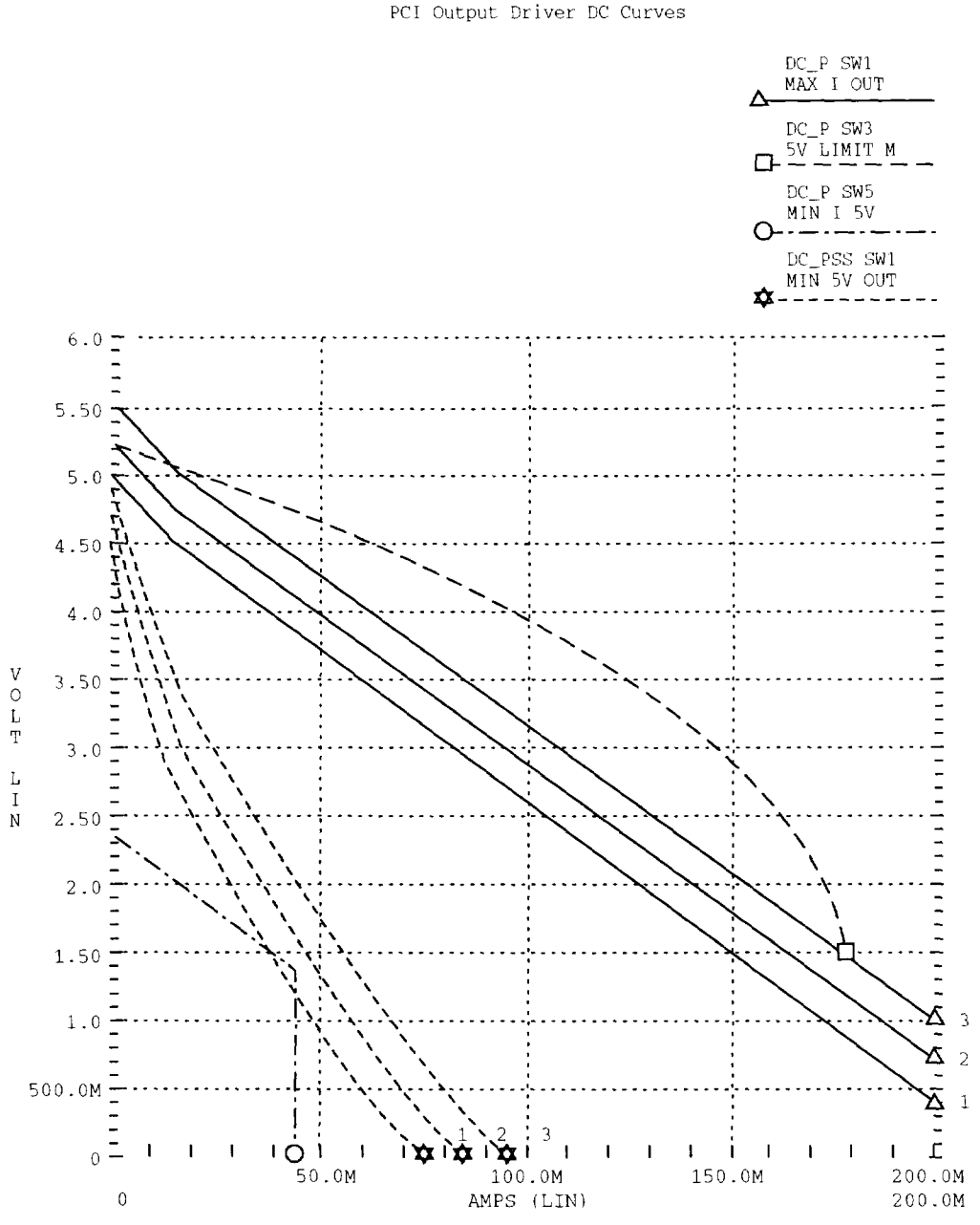


Figure 9-3. PCI Signal 5 Volt Pull-up Output V/I Curves

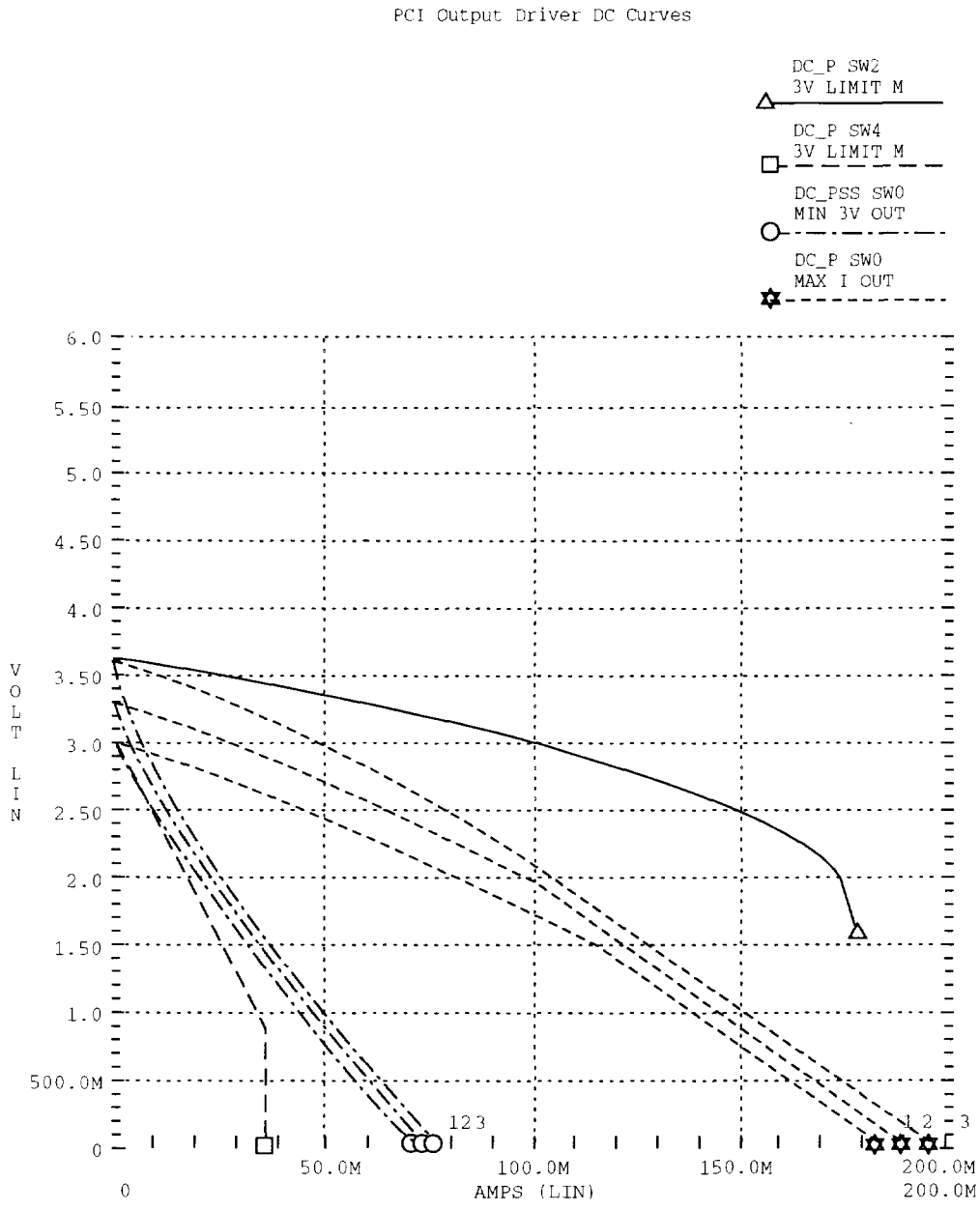


Figure 9-4. PCI Signal 3 Volt Pull-up Output V/I Curves

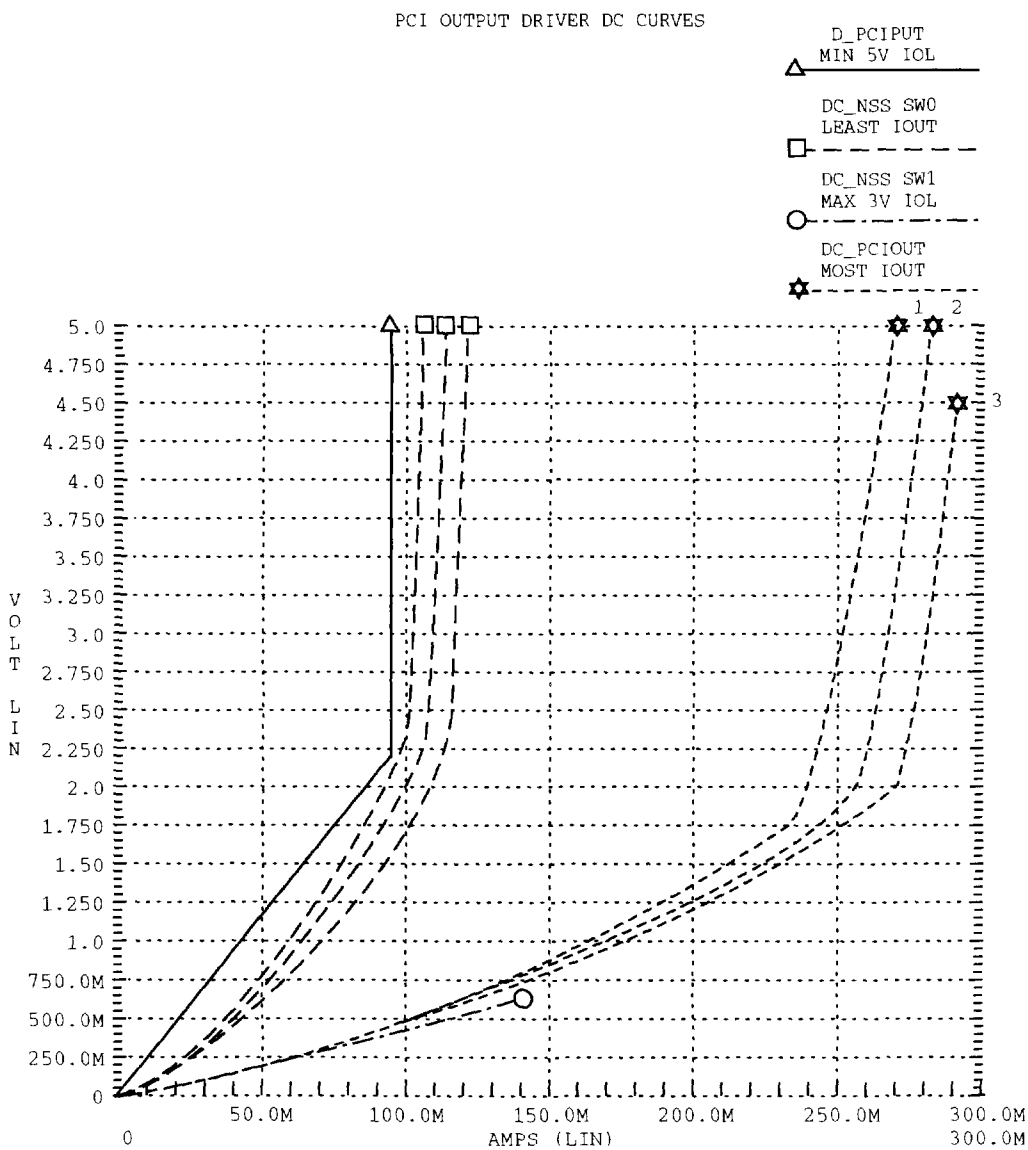


Figure 9-5. PCI Signal 3 Volt/5 Volt Pull-down Output V/I Curves

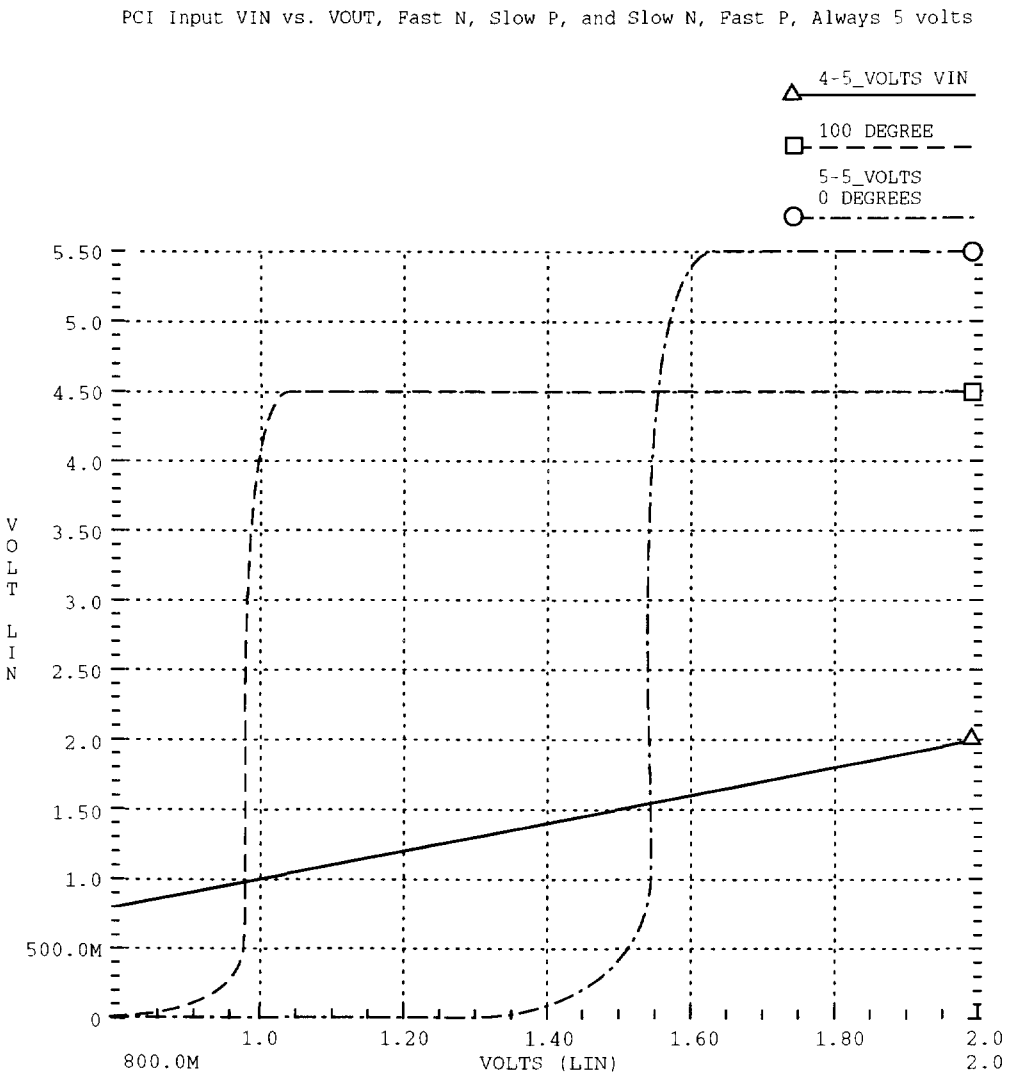


Figure 9-6. PCI Signal 5 Volt Input V/I Curves

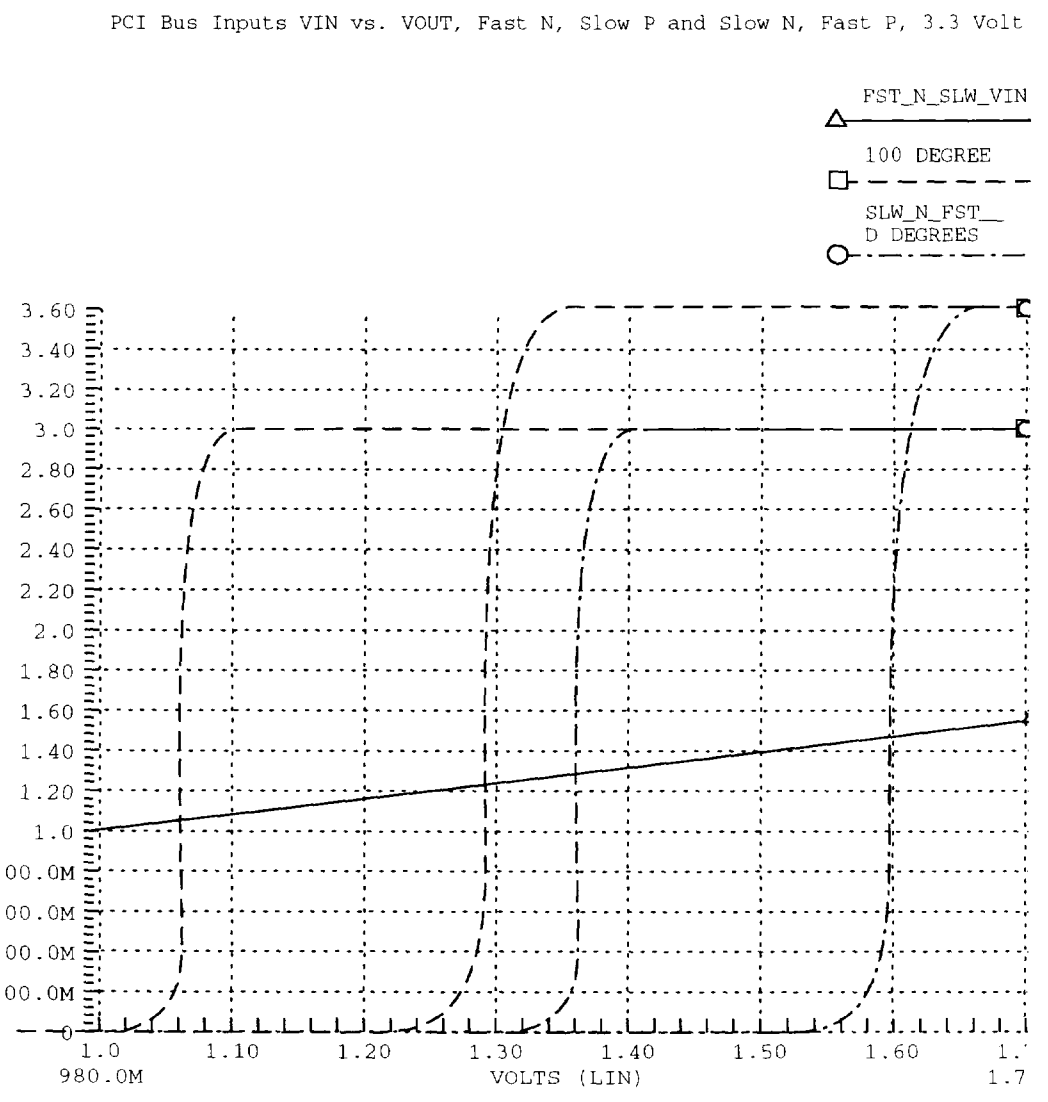


Figure 9-7. PCI Signal 3 Volt Input V/I Curves

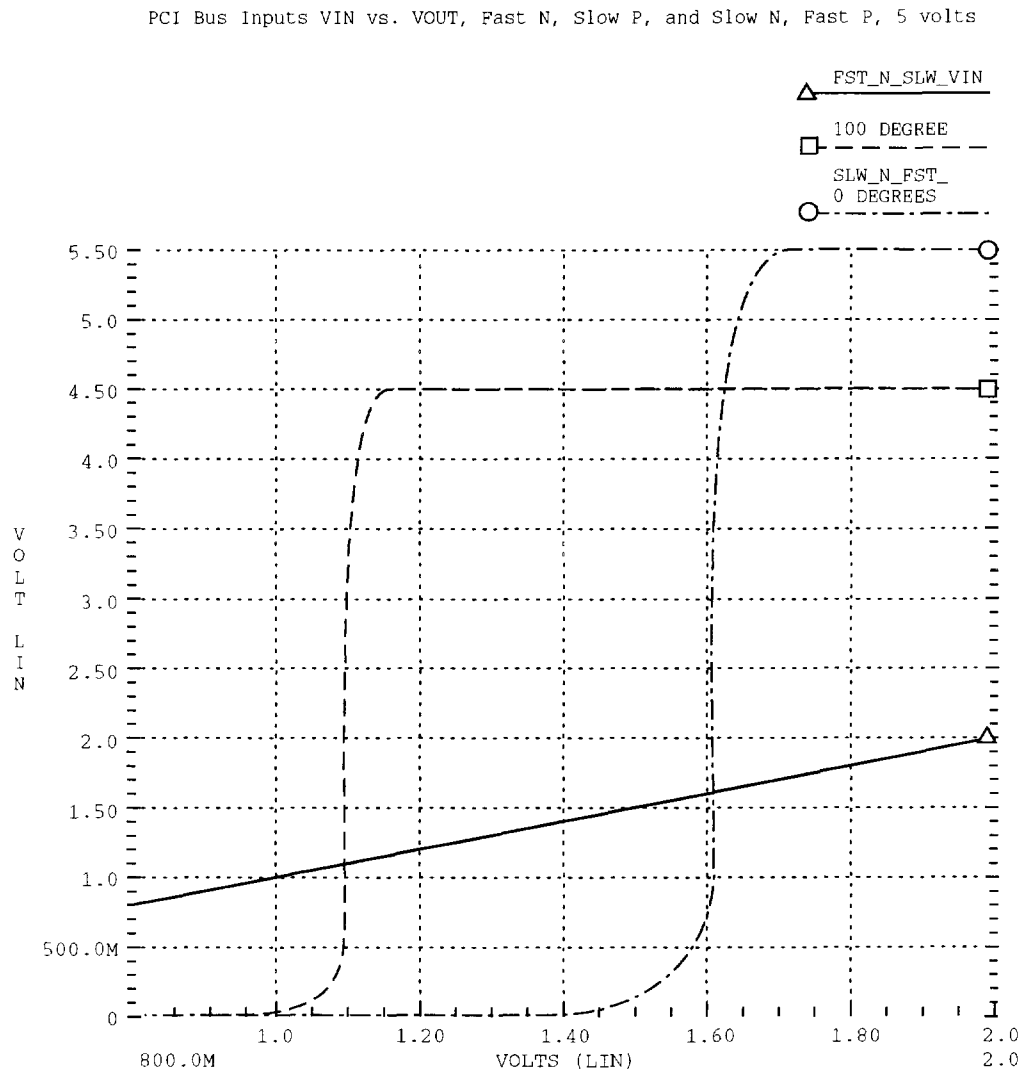


Figure 9-8. PCI Signal 3 Volt/5 Volt Input V/I Curves

Clock Timing

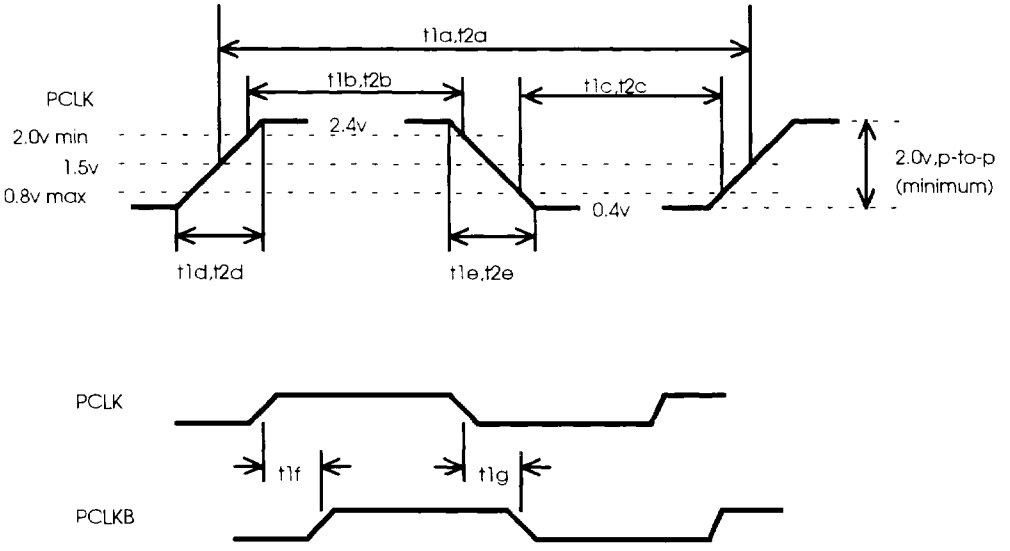


Figure 9-10. Clock Timing

Ta = 0° to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Figure	Notes
t1a	PCLK Period	30		ns	9-10	1
t1b	PCLK High Time	12		ns	9-10	1
t1c	PCLK Low Time	12		ns	9-10	1
t1d	PCLK Rise Time	1	4	ns	9-10	1
t1e	PCLK Fall Time	1	4	ns	9-10	1
t1f	PCLK rise to PCLKB rise		9.3	ns	9-10	1
t1g	PCLK fall to PCLKB fall		10.3	ns	9-10	1

[1] t1a referenced to 1.5V, other times referenced to 0.8V and 2.0V.
PCLKB reference RAMPS# pin. Active from point A in Figure 9-6.

Ta = 0° to 70°C
VCC = 5 V ± 5%
GND = 0 V

Symbol	Definition	Min	Max	Units	Figure	Notes
t2a	CLKIN Period	25		ns	9-10	1
t2b	CLKIN High Time	7		ns	9-10	
t2c	CLKIN Low Time	8		ns	9-10	
t2d	CLKIN Rise Time		3	ns	9-10	
t2e	CLKIN Fall Time		3	ns	9-10	

[1] t2a referenced to 1.5V, other times referenced to 0.8V and 2.0V.

PCI Bus Timing

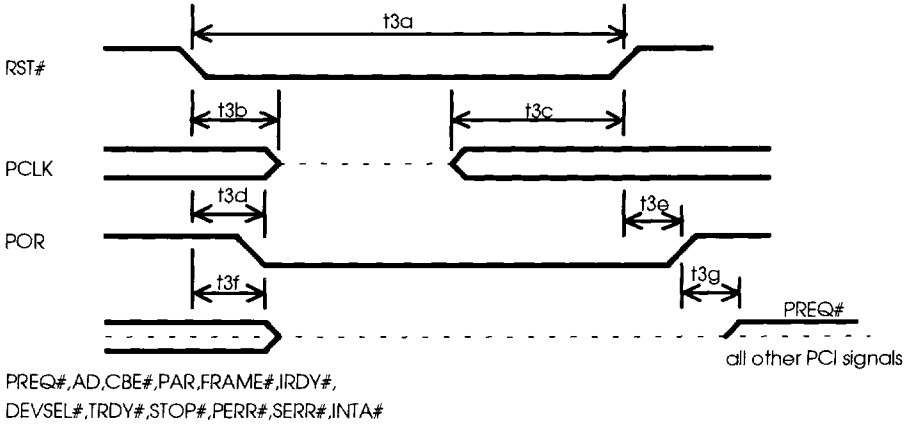
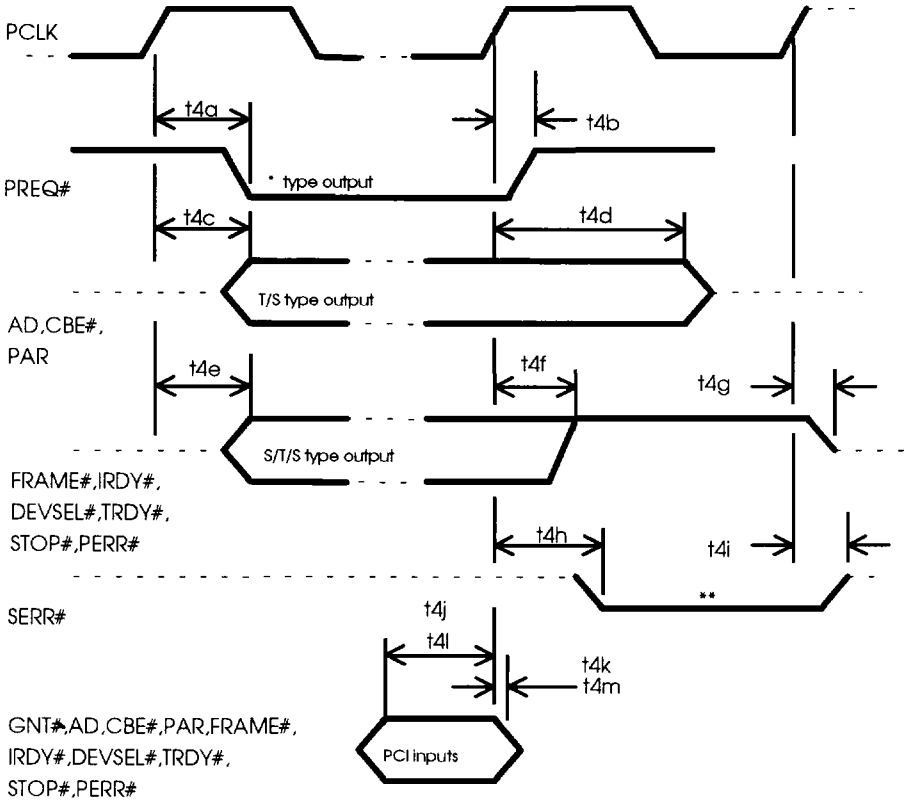


Figure 9-11. Reset Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t3a	RST# Period				9-11	1
t3b	PCLK active following RST# assertion	0		ns	9-11	
t3c	PCLK active preceeding RST# deassertion				9-11	2
t3d	POR active following RST# assertion		12	ns	9-11	3
t3e	POR inactive following RST deassertion				9-11	4
t3f	PCI Outputs Float Delay	2	28	ns	9-11	
t3g	PREQ# Output Enable Delay	2	11	ns	9-11	

[1] 8 PCLK cycles minimum.
[2] 6 PCLK cycles minimum.
[3] Internal reset active (asynchronous to PCLK).
[4] Internal reset inactive (synchronous to PCLK).



* = 'T/S' type output used as 'OUT' type output
(only floated when RST# is asserted).

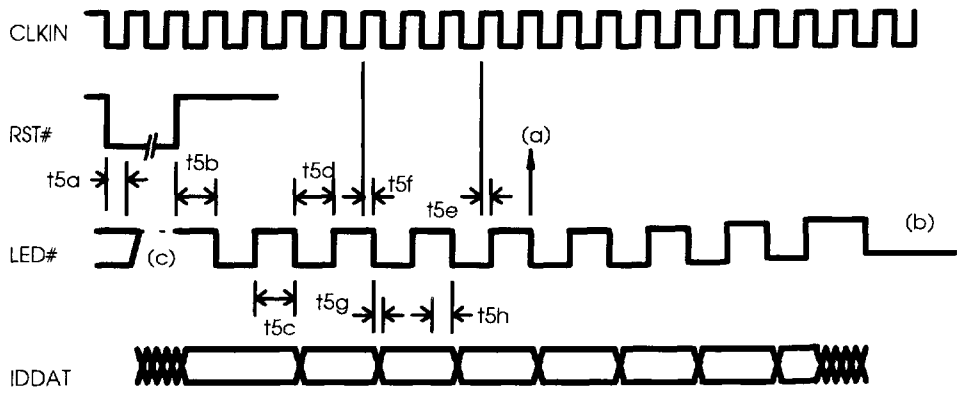
** = O/D type output

Figure 9-12. PCI Signal Input Output Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t4a	PREQ# Assertion Delay	2	12	ns	9-12	
t4b	PREQ# Deassertion Delay	2	12	ns	9-12	
t4c	AD[31:00], CBE[3:0]#, PAR Output Valid Delay	2	11	ns	9-12	1
t4d	AD[31:00], CBE[3:0]#, PAR Output Float Delay	2	28	ns	9-12	
t4e	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, PERR# Output Assertion Valid Delay	2	11	ns	9-12	1
t4f	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, PERR# Output Deassertion Valid Delay	2	11	ns	9-12	2
t4g	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP, PERR# Output Float Delay	2	28	ns	9-12	
t4h	SERR# Output Assertion Valid Delay	2	11	ns	9-12	
t4i	SERR# Output Deassertion Valid Delay	2	11	ns	9-12	
t4j	AD[31:00], CBE[3:0]#, PAR, FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, PERR# Input Valid Setup	7		ns	9-12	
t4k	AD[31:00], CBE[3:0]#, PAR, FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, PERR# Input Valid Hold	0		ns	9-12	
t4l	GNT# Input Valid Setup	10		ns	9-12	
t4m	GNT# Input Valid Hold	0		ns	9-12	

[1] Includes float to output enable delay.
[2] Starts s/t/s type output deassertion assurance period.



(a) = Note, this is the RAMPS# sample point for PCLKB output enable.

(b) = LED# is low until firmware is loaded, after which it goes high until a 7870 SCSI command is busy on the SCSI bus.

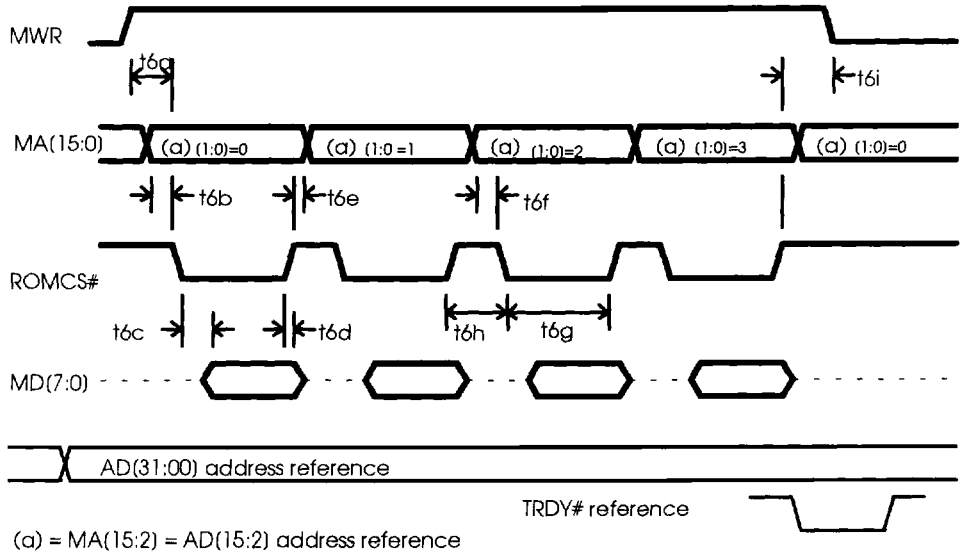
(c) = LED is in a float state.

Figure 9-13. PCI Device Identification Value Replacement Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t5a	RST# fall to LED# float delay		7	ns	9-13	
t5b	RST# rise to LED# first clock fall	2 (CLKIN period)	4 (CLKIN period)	ns	9-13	
t5c	LED# (IDDAT) clock low time	25		ns	9-13	
t5d	LED# (IDDAT) clock high time	25		ns	9-13	
t5e	CLKIN rise to LED# rise	9		ns	9-13	
t5f	CLKIN rise to LED# fall	8		ns	9-13	
t5g	IDDAT data hold from LED# rise	0		ns	9-13	
t5h	IDDAT data valid to LED# rise	5		ns	9-13	

Memory Port Timing



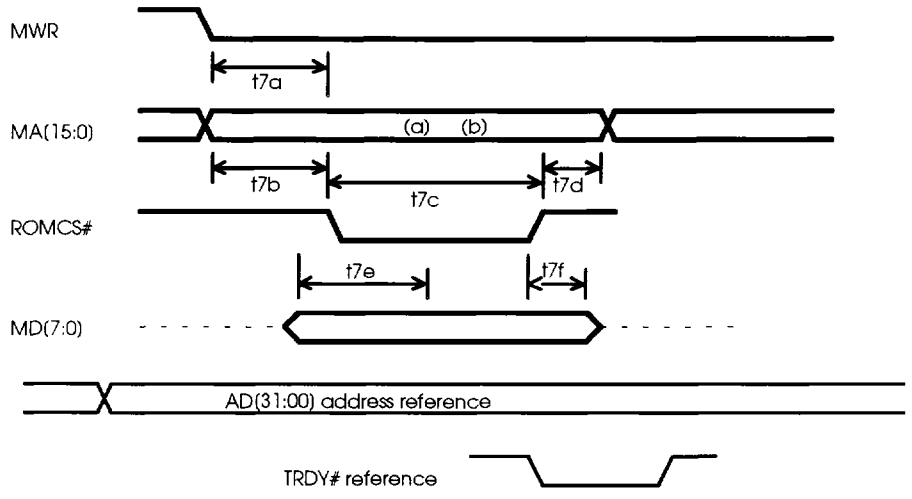
(a) = MA(15:2) = AD(15:2) address reference

note, MA15 is provided by use of MDP during ROM read and EEPROM write cycles.

Figure 9-14. PCI ROM/EEPROM Read Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t6a	MWR setup to ROMCS# fall	60		ns	9-14	
t6b	MA[15:0] setup to ROMCS# fall (byte 1)	60		ns	9-14	
t6c	MD[7:0] valid from ROMCS# fall		150	ns	9-14	
t6d	MD[7:0] hold from ROMCS# rise	0		ns	9-14	
t6e	MA[15:0] hold from ROMCS# rise	5		ns	9-14	
t6f	MA[15:0] setup to ROMCS# fall (bytes 2-4)	28		ns	9-14	
t6g	ROMCS# low time	180		ns	9-14	
t6h	ROMCS# high time (between bytes)	33		ns	9-14	
t6i	MWR hold from ROMCS# rise	30		ns	9-14	



(a) = MA(15:2) = AD(15:2) address reference

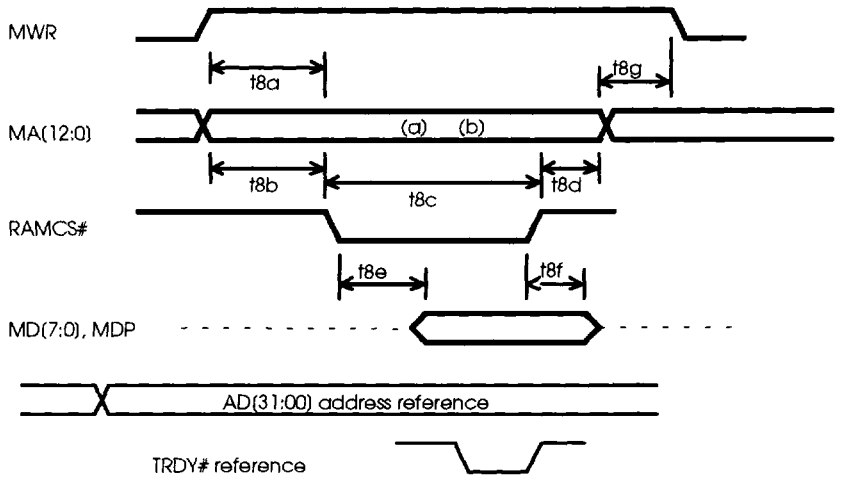
(b) = MA(1:0) = CBE(3:0) byte reference

Note, MA15 is provided by use of MDP during ROM read and EEPROM write cycles.

Figure 9-15. PCI EEPROM Write Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t7a	MWR setup to ROMCS# fall	60		ns	9-15	
t7b	MA[15:0] setup to ROMCS# fall	60		ns	9-15	
t7c	ROMCS# low time	150		ns	9-15	
t7d	MA[15:0] hold time from ROMCS# rise	60		ns	9-15	
t7e	MD[7:0] setup time to ROMCS# fall	30		ns	9-15	
t7f	MD[7:0] hold time from ROMCS# rise	30		ns	9-15	
t7g	MWR hold time from ROMCS# rise	30		ns	9-15	



(a) = MA(12:2) = AD(12:2) address reference

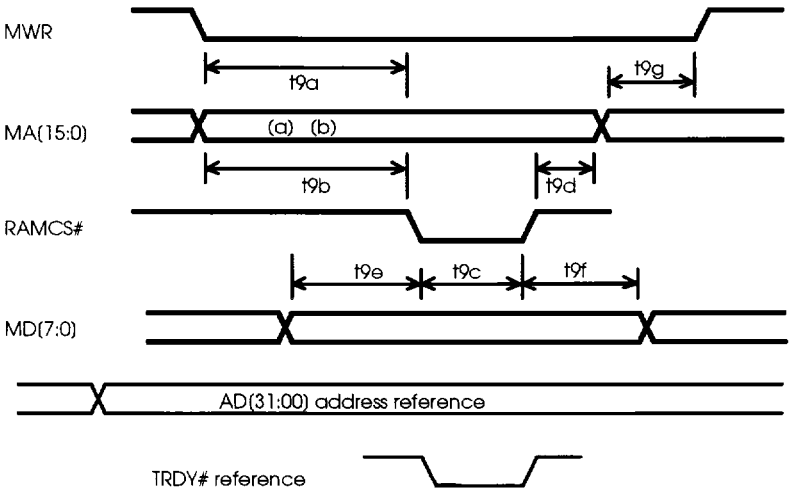
(b) = MA(1:0) = CBE(3:0) byte reference

Note, MA(14:13) not used.

Figure 9-16. PCI SRAM Read Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t8a	MWR setup to RAMCS# fall	30		ns	9-16	
t8b	MA[12:0] setup to RAMCS# fall	30		ns	9-16	
t8c	RAMCS# low time	90		ns	9-16	
t8d	MA[12:0] hold from RAMCS# rise	30		ns	9-16	
t8e	MD[7:0], MDP valid to RAMCS# fall	48		ns	9-16	
t8f	MD[7:0], MDP hold from RAMCS# rise	0		ns	9-16	
t8g	MWR hold from RAMCS# rise	30		ns	9-16	



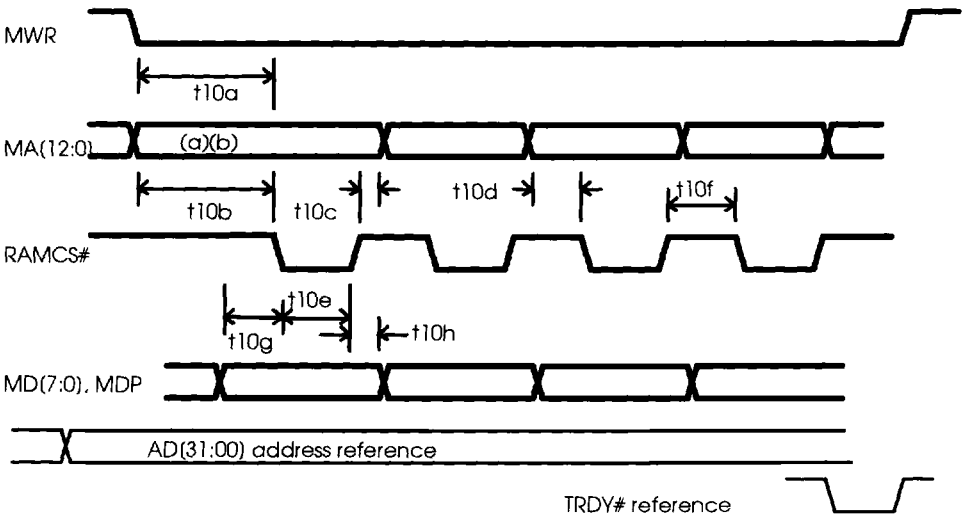
(a) = MA(15:2) = AD(15:2) address reference

(b) = MA(1:0) = CBE(3:0) byte reference

Figure 9-17. PCI SRAM Write Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t9a	MWR setup to RAMCS# fall	60		ns	9-17	
t9b	MA[12:0] setup to RAMCS# fall	60		ns	9-17	
t9c	RAMCS# low time	30		ns	9-17	
t9d	MA[12:0] hold from RAMCS# rise	30		ns	9-17	
t9e	MD[7:0], MDP setup to RAMCS# fall	30		ns	9-17	
t9f	MD[7:0], MDP hold from RAMCS# rise	30		ns	9-17	
t9g	MWR hold from RAMCS# rise	30		ns	9-17	



(a) = MA(12:2) = AD(12:2) address reference

(b) = MA(1:0) = CBE(3:0) byte reference
Note, MA(14:13) are not used.

Figure 9-18. PCI SRAM Double Word Write Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t10a	MWR setup to RAMCS# fall (byte 1)	30		ns	9-18	
t10b	MA[12:0] setup to RAMCS# fall (byte 1)	60		ns	9-18	
t10c	MA[12:0] hold from RAMCS# rise	12		ns	9-18	
t10d	MA[12:0] setup to RAMCS# fall (bytes 2-4)	21		ns	9-18	
t10e	RAMCS# low time	30		ns	9-18	
t10f	MD[7:0], MDP setup to RAMCS# fall (byte 1)	30		ns	9-18	
t10g	MD[7:0], MDP setup to RAMCS# fall (bytes 2-4)	21		ns	9-18	
t10h	MD[7:0], MDP hold RAMCS# rise	12		ns	9-18	
t10i	RAMCS# high time (bytes 2-3)	30		ns	9-18	
t10j	MWR hold time from RAMCS# rise (byte 4)	30		ns	9-18	

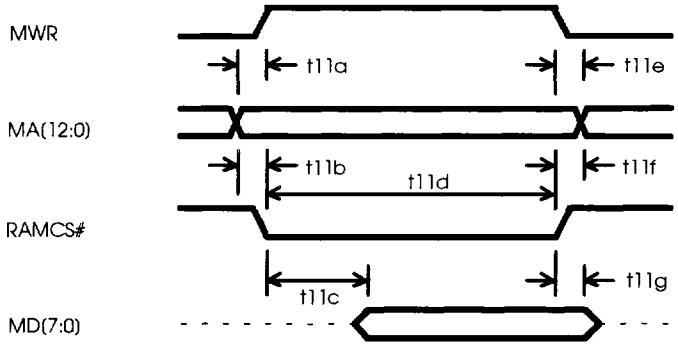


Figure 9-19. Sequencer Read Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t11a	MWR setup to RAMCS# fall	0		ns	9-19	
t11b	MA[12:0] setup to RAMCS# fall	0		ns	9-19	
t11c	MD[7:0], MDP valid from RAMCS# fall	20		ns	9-19	
t11d	RAMCS# low time	25		ns	9-19	
t11e	MWR hold from RAMCS# rise	0		ns	9-19	
t11f	MA[12:0] hold from RAMCS# rise	0		ns	9-19	
t11g	MD[7:0], MDP hold from RAMCS# rise	0		ns	9-19	

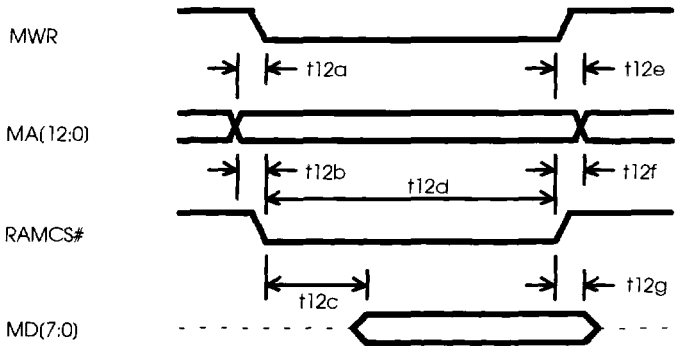


Figure 9-20. Sequencer Write Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t12a	MWR setup to RAMCS# fall	0		ns	9-20	
t12b	MA[12:0] setup to RAMCS# fall	0		ns	9-20	
t12c	MD[7:0], MDP valid from RAMCS# fall	20		ns	9-20	
t12d	RAMCS# low time	25		ns	9-20	
t12e	MWR hold from RAMCS# rise	0		ns	9-20	
t12f	MA[12:0] hold from RAMCS# rise	0		ns	9-20	
t12g	MD[7:0], MDP hold from RAMCS# rise	0		ns	9-20	

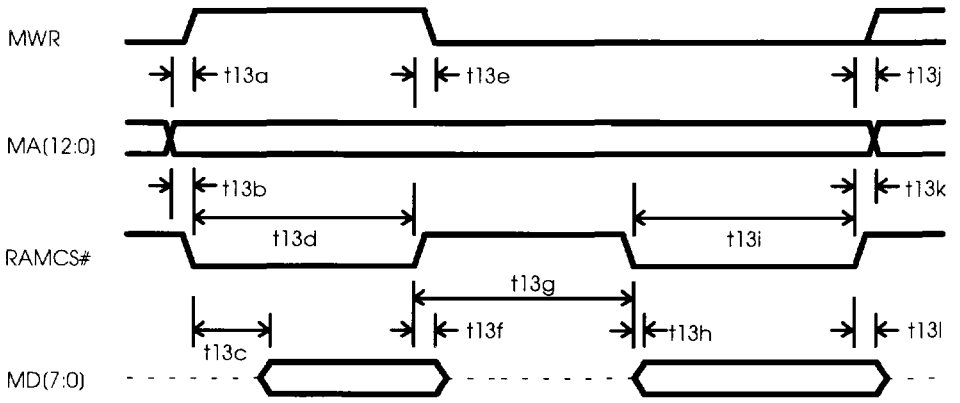


Figure 9-21. Sequencer Read-Modify-Write Memory Port Timing

Ta = 0 to 70 degrees C
VCC = 5 Volts +/- 5%
GND = 0 Volts

Symbol	Definition	Min	Max	Units	Figure	Notes
t13a	MWR setup to RAMCS# fall	0		ns	9-21	
t13b	MA[12:0] setup to RAMCS# fall	0		ns	9-21	
t13c	MD[7:0], MDP valid from RAMCS# fall	20		ns	9-21	
t13d	RAMCS# low time	25		ns	9-21	
t13e	MWR hold from RAMCS# rise	0		ns	9-21	
t13f	MD[7:0], MDP hold from RAMCS# rise	0	10	ns	9-21	
t13g	RAMCS# rise (read) to RAMCS# fall (write)	25		ns	9-21	
t13h	MD[7:0], MDP valid from RAMCS# fall	0	5	ns	9-21	
t13i	RAMCS# low time	25		ns	9-21	
t13j	MWR hold from RAMCS# rise	0		ns	9-21	
t13k	MA[12:0] hold from RAMCS# rise	0		ns	9-21	
t13l	MD[7:0], MDP hold from RAMCS# rise	0		ns	9-21	