1

Introduction

Overview

The AIC-7770 is a single chip, dual SCSI channel host adapter with Bus Master transfer capability which will attach to the ISA or EISA interface with no additional logic required. The AIC-7770 contains an advanced Sequencer which will execute a SCSI command described by a Sequencer Control Block (SCB). The AIC-7770 has within it an SCB Array which will hold up to four SCBs loaded by a driver. These SCBs will be executed independently of the SCSI Target ID or SCSI Channel 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 on board SCSI Sequencer will handle all phases of the SCSI bus, including the Disconnect/Reconnect and Command Complete message. On the host side, transfers will be made at the maximum rate available for the particular host. ISA mode will provide several transfer rates, and EISA will transfer up to the maximum burst rate of 33 MBytes/sec.

The device will also support new SCSI features and emerging host adapter features such as Fast/Wide/Differential SCSI data transfers, Tagged Queuing and Scatter/Gather.

Feature Summary

ISA Interface

- Direct connect to bus
- Eight I/O address ranges, 32 ISA locations each
- 16-bit Bus Master data transfer
- Video memory 8 or 16-bit transfers
- 24-bit address range
- Programmable data transfer rates 1 to 5 Megatransfers/second (2 to 10 Megabytes)
- Programmable bus on/bus off times for system performance tuning

EISA Interface

- Direct connect to bus
- 8-bit I/O transfers in 3 BCLKS vs. 6 when system responds to NOWS
- 32-bit burst transfers at 33 MBytes/sec
- 16-bit Downshift mode supported
- System translate cycles supported
- 1K address boundary detection in Burst mode
- 32-bit address range
- Programmable rising edge or negative level interrupts
- Programmable bus release times when preempted

Host Adapter Features

- Automatic data FIFO threshold selection
- Power-down mode
- Scatter/Gather supported
- Extremely low overhead
- Data residue reported on underruns
- One interrupt per command
- Queued commands per Target/LUN, overlapped execution
- Modify data pointers message handled
- Tagged Queuing supported

SCSI Features

- 2 SCSI busses
- Fast data transfers
- Wide data transfers, 1 SCSI bus

- Differential Controls for 1 SCSI bus
- Flexible configuration
  - Two 8-bit single ended, both Fast
  - One 8-bit single ended, and one 8-bit differential, both Fast
  - One 16-bit single ended or differential, Fast

Block Diagram

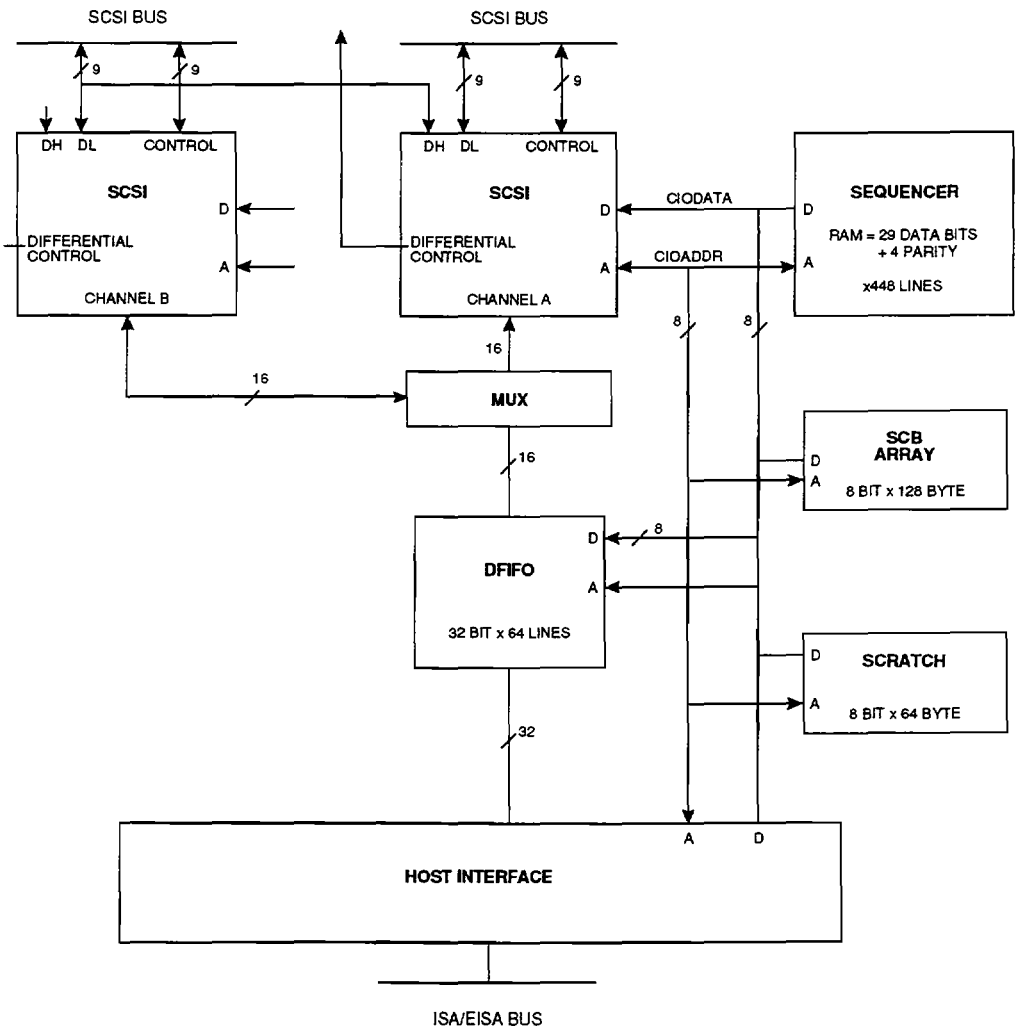


Figure 1-1. Block Diagram

2

Pin Description

Pin 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 the table below. The ISA and EISA pin definitions are mutually exclusive, only one group will be active at a time. A more complete description follows the summary.

Type Definition	Description
I	Input
3ST/#	Tristate output/Min drive current in mA
OC/#	Open collector output/Min drive current in mA
2ST/#	Two state output/Min drive current in mA

Host Interface

Common pin names are listed singly and shared pins are listed side by side. Pins with no counterpart on the other bus are listed in only one column.

Signal Name		Pin Number	I/O Driver ISA Master	I/O Driver ISA Slave	I/O Driver EISA Master	I/O Driver EISA Slave
ISA/EISA=1 ISA	ISA/EISA=0 EISA					
AEN		23	NA	I	NA	I
BCLK		24	I	I	I	I
ISA/EISA		7	I	I	I	I
D(15:0)		107, 106, 105, 104, 103, 102, 100, 99, 72, 73, 74, 75, 76, 77, 79, 80	I, 3ST/24	I, 3ST/24	I, 3ST/24	I, 3ST/24
IRQ		28	3ST/5	3ST/5	3ST/5	3ST/5
LA(23:17)		45, 46, 47, 48, 50, 51, 52	3ST/24	NA	3ST/24	NA
MASTER16-		25	OC/24	NA	OC/24	NA
NOWS-		15	NA	OC/24	NA	OC/24
RESDRV		11	I	I	I	I
	D(31:16)	98, 97, 96, 95, 93, 92, 91, 90, 89, 88, 87, 85, 84, 83, 82, 81	NA	NA	I, 3ST/24	NA
	LA(31:27)-	36, 37, 38, 39, 41	NA	NA	3ST/24	NA
CHRDY	EXRDY	13	I	NA	I	NA
DACK-	MAK-	27	I	NA	I	NA
DRQ	MREQ-	26	3ST/5	NA	3ST/5	NA
IORC-	START-	12	NA	I	3ST/24	I
IOWC-	CMD-	9	NA	I	I	I
M16-	BE(2)-	33	I	NA	3ST/24	I
MRDC-	MIO	21	3ST/24	NA	3ST/24	I
MWTC-	WR	22	3ST/24	NA	3ST/24	I
IOSEL1	EX32-	14	I	I	I	NA
IOSEL2	MSBURST-	19	I†	I†	3ST/24†	NA†
SBHE-	BE(3)-	32	3ST/24	NA	3ST/24	NA
SA(1:0)	BE(1:0)-	34, 35	3ST/24	I	3ST/24	I
SA(9:2)	LA(9:2)	62, 63, 64, 65, 66, 67, 68, 69	3ST/24	I	3ST/24	I
SA(15:10)	LA(15:10)	54, 55, 56, 58, 60, 61	3ST/24	I	3ST/24	NA*
SA(16)	LA(16)	53	3ST/24	NA	3ST/24	NA
SA(19:17)	LA(26:24)-	42, 43, 44	3ST/24	NA	3ST/24	NA
IOSEL0	SLBURST-	18	I	I	I	NA

*Input is not used in this case
†Input only in ISA mode, output only in EISA mode

SCSI Interface

Signal Name	Pin Number	I/O Driver
ASCD7-ASCD0	125, 126, 127, 128, 130, 131, 132, 133	I, OC/48
ASCDP	124	I, OC/48
ACD	115	I, OC/48
AIO	113	I, OC/48
AMSG	117	I, OC/48
AREQ	114	I, OC/48
AACK	120	I, OC/48
ARESET	119	I, OC/48
ASEL	116	I, OC/48
ABSY	121	I, OC/48
AATN	122	I, OC/48
ADIFFDAT(3:0)	139, 140, 141, 142	2ST/4
ADIFFADR(1:0)	136, 137	2ST/4
ADIFFSTRB	138	2ST/4
BSCD7-BSCD0	157, 158, 159, 160, 2, 3, 4, 5	I, OC/48
BSCDP	156	I, OC/48
BCD	147	I, OC/48
BIO	145	I, OC/48
BMSG	149	I, OC/48
BREQ	146	I, OC/48
BACK	152	I, OC/48
BRESET	151	I, OC/48
BSEL	148	I, OC/48
BBSY	153	I, OC/48
BATN	154	I, OC/48

Other Pins

Signal Name	Pin Number	I/O Driver
CLK	8	I
CVCC(1:0)	110, 17	NA
CGND(3:0)	143, 111, 59, 16	NA
SVCC0	135	NA
SAGND(4:0)	134, 129, 123, 118, 112	NA
SBGND(4:0)	6, 1, 155, 150, 144	NA
HVCC(5:0)	109, 94, 71, 57, 31, 20	NA
HGND(9:0)	108, 101, 86, 78, 70, 49, 40, 30, 29, 10	NA

Pin Description

ISA/AT Signal Definition

The signal names used in this specification correspond to the names in the EISA standard. A cross reference to original AT® bus signal names which differ is given here for convenience.

AT Names	ISA Names
RESET DRV	RESDRV
SD(15:0)	D(15:0)
I/O CH RDY	CHRDY
-IOR	IORC-
-IOW	IOWC-
-MEMR	MRDC-
-MEMW	MWRC-
SBHE	SBHE-
-MASTER	MASTER16-
-MEM CS16	M16-
0WS	NOWS-

ISA Interface

Signal Name	Description
AEN	Address Enable – AEN is driven by the system board and when de-asserted the device will respond as a slave to specific I/O addresses on the ISA bus. AEN will be de-asserted before, during and after IORC- or IOWC- is asserted for ISA. AEN is ignored when in Master mode.
BCLK	Bus Clock – BCLK provides the capability to synchronize device bus activities to the main system clock. The BCLK frequency operates between 8.333 MHz and 6MHz, with a normal duty cycle of 50 percent. BCLK is driven only by the system board. The BCLK period may be extended for synchronization to the system CPU or other system board devices with no adverse effect since events are synchronized to BCLK edges without regard to frequency or duty cycle. BCLK is driven by the system board. NOWS-, and DRQ are synchronized to the rising edge of BCLK.
ISA/EISA	ISA/EISA Select – ISA/EISA defines the pinout of the host interface section to be either ISA or EISA. When ISA/EISA is high the ISA interface is selected and when low the EISA interface is selected. This pin should be hard wired to VDD or GND.
CHRDY	Channel Ready – When operating as a Bus Master, this line is sampled two T cycles before the programmed rising edge of MWTC or MRDC. When CHRDY is detected de-asserted, the device will extend its assertion of MRDC- or MWTC- for as long as CHRDY is de-asserted plus three or less T cycles after sensing CHRDY asserted. CHRDY is also used as an input to enable writing to TESTCHIP.
DACK-	DMA Acknowledge – DACK- is asserted by the system board to acknowledge the grant of the ISA response to DRQ- asserted. The device asserts MASTER16- after detecting DACK- asserted to complete the ISA bus arbitration.
DRQ	DMA Request – DRQ is asserted by the device as an ISA Bus Master to arbitrate for the system bus. DRQ is deasserted/asserted synchronously with the rising edge of BCLK.
D(15:0)	Data – D15 is the most significant data bit of the data bus. All I/O commands are transferred on D(7:0). All data transferred as a Bus Master will be on D(15:0), except for leading and trailing byte offsets and when the address is between A0000 and BFFFF. When the address is between A0000 and BFFFF, if M16- is asserted, the transfer will be 16-bits wide and if de-asserted will be 8-bits wide.
IORC-	I/O Read Control – SA(15:0), AEN, and IOSEL(2:0) are decoded and if a valid address is detected
IOSEL(2:0)	I/O Address Select – The IOSEL(2:0) inputs provide a means to select the primary I/O address range that will respond to IORC- and IOWC-.
IOWC-	I/O Write Control – SA(15:0), AEN, and IOSEL(2:0) are decoded and if a valid address is detected, data from D(7:0) will be written into the selected register on the trailing edge.
IRQ	Interrupt Request – This is a high true signal which will become true when an interrupt event happens and that event has been enabled by the driver.
LA(23:17)	Latched Address – These high true signals are driven at the same time as the SA address lines.

Continued

Signal Name	Description
MASTER16-	Bus Master – This signal is asserted as part of the normal protocol when arbitrating for the ISA bus. It is asserted after the system board asserts DACK- to indicate a Bus Master now controls the ISA Bus. It will be de-asserted immediately after DACK- is de-asserted.
MRDC-	Memory Read Control – This signal is asserted as a Bus Master during memory reads. Data is latched on the trailing edge of the pulse, and the width and rate are programmable.
MWTC-	Memory Write Control – This signal is asserted as a Bus Master during memory writes. Data is valid before the leading edge of the pulse, and the width and rate are programmable.
M16-	Memory width 16-bit – This signal is asserted by a memory slave when the data is 16-bits wide. It is only considered valid for video RAM transfers and will take effect only when the address is between A0000 and BFFFF.
NOWS-	No Wait State – The device asserts NOWS- when responding as an I/O slave synchronous to the rising edge of BCLK. This action requests that the I/O cycle be shortened. The device may be programmed to shorten the cycle by 3 or 4 BCLKs by setting or clearing ENABLE (bit 0, BCTL).
RESDRV	Reset – Assertion of RESDRV causes a hardware reset to the AIC-7770. All bus drivers will be disabled until RESDRV is de-asserted. The AIC-7770 registers will be put in the initialized state, RAM is indeterminate. RESDRV is usually asserted at power up time.
SA(19:0)	System Address – As a Bus Master, the system address lines are driven with the appropriate address during memory reads and writes. SA (19:0) and LA(23:17) are driven with the total 24-bit address. As a slave, SA (15:0) are decoded to detect a valid I/O address.
SBHE-	System Byte High Enable – As a Bus Master, SBHE- is asserted to indicate that a 16-bit data transfer, or an 8-bit data transfer to an odd address is about to occur.

EISA Interface

Signal Name	Description
AEN	Address Enable – AEN is driven by the system and used by the I/O decode logic to validate the address on the LA lines. AEN will be de-asserted when START- is asserted for I/O accesses until after CMD- is asserted. AEN is ignored when in Master mode.
BCLK	Bus Clock – BCLK is used to synchronize the device bus activities to the main system clock. The BCLK frequency operates between 8.333 MHz and 8 MHz, with a normal duty cycle of 50 percent. BCLK is driven only by the system board. Events are synchronized to BCLK edges without regard to frequency or duty cycle. BCLK is always synchronous with the trailing edge of START- and the leading edge of CMD-. BCLK may not be synchronous with the leading edge of START- or the trailing edge of CMD-.
BE(3:0)-	Byte Enable – BE(3:0)- are the byte enable signals that identify the specific bytes addressed in a double word space. BE(3:0)- are pipelined from one cycle to the next and are inputs latched as an I/O slave and are outputs when a Bus Master. The timing of these signals varies depending on the cycle type. During normal cycles they are presented in reference to the rising edge of BCLK and for burst cycles they are presented in reference to the falling edge of BCLK and remain valid as long as LA(31:2) remain valid.
ISA/EISA	ISA/EISA Select – ISA/EISA selects either the ISA or EISA interface to be the active host interface. When ISA/EISA is asserted the ISA interface is selected and when de-asserted the EISA interface is selected. Changing this pin will redefine the I/O pins of the device. This pin should be hard wired to VDD or GND.
CMD-	Command – CMD- is driven by the system board and provides timing control within the EISA bus cycle. The system board asserts CMD- on the rising edge of BCLK, simultaneously with de-assertion of START-. The system board holds CMD- asserted until the end of a normal cycle (2 BCLKs) which may be extended by the system board when performing data size translations. When burst cycles are being performed the system board holds CMD- asserted until MSBURST- is de-asserted. The end of the cycle may be extended by EXRDY and is normally synchronized with the rising edge of BCLK, but in certain cases is asynchronous.
D(31:0)	Data – Data is 32-bits wide for normal or burst transfers. The Data bus is divided up into 4 byte lanes of 8-bits each. Valid data in a particular byte lane is indicated by one of the BE- lines being asserted. When responding as an I/O Slave, data is transferred on D(7:0) only. When transferring as a Bus Master, data will be transferred 32-bits at a time, or less if the transfer is less than 4 bytes long such as during leading or trailing partial double-word transfers. Data may be transferred with normal cycles or with burst cycles. If a 16-bit system memory is installed with the ability to transfer with burst cycles, the device will go into Downshift mode and transfer data on D(15:0).
EXRDY	EISA Transfer Ready – This signal is sampled when in Master mode to determine if wait-states should be added to the current cycle. It is sampled with BCLK and the cycle will be extended for as long as EXRDY is de-asserted. EXRDY is not driven as a slave.

Continued

Signal Name	Description
EX32-	EISA Transfer 32-bit – EX32- is asserted by a 32-bit EISA memory after decoding a valid address on signals LA(31:2) and MIO. EX32- is also asserted by the system board when it is performing data size translation to indicate the completion of the translation. EX32- is sampled by the device when transferring data as a Bus Master.
IRQ	Interrupt Request – IRQ may be selected to operate in either the edge triggered or level sensitive interrupt mode. IRQ is asserted when the INTEN bit is active and one of the interrupt conditions is true with its corresponding interrupt enabled. In the edge triggered mode, IRQ will drive high when active and drive low when inactive. In the level sensitive mode, IRQ will drive low when active and will remain turned off when inactive.
LA(23:2)	Latchable Address – These lines are driven along with LA(31:24)- and BE(3:0)- when in Bus Master mode. When responding as an I/O Slave, address lines LA(9:2) and BE(3:0)- are decoded with AEN and MIO for a valid address.
LA(31:24)-	Latchable Address – These lines have the same characteristics as LA(23:2) except they are active low.
MAK-	Master Acknowledge – MAK- asserted indicates to the device that it has been granted bus access as an EISA Bus Master. MAK- is driven by the system board in response to MREQ- asserted by the device.
MASTER16-	Master, 16-bit – MASTER16- is asserted as a Bus Master to indicate that it is able to enter the Downshift mode while transferring to or from a 16-bit EISA memory which has burst ability.
MIO	Memory, I/O – MIO is asserted by a EISA Bus Master or the system board to indicate a memory cycle is in progress. MIO de-asserted indicates that the cycle in progress is an I/O cycle.
MREQ-	Master Request – MREQ- is asserted by the device to request access to the bus as an Bus Master. The device will hold MREQ- asserted until the system board grants access by asserting MAK-. The system board samples MREQ- on the rising edge of BCLK. When the device has completed a transfer it can release the bus by de-asserting MREQ- on the falling edge of BCLK. The device will wait a minimum of two BCLK cycles between de-asserting MREQ- before again asserting MREQ-. Once MREQ- is asserted, it will not be released until at least one transfer takes place.
MSBURST-	Master Burst – MSBURST- is asserted by the device in Bus Master mode to indicate that burst cycles will be performed to the addressed memory slave if SLBURST- is asserted. The device begins a data transfer to or from memory with normal cycles (2 BCLKs/cycle) and will shift to burst cycles (1 BCLK/cycle) when it samples SLBURST- asserted with the rising edge of BCLK at the end of START-, as long as EXRDY is sampled asserted on the next falling edge of BCLK. 1K page boundary detection during a burst sequence will cause the burst to be interrupted with an normal cycle being performed when passing over the bounds, then the burst will be resumed until one of the following occurs (the Host byte count is zero, a Bus Master preemption occurred and the number of BCLKs determined by the BOFF value have passed or HDMAEN is set inactive).
NOWS-	No Wait State – The device asserts NOWS- when responding as a slave to shorten the standard 6 BCLK I/O cycle to 3 BCLKs.

Signal Name	Description
RESDRV	Reset – Assertion of RESDRV causes a hardware reset to the AIC-7770. All bus drivers will be disabled until RESDRV is de-asserted. The AIC-7770 registers will be put in the initialized state, RAM is indeterminate. RESDRV is usually asserted at power up time. In EISA mode, after RESDRV is de-asserted all bus drivers will remain inhibited until the ENABLE bit is set active (=1) in the BCTL register by an IO write cycle from the system board, except that data will be driven as requested by I/O Read cycles from the system board.
SLBURST-	Slave Burst – SLBURST- is driven by a memory slave to indicate it is able to perform burst transfers. SLBURST- is sampled by the device when transferring as a Bus Master. If it is asserted, the device will go into Burst mode.
START-	Start – START- is asserted by the device at the beginning of a cycle after LA(31:2) and MIO become valid. START- is de-asserted on the rising edge of BCLK after 1 BCLK time.
WR	Write/Read – The device asserts WR when in Master mode to indicate a memory Write. WR remains valid as long as LA(31:2) are valid. WR is driven from the same edge of BCLK that activates START-. As an I/O Slave, WR is sampled as an input and determines the direction of data transfer.

SCSI Interface

Signal Name	Description
ASCD(7:0)-	SCSI Data, port A – The SCSI data lines drive the ID during arbitration and selection, and command and data information as well as status and messages.
ASCDP -	SCSI Parity, port A – This bit provides odd parity for ASCD(7:0).
ACD-	Command/Data, port A – 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 de-asserted. This control signal is also used for 16-bit transfers.
AIO-	In/Out, port A – 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 de-asserted. This control signal is also used for 16-bit transfers.
AMSG-	Message, port A – 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 de-asserted. This control signal is also used for 16-bit transfers.
AREQ-	Request, port A – 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 also used for 16-bit transfers.

Continued

Signal Name	Description
AACK-	Acknowledge, port A – 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 or is needed by the Target. This control signal is also used for 16-bit transfers.
ARESET-	Reset, port A – This line is received and driven. It is interpreted as a Hard Reset and will clear all commands pending on the SCSI bus. This control signal is also used for 16-bit transfers.
ASEL-	Select, port A – 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 also used for 16-bit transfers.
ABSY-	Busy, port A – 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 also used for 16-bit transfers.
AATN-	Attention, port A – 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.
ADIFFDAT(3:0)-	Differential 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).
ADIFFADR(1:0)-	Differential Address – These lines determine the meaning of DIFFDAT(3:0).
ADIFFSTRB-	Differential Strobe – This signal clocks the data from DIFFDAT into the addressed latch specified by DIFFADR. Both DIFFDAT and DIFFADR are stable for the duration of DIFFSTRB.
BSCD0-SCD7-	SCSI Data, port B – The SCSI data lines drive the ID during arbitration and selection, and command and data information as well as status and messages.
BSCDP-	SCSI Parity, port B – This bit provides odd parity for ASCD(7:0).
BCD-	Command/Data, port B – 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 de-asserted. BCD- is grounded to indicate a 16-bit transfer hardware connection.
BIO-	In/Out, port B – 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 de-asserted.
BMSG-	Message, port B – 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 de-asserted. BMSG- is grounded to indicate that no hardware connection has been made to Channel B.
BREQ-	Request, port B – 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.
BACK-	Acknowledge, port B – 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 or is needed by the Target.
BRESET-	Reset, port B – This line is received and driven. It is interpreted as a Hard Reset and will clear all commands pending on the SCSI bus.

Signal Name	Description
BSEL-	Select, port B – This line is driven after a successful arbitration to Select as an Initiator or Reselect as a Target, and otherwise it is received.
BBSY-	Busy, port B – 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. BBSY- is tied to VDD when indicating a 16-bit hardware connection or no hardware connection. See BMSG- and BCD-.
BATN-	Attention, port B – This line is driven as an Initiator when a special condition occurs. It is received by the Target.

Other Pins

Signal Name	Description
CLK	Clock input. 40 MHz nominal input frequency.
CVCCx	Chip +Voltage supply
CGNDx	Chip Ground
SVCCx	SCSI A and B bus Driver +Voltage supply
SAGNDx	SCSI A bus Driver Ground
SBGNDx	SCSI B bus Driver Ground
HVCCx	Host bus Driver +Voltage supply
HGNDx	Host bus Driver Ground

Notes:

- Unused pins should not be connected
- The device should be decoupled using standard practice of locating capacitors as close as possible to the device. As the host and chip power connections are separated internal to the chip, care should be taken not to couple noise from HVCC pins to CVCC pins

❒

3

Register Description

ISA I/O Map

All I/O registers are decoded in groups of 32 registers so as not to use excessive ISA I/O space. The registers are decoded within 32 address spaces and the aliases of that space. The 32 address group may be moved to other locations, to provide flexibility for the system integrator. The register groups and the possible locations are summarized below.

IOSELn 210	SCSI Primary	Scratch Alias 1 and 2	Sequencer Alias 3	Host Alias 4	SCB Array Alias 5
000	0120-013F	0520-053F 0920-093F	0D20-0D3F	1120-123F	1520-153F
001	0140-015F	0540-055F 0940-095F	0D40-0D5F	1140-115F	1540-155F
010	0220-023F	0620-063F 0A20-0A3F	0E20-0E3F	1220-123F	1620-163F
011	0240-025F	0640-065F 0A40-0A5F	0E40-0E5F	1240-125F	1640-165F
100	0280-029F	0680-069F 0A80-0A9F	0E80-0E9F	1280-129F	1680-169F
101	02A0-02BF	06A0-06BF 0AA0-0ABF	0EA0-0EBF	12A0-12BF	16A0-16BF
110	0320-033F	0720-073F 0B20-0B3F	0F20-0F3F	1320-133F	1720-173F
111	0340-035F	0740-075F 0B40-0B5F	0F40-0F5F	1340-135F	1740-175F

EISA and Chip I/O Map

The EISA I/O map is contained entirely within the slot specific address range of zC00-zCFF where *z* is the slot number. The on-chip address may be obtained from the EISA address by using the lower eight bits of the EISA address. Address bits LA10 and LA11 are not decoded so that the chip will respond to addresses 000-0BF, 400-4BF, 800-8BF, and C00-CBF. This is to allow for multiple chips attached to a single EISA slot with some additional external decoding logic. The EISA ranges are summarized below.

Register Group	Address Range
SCSI	C00-C1F
SCRATCH	C20-C5F
SEQUENCER	C60-C7F
HOST	C80-C9F
SCB Array	CA0-CBF

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. The ISA range given is an example of one of the ranges offered for this group. The EISA and chip addresses uniquely specify the address. When the host must access these registers the Sequencer must be paused, except when noted otherwise.

SCSI Registers

ISA 0340-035F, EISA zC00-zC1F, CHIP 00-1F

Register Name	Mnemonic	Read/Write	Comments
SCSI Sequence control	SCSISEQ	R/W	
SCSI Transfer Control 0	SXFRCTL0	R/W	
SCSI Transfer Control 1	SXFRCTL1	R/W	
SCSI Signal Out	SCSISIGO	W	
SCSI Signal In	SCSISIGI	R	
SCSI Rate	SCSIRATE	R/W	
SCSI ID	SCSIID	R/W	
SCSI Latched Data Low	SCSIDATL	R/W	
SCSI Latched Data High	SCSIDATH	R/W	
SCSI Transfer Count 0	STCNT0	R/W	
SCSI Transfer Count 1	STCNT1	R/W	
SCSI Transfer Count 2	STCNT2	R/W	
Clear SCSI Interrupt 0	CLRSINT0	W	
SCSI Status 0	SSTAT0	R	
Clear SCSI Interrupt 1	CLRSINT1	W	
SCSI Status 1	SSTAT1	R	
SCSI Status 2	SSTAT2	R	
SCSI Status 3	SSTAT3	R	
SCSI Test	SCSITEST	R/W	
SCSI Interrupt Mode 0	SIMODE0	R/W	
SCSI Interrupt Mode 1	SIMODE1	R/W	
SCSI Data Bus Low	SCSIBUSL	R	
SCSI Data Bus High	SCSIBUSH	R	
SCSI Count Host Address 0	SHADDR0	R	Write address is same as HADDR0
SCSI Count Host Address 1	SHADDR1	R	Write address is same as HADDR1
SCSI Count Host Address 2	SHADDR2	R	Write address is same as HADDR2
SCSI Count Host Address 3	SHADDR3	R	Write address is same as HADDR3
Selection Time-out Timer	SELTIMER	R	
Selection/Reselection ID	SELID	R	
SCSI Block Control	SBLKCTL	R/W	

Scratch

ISA 0740-075F and 0B40-0B5F, EISA zC20-zC5F, CHIP 20-5F

The Scratch area is addressed directly by the Sequencer or Host driver. 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. There are 64 locations total.

Scratch RAM area	SCRATCH	R/W

Sequencer Registers

ISA 0F40-0F5F, EISA zC60-zC7F, CHIP 60-7F

Register Name	Mnemonic	Read/Write	Comments
Sequencer Control	SEQCTL	R/W	
Sequencer RAM	SEQRAM	R/W	
Sequencer Address Low	SEQADDRL	R/W	
Sequencer Address High	SEQADDRH	R/W	
Accumulator	ACCUM	R/W	
Source Index	SINDEX	R/W	
Destination Index	DINDEX	R/W	
Break Address Low	BRKADDRL	R/W	
Break Address High	BRKADDRH	R/W	
All Ones	ALLONES	R	
All Zeros	ALLZEROS	R	
No destination	NONE	W	
Flags	FLAGS	R	
Source Index Indirect	SINDIR	R	Not usable by host driver
Destination Index Indirect	DINDIR	W	Not usable by host driver
Function 1	FUNCT1	R/W	

Host Registers

ISA 1340-135F, EISA zC80-zC9F, CHIP 80-9F

Register Name	Mnemonic	Read/Write	Comments
Board ID 0	BID0	R	Host only, no pause
Board ID 1	BID1	R	Host only, no pause
Board ID 2	BID2	R	Host only, no pause
Board ID 3	BID3	R	Host only, no pause
Board Control	BCTL	R/W	Host only, no pause
Bus On/Off Time	BUSTIME	R/W	
Bus Speed	BUSSPD	R/W	
Host Control	HCNTRL	R/W	Host only, no pause
Host Address 0	HADDR0	R/W	
Host Address 1	HADDR1	R/W	
Host Address 2	HADDR2	R/W	
Host Address 3	HADDR3	R/W	
Host Byte Count 0	HCNT0	R/W	
Host Byte Count 1	HCNT1	R/W	
Host Byte Count 2	HCNT2	R/W	
SCB pointer	SCBPTR	R/W	
Interrupt Status	INTSTAT	R	Host only, no pause
Clear Interrupt	CLRINT	W	Host only, no pause
Hard Error	ERROR	R	Host only
Data Fifo Control	DFCNTRL	R/W	
Data Fifo Status	DFSTATUS	R	
Data Fifo Write Address	DFWADDR	R/W	
Data Fifo Read Address	DFRADDR	R/W	
Data Fifo Data	DFDAT	R/W	
SCB Auto Increment	SCBCNT	R/W	
Queue in FIFO (4 deep)	QINFIFO	R/W	
Queue out FIFO (4 deep)	QOUTFIFO	R/W	Read by Host only, no pause
Queue In Count	QINCNT	R/W	
Queue Out Count	QOUTCNT	R/W	Read by Host only, no pause
Test Chip	TESTCHIP	R/W	

SCB Array

ISA 1740-175F, EISA zCA0-zCBF, CHIP A0-BF

The SCB Array is addressed by the SCB pointer in the Host register block and the lower 5 bits of the register address if SCBAUTO (bit 7, SCBCNT) is cleared. If SCBAUTO is set, then SCBCNT(4:0) determines the address, and the SCB address is automatically incremented on an I/O Read or Write.

SCB Area (32 I/O spaces) SCB R/W

Bit Definition Summary

Addresses refer to ISA default values and internal chip addresses. EISA addresses are created by adding a prefix to the chip address. All unused and reserved register bits read as 0, and should be written with 0. All unassigned register locations return 0. *Reserved* denotes anticipated usage. Unused bits are available for future definition.

SCSI Registers

SCSISEQ	SXFRCTL0	SXFRCTL1	SCSISIGI	SCSISIGO	SCSIRATE
R/W I-0340 C-00	R/W I-0341 C-01	R/W I-0342 C-02	R I-0343 C-03	W I-0343 C-03	R/W I-0344 C-04
7 TEMODEO	7	7 BITBUCKET	7 CDI	7 CDO	7 WIDEXFER
6 ENSELO	6	6 SWRAPEN	6 IOI	6 IOO	6 SXFR(2)
5 ENSELI	5	5 ENSPCHK	5 MSGI	5 MSGO	5 SXFR(1)
4 ENRSELI	4 CLRSTCNT	4 STIMESEL(1)	4 ATNI	4 ATNO	4 SXFR(0)
3 ENAUTOATNO	3 SPIOEN	3 STIMESEL(0)	3 SELI	3 SELO	3 SOFS(3)
2 ENAUTOATNI	2	2 ENSTIMER	2 BSYI	2 BSYO	2 SOFS(2)
1 ENAUTOATNP	1 CLRCHN	1	1 REQI	1 REQO	1 SOFS(1)
0 SCSIRSTO	0	0	0 ACKI	0 ACKO	0 SOFS(0)

SCSIID	SCSIDATL	SCSIDATH	STCNT0	STCNT1	STCNT2
R/W I-0345 C-05	R/W I-0346 C-06	R/W I-0347 C-07	R/W I-0348 C-08	R/W I-0349 C-09	R/W I-034A C-0A
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	SSTAT0	CLRSINT1	SSTAT1	SSTAT2	SSTAT3
W I-034B C-0B	R I-034B C-0B	W I-034C C-0C	R I-034C C-0C	R I-034D C-0D	R I-034E C-0E
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	SIMODE0	SIMODE1	SCSIBUSL	SCSIBUSH	SHADDR0
R/W I-034F C-0F	R/W I-0350 C-10	R/W I-0351 C-11	R I-0352 C-12	R I-0353 C-13	R I-0354 C-14
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	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	SHADDR2	SHADDR3	SELTIMER	SELID	SBLKCTL
R I-0355 C-15	R I-0356 C-16	R I-0357 C-17	R I-0358 C-18	R I-0359 C-19	R/W I-035F C-1F
7 SHADDR(15)	7 SHADDR(23)	7 SHADDR(31)	7	7 SELID(3)	7
6 SHADDR(14)	6 SHADDR(22)	6 SHADDR(30)	6	6 SELID(2)	6
5 SHADDR(13)	5 SHADDR(21)	5 SHADDR(29)	5 STAGE 6	5 SELID(1)	5
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
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

Scratch RAM Area

The scratch RAM area appears at addresses 0740-075F and 0B40-0B5F on the ISA bus, and C20-C5F on the EISA bus. The chip address is 20-5F. The scratch RAM data is user defined and depends on the application of the chip.

Sequencer Registers

SEQCTL	SEQRAM	SEQADDR0	SEQADDR1	ACCUM	SINDEX
R/W I-0F40 C-60	R/W I-0F41 C-61	R/W I-0F42 C-62	R/W I-0F43 C-63	R/W I-0F44 C-64	R/W I-0F45 C-65
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	BRKADDR0	BRKADDR1	ALLONES	ALLZEROS	NONE
R/W I-OF46 C-66	R/W I-0F47 C-67	R/W I-0F48 C-68	R I-0F49 C-69	R I-0F4A C-6A	W I-0F4A C-6A
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	SINDIR	DINDIR	FUNCTION1	FUNCTION1	STACK
R I-0F4B C-6B	R C-6C	W C-6D	W I-0F4E C-6E	R I-0F4E C-6E	R I-0F4F C-6F
7	CONTENTS	CONTENTS	7 RSVD	1 OF 8 DECODED	7 STACK(07)
6	POINTED	POINTED	6 FUN1DAT(2)	VALUE OF	6 STACK(06)
5	TO BY SINDEX	TO BY DINDEX	5 FUN1DAT(1)	FUN1DAT2-0	5 STACK(05)
4			4 FUN1DAT(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 Registers

BID0	BID1	BID2	BID3	BCTL	BUSTIME
R I-1340 C-80	R I-1341 C-81	R I-1342 C-82	R I-1343 C-83	R/W I-1344 C-84	R/W I-1345 C-85
7 "0"	7 "1"	7 "0"	7 "0"	7 RSVD	7 BOFF(3)
6 "0"	6 "0"	6 "1"	6 "1"	6 RSVD	6 BOFF(2)
5 "0"	5 "0"	5 "1"	5 "1"	5 RSVD	5 BOFF(1)
4 "0"	4 "1"	4 "1"	4 "1"	4 RSVD	4 BOFF(0)
3 "0"	3 "0"	3 "0"	3 "0"	3 ACE	3 BON(3)
2 "1"	2 "0"	2 "1"	2 "0"	2 RSVD	2 BON(2)
1 "0"	1 "0"	1 "1"	1 "0"	1 RSVD	1 BON(1)
0 "0"	0 "0"	0 "1"	0 "0"	0 ENABLE	0 BON(0)

BUSSPD	HCNTRL	HADDR0	HADDR1	HADDR2	HADDR3
R/W I-1346 C-86	R/W I-1347 C-87	R/W I-1348 C-88	R/W I-1349 C-89	R/W I-134A C-8A	R/W I-134B C-8B
7 DFTHRSH1	7	7 HADDR(07)	7 HADDR(15)	7 HADDR(23)	7 HADDR(31)
6 DFTHRSH0	6 POWRDN	6 HADDR(06)	6 HADDR(14)	6 HADDR(22)	6 HADDR(30)
5 STBOFF2	5	5 HADDR(05)	5 HADDR(13)	5 HADDR(21)	5 HADDR(29)
4 STBOFF1	4 SWINT	4 HADDR(04)	4 HADDR(12)	4 HADDR(20)	4 HADDR(28)
3 STBOFF0	3 IRQMS	3 HADDR(03)	3 HADDR(11)	3 HADDR(19)	3 HADDR(27)
2 STBON2	2 PAUSE[ACK]	2 HADDR(02)	2 HADDR(10)	2 HADDR(18)	2 HADDR(26)
1 STBON1	1 INTEN	1 HADDR(01)	1 HADDR(09)	1 HADDR(17)	1 HADDR(25)
0 STBON0	0 CHIPRST[ACK]	0 HADDR(00)	0 HADDR(08)	0 HADDR(16)	0 HADDR(24)

HCNT0	HCNT1	HCNT2	RESERVED	SCBPTR	INTSTAT
R/W I-134C C-8C	R/W I-134D C-8D	R/W I-134E C-8E	R/W I-134F C-8F	R/W I-1350 C-90	R/W I-1351 C-91
7 HCNT(07)	7 HCNT(15)	7 HCNT(23)	7 RSVD	7 RSVD	7 INTCODE(3)
6 HCNT(06)	6 HCNT(14)	6 HCNT(22)	6 RSVD	6 RSVD	6 INTCODE(2)
5 HCNT(05)	5 HCNT(13)	5 HCNT(21)	5 RSVD	5 RSVD	5 INTCODE(1)
4 HCNT(04)	4 HCNT(12)	4 HCNT(20)	4 RSVD	4 RSVD	4 INTCODE(0)
3 HCNT(03)	3 HCNT(11)	3 HCNT(19)	3 RSVD	3 RSVD	3 BRKADRINT
2 HCNT(02)	2 HCNT(10)	2 HCNT(18)	2 RSVD	2 SCBVAL(2)	2 SCSIINT
1 HCNT(01)	1 HCNT(09)	1 HCNT(17)	1 RSVD	1 SCBVAL(1)	1 CMDCMPLT
0 HCNT(00)	0 HCNT(08)	0 HCNT(16)	0 RSVD	0 SCBVAL(0)	0 SEQINT

CLRINT	ERROR	DFCNTRL	DFSTATUS	DFWADDR0	RESERVED
W I-1352 C-92	R I-1352 C-92	R/W I-1353 C-93	R I-1354 C-94	R/W I-1355 C-95	R/W I-1356 C-96
7	7	7	7	7 TESTBIT(R)	7 RSVD
6	6	6 WIDEODD	6	6 DFWADDR(6)	6 RSVD
5	5	5 SCSIEN[ACK]	5 DWORDEMP	5 DFWADDR(5)	5 RSVD
4	4	4 SDMAEN[ACK]	4 MREQPEND	4 DFWADDR(4)	4 RSVD
3 CLRBRKADRINT	3 PARERR	3 HDMAEN[ACK]	3 HDONE	3 DFWADDR(3)	3 RSVD
2	2 ILLOPCODE	2 DIRECTION[ACK]	2 DFTHRSH	2 DFWADDR(2)	2 RSVD
1 CLRCMDINT	1 ILLSADDR	1 FIFOFLUSH[ACK]	1 FIFOFULL	1 DFWADDR(1)	1 RSVD
0 CLRSEQINT	0 ILLHADDR	0 FIFORESET	0 FIFOEMP	0 DFWADDR(0)	0 RSVD

DFRADDR0	RESERVED	DFDAT	SCBCNT	QINFIFO	QINCNT
R/W I-1357 C-97	R/W I-1358 C-98	R/W I-1359 C-99	R/W I-135A C-9A	R/W I-135B C-9B	R I-135C C- 9C
7 TESTBIT(R)	7 RSVD	7 FDAT(7)	7 SCBAUTO	7 RSVD	7 RSVD
6 DFRADDR(6)	6 RSVD	6 FDAT(6)	6 RSVD	6 RSVD	6 RSVD
5 DFRADDR(5)	5 RSVD	5 FDAT(5)	5 RSVD	5 RSVD	5 RSVD
4 DFRADDR(4)	4 RSVD	4 FDAT(4)	4 SCBCNT(4)	4 RSVD	4 RSVD
3 DFRADDR(3)	3 RSVD	3 FDAT(3)	3 SCBCNT(3)	3 RSVD	3 RSVD
2 DFRADDR(2)	2 RSVD	2 FDAT(2)	2 SCBCNT(2)	2 RSVD	2 QINCNT(2)
1 DFRADDR(1)	1 RSVD	1 FDAT(1)	1 SCBCNT(1)	1 QIN(1)	1 QINCNT(1)
0 DFRADDR(0)	0 RSVD	0 FDAT(0)	0 SCBCNT(0)	0 QIN(O)	0 QINCNT(0)

QOUTFIFO	QOUTCNT	TESTCHIP
R/W I-135D C-9D	R I-135E C-9E	R/W I-135F C-9F
7 RSVD	7 RSVD	7 TESTSEL2
6 RSVD	6 RSVD	6 TESTSEL1
5 RSVD	5 RSVD	5 TESTSEL0
4 RSVD	4 RSVD	4 TESTRAM
3 RSVD	3 RSVD	3 TESTHOST
2 RSVD	2 QOUTCNT(2)	2 TESTSEQ
1 QOUT(1)	1 QOUTCNT(1)	1 TESTFIFO
0 QOUT(0)	0 QOUTCNT(0)	0 TESTSCSI

SCB Array

The SCB Array appears at addresses 1740-175F on the ISA bus, zCA0-CBF on the EISA bus, and A0-BF on the chip. The bits in this area are user defined, and depend on the application.

Register Definition

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
- **I-xxxxh:** ISA I/O address decode
- **E-zxxxh:** EISA I/O address decode
- **C-xxh:** Device internal address decode

SCSI Sequence Control (SCSISEQ)

Type: **R/W**
Address: **I-0340h, E-zC00h, C-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

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 a 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 timeout is enabled (bit 2, SXFRCTL1 is set to one), or for the software selection timeout if the hardware enable timeout 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 a 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 a 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 a 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 a 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 ASCD(7:0)- or BSCD(7:0)-. 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 a one SCSI RST 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: I-0341h, E-zC01h, C-01h

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

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

7-5	(0)	Not Used	Always reads 0.
4	(0)	CLRSTCNT	When set to a one both the SCSI Transfer Counter (STCNT) and the SCSI 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	When set to a 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.
2	(0)	Not Used	Always reads 0.

1	(0)	CLRCHN	When set to a one, the SCSI FIFO (SFIFO), 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: I-0342h, E-zC02h, C-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	

7	(0)	BITBUCKET	SCSI Bit Bucket Mode. When this bit is set to a 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	When this bit is set to one the STCNT register is allowed to wrap past 0 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	When set to a 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]	These bits define the Selection Timeout time used by the hardware selection timer. The Selection Timeout timer may be monitored via SELTIMER (bits 5-0). 4 3 - Bit 0 1 - 128 ms 1 1 - 32 ms 0 0 - 256 ms 1 0 - 64 ms
2	(0)	ENSTIMER	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	This bit defines Active Negation. When active (=1), this bit allows the SCSI outputs to perform active negation. When enabled, the SCSI output switches from asserted (=0) to de-asserted condition (=1) to improve the SCSI bus signal rising transition time.
0	(0)	Not Used	Always reads 0.

SCSI Control Signal Read Register (SCSISIGI)

Type: **R**
Address: **I-0343h, E-zC03h, C-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

7	(x)	CDI	Reads the state of the C/D signal on the SCSI bus.
6	(x)	IOI	Reads the state of the I/O 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: W
Address: I-0343h, E-zC03h, C-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 or Initiator) 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

7	(0)	CDO	If in Target mode, sets C/D on SCSI bus. If Initiator mode, sets the state of C/D expected on the next REQ pulse.
6	(0)	IOO	If in Target mode, sets I/O on SCSI bus. If Initiator mode, sets the state of I/O expected on the next REQ pulse.
5	(0)	MSGO	If in Target mode sets MSG on SCSI bus. If Initiator mode, sets the state of MSG expected on the next REQ pulse.
4	(0)	ATNO	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	When set to a one asserts SEL on the SCSI bus. Can be used to negate SEL.
2	(0)	BSYO	When set to a one asserts BSY on the SCSI bus. May also be used to negate BSY.
1	(0)	REQO	If in Target mode, sets REQ on the SCSI bus. It is not functional in Initiator mode.
0	(0)	ACKO	If in Initiator mode, sets ACK on the SCSI bus. It is not functional in Target mode.

SCSI Rate Control (SCSIRATE)

Type: R/W
Address: I-0344h, E-zC04h, C-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)

7 (0) WIDEXFER — When SELWIDE (bit 1, SBLKCTL) is set, and this bit is set, 16-bit transfers will take place on the SCSI bus. When SELWIDE is cleared, this bit is ignored.

6-4 (0) SXFR(2:0) — Synchronous SCSI Transfer Rate 2:0. These bits select the Data phase transfer rate per the below table. Times are shown for a 40 MHz clock with CLK 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 transfer rates below 3.6 MB/S, use Asynchronous Transfer mode.

3-0 (0) SOFS(3:0) SCSI Offset. 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 0001-1111 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.

If loaded with an incorrect value when doing Wide transfers, in Initiator mode the REQ overrun condition may not be detected, and in Target mode, an incorrect number of REQs will be sent out.

SCSI ID (SCSIID)

Type: **R/W**
Address: **I-0345h, E-zC05h, C-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)

7-4 (0) TID(3:0) 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: I-0346/0347h, E-zC06/zC07h, C-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: I-0348/0349/034Ah, E-zC08/zC09/zC0Ah, C-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 DFIFO. 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 Register 0 (CLRSINT0)

Type: W
Address: I-034Bh, E-zC0Bh, C-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	

7	(0)	Not Used	Always reads 0.
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	Always reads 0.
1	(0)	CLRSPIORDY	Clears SPIORDY interrupt and status
0	(0)	Not Used	Always reads 0.

SCSI Status Register 0 (SSTAT0)

Type: R
Address: I-034Bh, E-zC0Bh, C-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

7	(0)	TARGET	When this bit is a 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	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.
5	(0)	SELDI	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, SIMODE0) to one.

4	(0)	SELINGO	After successful arbitration, this bit is set to a one when you start the attempt to select or reselect another device. This interrupt is used to start looking for SELDO or Bus Timeout. 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	This bit is a one when STCNT counts from 000000h to FFFFFFh and SWRAPEN is set. SWRAPEN (bit 6, SXFRCTL1) 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	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 non-zero count to STCNT, setting SWRAPEN, 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	When this bit is a 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, the 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	This bit is the logical AND of HDONE (bit 3, DFSTATUS) and SDONE. It indicates the current transfer has completely finished.

Clear SCSI Interrupt 1 (CLRSINT1)

Type: W
Address: I-034Ch, E-zC0Ch, C-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

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	Always reads 0.
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: I-034Ch, E-zC0Ch, C-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

7 (0) SELTO — This bit is set when the hardware selection timer is enabled and a Selection or Reselection Timeout occurs. The timer is enabled by setting ENSTIMER (bit 2, SXFRCTL1) to one along with the timeout value in bits 3 and 4. The bit is cleared by setting CLRSELTIMO (bit 7, CLRSINT1) to one.

6 (0) ATNTARG — This bit is set to a one when you are a Target and the Initiator asserts ATN. It is latched and will be cleared when ATN is de-asserted or when CLRATNO (bit 6, CLRSINT1) is set.

5 (0) SCSIRSTI — This bit is set to a 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 — Initiator Mode only. This bit is set to a one when the last phase on the SCSI bus sampled by REQ does not match 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.

3 (0) BUSFREE — This bit is set to a 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	This bit is set to a 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	This bit is set to a 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	Initiator Mode only. This bit is set to a 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: I-034Dh, E-zC0Dh, C-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)

7	(0)	OVERRUN	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: I-034Eh, E-zC0Eh, C-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)

7-4	(0)	SCSICNT(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)	Gives the current SCSI offset count. Do not read this counter unless transfers are stopped.

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.

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

7-3	(0)	Not Used	Always reads 0.
2	(0)	RQAKCNT	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	When set to a one the SCSI transfer counter STCNT and the Selection timeout 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.

SCSI Interrupt Mode 0 (SIMODE0)

Type: R/W
Address: I-0350h, E-zC10h, C-10h

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

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

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: I-0351h, E-zC11h, C-11h

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

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

7	(0)	ENSELTIMO	Enables the SELTO status to 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 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: I-0352/0353h, E-zC12/zC13h, C-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/Host Address (SHADDR(0:3))

Type: R
Address: I-0354/0355/0356/0357h, E-zC14/zC15/zC16/zC17h, C-14/15/16/17h

These registers are written when the Host Address registers (HADDR) are written. 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 Timeout Timer (SELTIMER)

Type: R
Address: I-0358h, E-zC18h, C-18h

This register is used to monitor the state of the hardware Selection Timeout timer.

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

7-6	(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: **I-0359h, E-zC19h, C-19h**

This register contains the SCSI ID of the selecting or reselecting device which was asserted during the last (Re)Selection SCSI bus phase. Hardware will remove the device ID and decode the remaining ID. After a (Re)Selection In has taken place, the ID may be read from this register to determine the ID of the device which initiated the (Re)Selection. 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	

7-4	(0)	SELID(3:0)	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 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: I-035Fh, E-zC1Fh, C-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	
6	
5	AUTOFLUSHDIS
4	
3	SELBUSB
2	
1	SELWIDE
0	

7, 6, 4	(0)	Not Used	Always reads 0.
5	(0)	AUTOFLUSHDIS	This bit defines Auto Flush. When active this bit prevents the SCSI hardware activated autoflush action of the DFIFO to the PCI bus system memory and is due to SCSI PHASEMIS or SDONE becoming active. FIFOFLUSH in the DFCNTRL register may then be used to activate the flush action as needed.
3	(*)	SELBUSB	When this bit is set, SCSI Channel B is selected. Device addresses 00h-1Eh reflect the Channel B registers. When this bit is cleared, addresses 00h-1Eh reflect Channel A registers. If SELWIDE (bit 1) is set, this bit will be cleared.
2, 0	(0)	Not Used	Always reads 0.
1	(*)	SELWIDE	When this bit is set, the internals of the device are configured for one 16-bit Wide SCSI Channel. The SCSI data lines of Channel B are gated to the upper lines of Channel A, and the control lines of Channel A are used for phase detection and data transfer. It is expected that the external bus is Wide. When this bit is cleared, the device is configured for two 8-bit channels.

Note

SELBUSB and SELWIDE may be initialized on Chip Reset to indicate the hardware connection to the chip by connecting BMSG, BCD, AND BBSY to VDD and GND in various combinations. BMSG-=GND and BBSY-=VDD indicates that Channel B is not being used and will force SELBUSB and SELWIDE to be cleared. BCD-=GND and BBSY-=VDD indicates a wide connection and will clear SELBUSB and set SELWIDE. The following table summarizes the reset values.

Hardware Configuration	BBSY-	BCD-	BMSG-	SBLKCTL (hex)
Dual Busses	VDD	VDD	VDD	08
Channel A only	VDD	VDD	GND	00
Wide	VDD	GND	Don't care	02
Dual Busses	GND	Don't care	Don't care	08

Sequencer Control (SEQCTL)

Type: R/W
Address: I-0F40h, E-zC60h, C-60h

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

7 (1) PERRORDIS — When cleared, allows Sequencer RAM Parity Errors to be detected. When set, disables Parity Error detection. Sequencer RAM Parity Error detection should be disabled while loading the RAM in order to prevent false errors.

6 (0) PAUSEDIS — 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, a Sequencer RAM Parity Error interrupt, and an Illegal Address interrupt, reset this bit. Host software may not write to this bit.

5 (0) FAILDIS — If set, disables the Illegal Opcode or Address interrupt feature. If cleared, an Illegal Opcode or Address will cause a BRKADRINT to occur and will pause the Sequencer.

4 (0) FASTMODE — If set to one, then the clock to the Sequencer is divided by 4 from the input CLK. If set to zero, then the clock is divided by 5.

3 (0) BRKADRINTEN — 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 IRQ will not be asserted.

2	(0)	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	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	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: I-0F41h, E-zC61h, C-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 Address (SEQADDR(1:0))

Type: R/W
Address: I-0F42/0F43h, E-zC62/zC63h, C-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. See the section on loading the Sequencer under Functional Description.

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: I-0F44h, E-zC64h, C-64h

This register is a temporary holding place for the results of 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 (SINDEX)

Type: R/W
Address: I-0F45h, E-zC65h, C-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 (DINDEX)

Type: R/W
Address: I-0F46h, E-zC66h, C-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 Low (BRKADDR0)

Type: R/W
Address: I-0F47h, E-zC67h, C-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. See the Breakpoint section under Functional Description.

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 High (BRKADDR1)

Type: R/W
Address: I-0F48h, E-zC68h, C-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 the Breakpoint section under Functional Description.

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

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.
5-1	(0)	RSVD	Always reads 0.
0	(0)	BRKADDR(08)	Address bit 08 used for comparison with BRKADDR0.

All Ones (ALLONES)

Type: R
Address: I-0F49h, E-zC69h, C-69h

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

All Zeros (ALLZEROS)

Type: R
Address: I-0F4Ah, E-zC6Ah, C-6Ah

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

No Destination (NONE)

Type: **W**
Address: **I-0F4Ah, E-zC6Ah, C-6Ah**

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

Flags (FLAGS)

Type: **R**
Address: **I-0F4Bh, E-zC6Bh, C-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: **C-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: C-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: I-0F4Eh, E-zC6Eh, C-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: I-0F4Fh, E-zC6Fh, C-6Fh

The contents of the stack are 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)

Board ID 0 (BID0)

Type: R
Address: I-1340h, E-zC80h, C-80h

This is the most significant ID of the EISA Product Identifier. This register is hard wired to a value of 04h.

	BID0 R/W
7	"0"
6	"0"
5	"0"
4	"0"
3	"0"
2	"1"
1	"0"
0	"0"

Board ID 1 (BID1)

Type: R
Address: I-1341h, E-zC81h, C-81h

This is the next to the most significant ID of the EISA Product Identifier. This register is hard wired to a value of 90h.

	BID1 R/W
7	"1"
6	"0"
5	"0"
4	"1"
3	"0"
2	"0"
1	"0"
0	"0"

Board ID 2 (BID2)

Type: R
Address: I-1342h, E-zC82h, C-82h

This is the next to the least significant ID of the EISA Product Identifier. This register is hard wired to a value of 77h.

	BID2 R/W
7	"0"
6	"1"
5	"1"
4	"1"
3	"0"
2	"1"
1	"1"
0	"1"

Board ID 3 (BID3)

Type: R
Address: I-1343h, E-zC83h, C-83h

This is the least significant ID of the EISA Product Identifier. This register is hard wired to a value of 70h.

	BID3 R/W
7	"0"
6	"1"
5	"1"
4	"1"
3	"0"
2	"0"
1	"0"
0	"0"

Board Control (BCTL)

Type: R/W
Address: I-1344h, E-zC84h, C-84h

	BCTL R/W
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	ACE
2	RSVD
1	RSVD
0	ENABLE

7-4	(0)	Reserved	Always reads 0.
3	(0)	ACE	This is a programmable bit which may be used to indicate an external ROM and intercept logic is available to support other processors. (See *Advanced RISC Computing Standard Specification* and *Addendum* for EISA based systems).
2, 1	(0)	Reserved	Always reads 0.
0	(0)	ENABLE	Enable Board. When set, and using the EISA interface, the board is enabled for normal operation. When cleared, the board is disabled and will not drive any bus signals as a Master, but will respond as an I/O Slave. This signal is cleared by RESDRV or CHIPRESET (bit 0, HCNTRL). When in EISA mode it may be programmed to a 1 or 0 by the software driver. In ISA mode the interface is always enabled and this bit is used to determine the number of BCLKS that NOWS will be active. If a 0 is written to this bit, then NOWS will be active for 4 BCLKs, and if 1 is written, then NOWS will be active for 3 BCLKs.

Bus On/Off Time (BUSTIME)

Type: R/W
Address: I-1345h, E-zC85h, C-85h

This register governs the bus-on and off time during System Memory transfers.

In ISA mode, the bus-on time is limited to 15 us in increments of 1 us. A value of zero will cause the device to stay on the bus for 2 BCLKS minimum, and will always complete at least one transfer. The bus-off time is limited to 60 us in increments of 4 us. A value of zero will cause the device to stay off the bus for 2 BCLKS. The times indicated are derived from a 40 MHz input clock except where noted. Bus-on time also adds any remaining time that the read or write strobe is asserted, CHRDY extensions, the minimum strobe off to DREQ time, and any additional time required to synchronize to BCLK.

In EISA mode the BOFF value is also used to determine the number of BCLKs to continue transfers after being preempted by the system.

BUSTIME	R/W
7	BOFF(3)
6	BOFF(2)
5	BOFF(1)
4	BOFF(0)
3	BON(3)
2	BON(2)
1	BON(1)
0	BON(0)

7-4 (0) BOFF(3:0) The meaning of this register changes depending on whether ISA or EISA mode is selected. In ISA mode, this value gives the minimum time that the device will stay off the bus during data transfers, except that a value of 0 means the off time is determined by counting BCLKS. The time is coded according to the following table. The time is measured from DACK going false to DREQ going true. In EISA mode, this register is used to count the number of BCLKS before releasing the bus when preempted by the system.

BOFF (hex)	ISA Time (us)	EISA BCLKS
0	2 BCLKS	2
1	4	4
2	8	8
3	12	12
...	...	...
F	60	60

3-0 (0) BON(3:0) This value gives the maximum time on the bus. In ISA mode, the values loaded are the number of microseconds that the device will keep the bus provided no other event such as a lack of data causes it to release the bus. The time is measured from DACK going true to DREQ going false; however, DREQ will not go false unless at least one transfer takes place. In EISA mode, this count is not used.

BON (hex)	Time (us)
0	2 BCLKS
1	1
2	2
3	3
...	...
F	15

In EISA mode, bits 3-0 are redefined as follows:

3-1 (0) Not Used Always reads 0.

0 (0) MSBRSTCTL Master Burst Control. This control bit provides alternate timing for MSBURST-. When cleared, and during the start of a burst transfer, MSBURST- will go active upon the detection of SLBURST- active on the rising edge of BCLK during START- active, followed by the detection of EXRDY active on the falling edge of BCLK. When set and during the start of a burst transfer, MSBURST- will go active upon the detection of SLBURST- active only.

Bus Speed (BUSSPD)

Type: **R/W**
Address: **I-1346h, E-zC86h, C-86h**

This register adjusts the speed of transfer on the ISA bus to accommodate various implementations. The speeds and a discussion of the values loaded are given in the Functional Description under Bus Speed. In EISA mode, STBON(3:0) and STBOFF (3:0) have no meaning, but DFTHRSH(1:0) is still used.

BUSSPD	R/W
7	DFTHRSH1
6	DFTHRSH0
5	STBOFF2
4	STBOFF1
3	STBOFF0
2	STBON2
1	STBON1
0	STBON0

7-6	(0)	DFTHRSH1, DFTHRSH0	0,0 – Slow Host, fast SCSI; SCSI is much faster than the Host. Write operation – The Host will start transmitting when the FIFO count is FULL-4 double words until the FIFO is full. Read operation – The Host will read data from the FIFO as soon as the count reaches 4 double words and will continue until the FIFO is empty. 0,1 – Nearly equal speeds, SCSI is faster than Host; 50% threshold. Write operation – When the FIFO empties to 50% full, the Host transfer logic will request the bus and transfer until the FIFO is full. Read operation – When the FIFO fills to 50% full, the Host transfer logic will request the bus and transfer until the FIFO is empty. 1,0 – Nearly equal speeds, SCSI equals or is slower than Host; 75% threshold. Write operation – When the FIFO empties to 75% empty, the Host transfer logic will request the bus and transfer until the FIFO is full. Read operation – When the FIFO fills to 75% full, the Host transfer logic will request the bus and transfer until the FIFO is empty. 1,1 – Fast Host, slow SCSI; Host is much faster than SCSI. Write operation – When the FIFO is empty, the Host transfer logic will request the bus and transfer until the FIFO is full. Read operation – When the FIFO is full, the Host transfer logic will request the bus and transfer until the FIFO is empty.
5-3	(0)	STBOFF(2:0)	The value loaded in STBOFF determines high or off time that the ISA memory write or read strobe will remain between strobes. The values will be set according to the following table.
2-0	(0)	STBON(2:0)	The value loaded in STBON determines the minimum low or on time that the ISA memory write or read strobe will remain active. This time may be extended by the assertion of CHRDY. The values will be set according to the following table. The times noted assume a 40 MHz input clock frequency.

STBON/STBOFF 2 1 0	TIME ns	Address/Data Setup ns	Address/Data Hold ns
0 0 0	100	70	30
0 0 1	150	120	30
0 1 0	200	120	80
0 1 1	250	170	80
1 0 0	300	170	130
1 0 1	350	220	130
1 1 0	400	270	130
1 1 1	500	320	180

Host Control (HCNTRL)

Type: R/W
Address: I-1347h, E-zC87h, C-87h

This register provides overall host control of the device. It may be written at any time without consideration of the state of the Sequencer.

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

7 (0) Not used — Always reads 0.

6 (0) POWRDN — When set, it degates both the clock internal to the device and the internal BCLK. When cleared, it enables both clocks. POWRDN also disables interrupts to the Host bus and limits access to Host only registers. PAUSE must be set before POWRDN is set.

5 (0) Not used — Always reads 0.

4 (0) SWINT — Software Interrupt. When set, will assert the IRQ signal. IRQ will remain asserted until this bit is cleared. Clearing INTEN will inhibit SWINT and will de-assert IRQ.

3 (0) IRQMS — IRQ Mode Select. When set, a low true level interrupt on the IRQ pin is selected. When cleared, a high true edge interrupt on the IRQ pin is selected.

2	(1)	PAUSE[ACK]	Pause/Pause Acknowledge. When this bit is set, the Sequencer will be paused. When this bit is read, it gives the PAUSEACK 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 program counter (SEQADDR). Clearing this bit will release the Sequencer and it will continue at the current value of the program counter. This bit is also set by certain hardware conditions listed below. See the section Pause under Sequencer for more information.

BRKADRINT	bit 3, INTSTAT
SCSIINT	bit 2, INTSTAT
STEP	bit 2, SEQCTL
RSTDRV	pin
CHIPRST	bit 0, HCNTRL

1	(0)	INTEN	Interrupt Enable. When set, system interrupts are enabled and will occur when an interrupt event happens. Interrupt events are those listed in INTSTAT, SIMODE0 and SIMODE1, or SWINT.
0	(1)	CHIPRESET [ACK]	Device Reset. When set, the device is put in a reset state for a maximum of 3 input clocks. CHIPRESETACK will remain set until explicitly cleared by a write to this register. RESDRV active will also set CHIPRESETACK.

Host Address (HADDR(3:0))

Type: R/W
Address: I-1348/1349/134A/134Bh, E-zC88/zC89/zC8A/zC8Bh, C-88/89/8A/8Bh

These registers contain the Host Address of the data being transferred. They are counted up as the data is transferred across the host interface. An attempt to write these registers with HDMAEN (bit 3, DFCNTRL) set will cause ILLSADDR (bit 1, ERROR) to be set.

HADDR0 R/W		HADDR1 R/W		HADDR2 R/W		HADDR3 R/W	
7	HADDR(07)	7	HADDR(15)	7	HADDR(23)	7	HADDR(31)
6	HADDR(06)	6	HADDR(14)	6	HADDR(22)	6	HADDR(30)
5	HADDR(05)	5	HADDR(13)	5	HADDR(21)	5	HADDR(29)
4	HADDR(04)	4	HADDR(12)	4	HADDR(20)	4	HADDR(28)
3	HADDR(03)	3	HADDR(11)	3	HADDR(19)	3	HADDR(27)
2	HADDR(02)	2	HADDR(10)	2	HADDR(18)	2	HADDR(26)
1	HADDR(01)	1	HADDR(09)	1	HADDR(17)	1	HADDR(25)
0	HADDR(00)	0	HADDR(08)	0	HADDR(16)	0	HADDR(24)

Host Count (HCNT(n))

Type: R/W
Address: I-134C/134D/134Eh, E-zC8C/zC8D/zC8Eh, C-8C/8D/8Eh

These registers contain a count of the data being transferred. They are loaded by the controlling firmware before the transfer begins. They are counted down during the transfer and prevent any host cycles when equal to zero. An attempt to write these registers with HDMAEN (bit 3, DFCNTRL) set will cause ILLSADDR (bit 1, ERROR) to be set.

HCNT0 R/W		HCNT1 R/W		HCNT2 R/W	
7	HCNT(07)	7	HCNT(15)	7	HCNT(23)
6	HCNT(06)	6	HCNT(14)	6	HCNT(22)
5	HCNT(05)	5	HCNT(13)	5	HCNT(21)
4	HCNT(04)	4	HCNT(12)	4	HCNT(20)
3	HCNT(03)	3	HCNT(11)	3	HCNT(19)
2	HCNT(02)	2	HCNT(10)	2	HCNT(18)
1	HCNT(01)	1	HCNT(09)	1	HCNT(17)
0	HCNT(00)	0	HCNT(08)	0	HCNT(16)

SCB Pointer (SCBPTR)

Type: R/W
Address: I-1350h, E-zC90h, C-90h

The SCB Pointer provides the upper address to the SCB Array. The value loaded in this register selects a group of 32 registers which contain a description of an executable command. Changing this value during execution will not alter any data, but will address a different element of the array.

SCBPTR R/W	
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	SCBVAL(2)
1	SCBVAL(1)
0	SCBVAL(0)

7-3	(0)	Reserved	Always reads 0.
2	(0)	SCBVAL(2)	This bit returns the value that was written to it, and has no other function.
1-0	(0)	SCBVAL(1:0)	This value selects the page of the SCB Array which appears in the SCB address range.

Interrupt Status (INTSTAT)

Type: **R**
Address: **I-1351h, E-zC91h**

Type: **W**
Address: **C-91h**

This register provides interrupt status for the driver when a system interrupt occurs. This status is written to by the Sequencer, and may be read by the driver without pausing the Sequencer. The Sequencer is paused when the SEQINT, SCSIINT, or BRKADRINT is set. The INTCODE is valid only when SEQINT is set.

INTSTAT	R/W
7	INTCODE(3)
6	INTCODE(2)
5	INTCODE(1)
4	INTCODE(0)
3	BRKADRINT
2	SCSIINT
1	CMDCMPLT
0	SEQINT

7-4	(0)	INTCODE(3:0)	This value is a code which further describes the situation of the interrupt. It is valid when SEQINT is set. See the discussion on Interrupts for a definition of this code.
3	(0)	BRKADRINT	This bit is set when the program counter of the Sequencer and the break address are equal and the breakpoint is enabled (BRKDIS=0). When this bit is set, the Sequencer is paused immediately. IRQ will be active if BRKADRINTEN (bit 3, SEQCTL) is set. BRKADRINT is cleared by setting CLRBRKADRINT (bit 3, CLRINT). This bit is also set by a hardware failure upon detection of the following events: Illegal Sequencer opcode Illegal I/O address HADDR write with HDMAEN set HCNT write with HDMAEN set DFIFO I/O write when SDMAEN or HDMAEN set
2	(0)	SCSIINT	SCSI Interrupt. This bit is set when there is a catastrophic SCSI event. Causes are SCSI Reset, Parity Error, Selection Timeout, 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. This interrupt is also qualified with SELBUSB (bit 3, SBLKCTL) so that it reflects only the currently selected channel. When this bit is set, the Sequencer is paused immediately.
1	(0)	CMDCMPLT	Command Complete Interrupt. This bit is set during normal operation after a command has been completed and the SCB Pointer has been loaded on the Queue Out FIFO. The Sequencer will continue running when this bit is set.
0	(0)	SEQINT	Sequencer Interrupt. This bit is set by the Sequencer when it requires driver intervention to complete a command or to handle an exception condition. The Sequencer is paused by this interrupt immediately.

Clear Interrupt Status (CLRINT)

Type: W
Address: I-1352, E-zC92h

This register allows the driver to clear the cause of the interrupt from the device. When a bit is set, the interrupt will be cleared. 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. The reset state will cause all bits to be active as if written by the host.

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

7-4	(0)	Not Used	Always reads 0.
3	(0)	CLRBRKADRINT	Clear Break Address Interrupt. When this bit is set, the BRKADRINT will be cleared. This bit will self-clear and does not need to be cleared. If BRKADRINTEN (bit 3, SEQCTL) and FAILDIS (bit 5, SEQCTL) are cleared, BRKADRINT has been set because of a hardware failure. It may only be cleared in this case by setting CHIPRST (bit 0, HCNTRL).
2	(0)	Not Used	Always reads 0.
1	(0)	CLRCMDINT	Clear Command Complete Interrupt. When this bit is set, CMDCMPLT is cleared. This bit will self clear and does not need to be cleared.
0	(0)	CLRSEQINT	Clear Sequencer Interrupt. When this bit is set, SEQINT is cleared. This bit will self-clear and does not need to be cleared.

Hard Error (ERROR)

Type: R
Address: I-1352h, E-zC92h, C-92h

This register reports errors that are catastrophic in nature. These errors will cause BRKADRINT to be set and the sequencer to be paused.

ERROR R	
7	
6	
5	
4	
3	PARERR
2	ILLOPCODE
1	ILLSADDR
0	ILLHADDR

7-4	(0)	Not Used	Always reads 0.
3	(0)	PARERR	Sequencer RAM Parity Error. This bit is a one when a Parity Error is detected while the Sequencer firmware is running. It may also be set if reading uninitialized RAM.
2	(0)	ILLOPCODE	Illegal Opcode. This bit is set when an instruction is executed by the Sequencer which is not defined.
1	(0)	ILLSADDR	Illegal Sequencer Address. This bit is set when the Sequencer accesses an address which does not decode to a defined register, an Illegal Opcode is detected, HADDR or HCNT is written with HDMAEN set, or DFIFO is written when HDMAEN or SDMAEN is set.
0	(0)	ILLHADDR	Illegal Host Address. This bit is set when the Host accesses a register, which is unavailable to the Host, while the Sequencer is not paused .

Data FIFO Control Register (DFCNTRL)

Type: R/W
Address: I-1353h, E-zC93h, C-93h

This register provides data path hardware control. Some bits are self-clearing and some must be cleared by the firmware. The enable bits, FIFORESET, and DIRECTION may be set at the same time. The hardware is designed to prevent any false triggering due to clock delays in different parts of the circuit.

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

7 (0) Not Used — Always reads 0.

6 (0) WIDEODD — This bit has no effect except during Wide transfer with an odd byte count. When set, the last byte 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. It should always be set for Scatter/Gather transfers when 16-bit SCSI transfers are used, except for the last list element.

5 (0) SCSIEN[ACK] — SCSI Transfer Enable/SCSI Transfer Enable Acknowledge. When this bit is set to a one it enables transfers between the SCSI bus and one of the SCSI FIFOs. Clearing this bit will cleanly halt the transfer by preventing ACKs to the SCSI bus. Reading this bit gives SCSIENACK. This is a status bit 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 non-zero and the SCSI bus is in Data-in phase.

4 (0) SDMAEN[ACK] — SCSI DMA Enable/SCSI DMA Enable Acknowledge. When this bit is set to a one it enables transfers between the SCSI block and Data FIFO. Reading this bit gives SDMAENACK, a status bit which indicates the state of the hardware. When this bit is cleared, transfers are disabled, but it must be read back as zero before the transfer is guaranteed to have halted.

3	(0)	HDMAEN[ACK]	Host DMA Enable/Host DMA Enable Acknowledge. When set this bit enables the host interface to transfer data to or from system memory. The address pointer and byte counter must be set up prior to setting this bit. Clearing this bit will halt transfers without losing data, status or byte count. Transfers may be continued after halting. Reading this bit gives HDMAENACK. This is a status bit which indicates the state of the hardware. When this bit is cleared, transfers are disabled, but it must be read back as zero before the transfer is guaranteed to have halted.
2	(0)	DIRECTION [ACK]	DMA Direction. This bit configures the hardware for the direction of transfer. The bit is set for a write operation, that is from the Host bus to the SCSI bus. When the bit is cleared, a read operation is assumed, that is from the SCSI bus to the Host bus. DIRECTIONACK will not change unless the enable bits (bits 3, 4, and 5) are cleared.
1	(0)	FIFOFLUSH [ACK]	During a SCSI to Host transfer (a read operation as initiator), this bit is set to force the remaining bytes in the Data FIFO to be sent to the host memory. If FIFOEMP is set, then setting this bit will have no effect. This bit is self-clearing and has no effect during a write operation. The FIFO will also be flushed by hardware on STCNT=0 or a SCSI phase change. When this bit is read as a one, it indicates a flush operation is pending or in progress due to either a firmware or hardware flush. It will be read as a zero when the flush operation is done. During Sequencer writes to the FIFO, setting this bit will force the Host or SCSI logic to recognize a partial double word load. This feature allows any number of bytes to be loaded without considering byte alignment in the Data FIFO.
0	(0)	FIFORESET	FIFO Reset. When this bit is set, the Data FIFO pointers are reset. The status will reflect that the FIFO is empty. This bit must be set after loading HADDR in order to set the byte offset pointers which are determined from HADDR(00) and HADDR(01). This bit is self-clearing.

Data FIFO Status (DFSTATUS)

Type: R
Address: I-1354h, E-zC94h, C-94h

This register contains status about the state of various hardware.

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

7, 6	(0)	Not Used	Always reads 0.
5	(0)	DWORDEMP	Double Word Empty. This bit indicates that the DFIFO read and write addresses are the same. There may be data elsewhere in the channel.
4	(0)	MREQPEND	Master Request Pending. This bit is set when the Host interface logic has reached a condition which requires a data transfer on the Host bus. When this occurs, this bit is set and a request for Bus Master control of the bus is generated. The MREQ signal is still subject to other conditions and may not appear on the bus at the same time this bit is set. The bit is cleared when there is no requirement for a host transfer, or when HDONE is set, or when FIFORESET (bit 0, DFCNTRL) is set.
3	(0)	HDONE	Host Done. This bit is the logical AND of HCNT=0 and the last transfer complete. This bit is automatically cleared when a non-zero value is stored in the HCNT(2:0) register.
2	(0)	DFTHRSH	When set, indicates that the threshold condition is now met. When cleared, indicates that the threshold condition is not being met. This signal is only valid when the count of bytes in the FIFO is equal to the threshold limit written to DFTHRSH(1:0). It is used for IC test only.
1	(0)	FIFOFULL	Data FIFO Full. When set, it indicates that the Data FIFO is full.
0	(0)	FIFOEMP	Data FIFO Empty. When set, it indicates that the Data FIFO and all data holding registers are empty.

Data FIFO Write Address 0 (DFWADDR0)

Type: R/W
Address: I-1355/1356, E-zC95/zC96h, C-95/96h

This register contains the address of the Data FIFO write pointer. Each location points to a 32-bit word. Writing to this register will alter the write location to the Data FIFO. If a write is attempted when HDMAEN (bit 3, DFCNTRL) is set, then BRKADDRINT (bit 3, INTSTAT) will be set with ILLSADDR (bit 1, ERROR). Data may be written to the FIFO through DFDAT. The Data FIFO Write Address is incremented automatically when the high byte is written to the FIFO. The starting location (Low or High) is determined by the state of HADDR(00). If this bit is a zero, the first byte written goes to the Low byte and if a one, will go to the High byte. One register is reserved for future expansion. See the section on Reading and Writing the Data FIFO for more information.

DFWADDR0 R/W		RESERVED R/W	
7	TESTLOCK(R)	7	RSVD
6	DFWADDR(6)	6	RSVD
5	DFWADDR(5)	5	RSVD
4	DFWADDR(4)	4	RSVD
3	DFWADDR(3)	3	RSVD
2	DFWADDR(2)	2	RSVD
1	DFWADDR(1)	1	RSVD
0	DFWADDR(0)	0	RSVD

7 (0) TESTLOCK — This bit gives the status of the DFIFO RAM stress test. See TESTCHIP register.

6 (0) DFWADDR(6) — DFIFO Write Address bit 6. This bit indicates that the write address bits 5-0 have rolled over from 000000 to 111111. This bit is used to determine DFIFO full and empty status.

5-0 (0) DFWADDR(5:0) — DFIFO Write Address bits 5 to 0. Normally used DFIFO write address.

Data FIFO Read Address 0 (DFRADDR0)

Type: R/W
Address: I-1357/1358h, E-zC97/zC98h, C-97/98h

This register contains the address of the Data FIFO read pointer. Each location points to a 32-bit word. Writing to this register will alter the read location to the Data FIFO. If a write is attempted when HDMAEN (bit 3, DFCNTRL) is set, then BRKADDRINT (bit 3, INTSTAT) will be set with ILLSADDR (bit 1, ERROR). Data may be read from the FIFO through DFDAT. The Data FIFO read address is incremented automatically when the High byte is read from the FIFO. The starting location (Low or High) is determined by the state of HADDR(00). If this bit is a zero, the

first byte read comes from the Low byte and if a one, will come from the High byte. One register is reserved for future expansion. See the section on Reading and Writing the Data FIFO for more information.

DFRADDR0 R/W		RESERVED R/W	
7	DFSDH(R)	7	RSVD
6	DFRADDR(6)	6	RSVD
5	DFRADDR(5)	5	RSVD
4	DFRADDR(4)	4	RSVD
3	DFRADDR(3)	3	RSVD
2	DFRADDR(2)	2	RSVD
1	DFRADDR(1)	1	RSVD
0	DFRADDR(0)	0	RSVD

7 (0) DFSDH — This bit indicates the DFIFO contains data up to Full-2 double-word location. See TESTCHIP register.

6 (0) DFRADDR(6) — DFIFO Read Address bit 6. This bit indicates that the read address bits 5-0 have rolled over from 000000 to 111111. This bit is used to determine DFIFO full and empty status.

5-0 (0) DFRADDR(5:0) — DFIFO Read Address bits 5 to 0. Normally used DFIFO Read Address.

Data FIFO Data Register (DFDAT)

Type: R/W
Address: I-1359h, E-zC99h, C-99h

This register stores data into the Data FIFO using DFWADDR0 when written and reads data from the Data FIFO using DFRADDR0 when read. Before writing or reading to system memory, HADDR should be set up and FIFORESET (bit 0, DFCNTRL) should be set to initialize the correct offset into the FIFO. DFWADDR and DFRADDR may be adjusted after the FIFORESET if a specific location is desired. For more information, see Reading and Writing the Data FIFO.

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

SCB Auto Increment (SCBCNT)

Type: R/W
Address: I-135Ah, E-zC9Ah, C-9Ah

This register provides the starting address and the enable bit for the auto increment feature of the SCB RAM. When SCBAUTO is set, any access to/from the SCB I/O address space will use the contents of the counter for an offset into the SCB area for both read and write accesses. Each access will increment the counter by one. When SCBAUTO is cleared, any access will use the value of the I/O address directly.

SCBCNT	R/W
7	SCBAUTO
6	RSVD
5	RSVD
4	SCBCNT(4)
3	SCBCNT(3)
2	SCBCNT(2)
1	SCBCNT(1)
0	SCBCNT(0)

7	(0)	SCBAUTO	When set, enables SCBCNT to supply the address offset into the SCB Array.
6, 5	(0)	Reserved	Always reads 0.
4-0	(0)	SCBCNT(4:0)	SCB Array counter. When SCBAUTO is set, the value written to these bits is used as the offset into the SCB Array. After each access, the count is incremented by one. When SCBAUTO is cleared, this field is ignored.

Queue In FIFO (QINFIFO)

Type: R/W
Address: I-135Bh, E-zC9Bh, C-9Bh

This register is written by the controlling driver with the SCB Pointer value of the SCB which was just loaded. Writing to QINFIFO puts a value onto the queue. The FIFO is as deep as the number of SCB Array pages, and so it can queue as many commands as may be loaded in the SCB area. The Sequencer will read the FIFO which will remove one value from the queue. The status of the FIFO is given in QINCNT. Writes when QINCNT=4 or reads when QINCNT=0 are ignored.

	QINFIFO R/W
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	RSVD
1	QIN(1)
0	QIN(O)

Queue In Count (QINCNT)

Type: R
Address: I-135Ch, E-zC9Ch, C-9Ch

This register contains the count of the number of entries in the Queue In FIFO.

	QINCNT R
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	QINCNT(2)
1	QINCNT(1)
0	QINCNT(0)

7-3 (0) Reserved — Always reads 0.

2-0 (0) QINCNT — These bits contain the count of the number of entries in the QINFIFO. A value of zero means the FIFO is empty, a value of one means there is one entry, etc. QINCNT will only have values between 0 and 4.

Queue Out FIFO (QOUTFIFO)

Type: R/W
Address: I-135Dh, E-zC9Dh, C-9Dh

This register is written by the Sequencer with the SCB Pointer value of the SCB which was just completed. Writing to QOUTFIFO puts a value onto the queue. The FIFO is as deep as the number of SCB Array pages, and so it can queue as many commands as may be loaded in the SCB area. The controlling driver will read the FIFO which will remove one value from the queue. The status of the FIFO is given in QOUTCNT. QOUTFIFO may be read by the driver without pausing the Sequencer. Writes when QOUTCNT=4 or reads when QOUTCNT=0 are ignored.

QOUTFIFO	R/W
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	RSVD
1	QOUT(1)
0	QOUT(0)

Queue Out Count (QOUTCNT)

Type: R
Address: I-135Eh, E-zC9Eh, C-9Eh

This register contains the count of the number of entries in the QOUTFIFO. QOUTCNT may be read by the driver without pausing the Sequencer.

QOUTCNT	R
7	RSVD
6	RSVD
5	RSVD
4	RSVD
3	RSVD
2	QOUTCNT(2)
1	QOUTCNT(1)
0	QOUTCNT(0)

7-3	(0)	Reserved	Always reads 0.
2-0	(0)	QOUTCNT	These bits contain the count of the number of entries in the QOUTFIFO. A value of zero means the FIFO is empty, a value of one means there is one entry, etc. QOUTCNT will only have values between 0 and 4.

Test Chip (TESTCHIP)

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

These bits select certain sections of the chip for test purposes. The pins may be redefined to provide ample I/O for chip test. This register should not be written during the normal course of operation. See the section on Test Features for a more complete description of this register.

TESTCHIP R/W	
7	TESTSEL(2)
6	TESTSEL(1)
5	TESTSEL(0)
4	TESTRAM
3	TESTHOST
2	TESTSEQ
1	TESTFIFO
0	TESTSCSI

7-5	(0)	TESTSEL(2:0)	Select the appropriate hardware configuration for testing. Used only for chip test. The meaning varies with the specific hardware section selected.
4	(0)	TESTRAM	Used only during chip test for special RAM stress testing.
3	(0)	TESTHOST	Select the Host block for testing.
2	(0)	TESTSEQ	Select the Sequencer block for testing.
1	(0)	TESTFIFO	Select the FIFO block for testing.
0	(0)	TESTSCSI	Select the SCSI block for testing.

4

Functional Description

SCSI

Wide/Dual Busses

The AIC-7770 contains two separate SCSI busses, SCSI Channel A and SCSI Channel B. These may be connected independently, or in tandem for 16-bit data transfer. Channel A is selected by clearing the SELBUSB bit in SBLKCTL with SELWIDE cleared and Channel B is selected by setting that bit with SELWIDE cleared. Both channels are mapped to the same I/O space. SCSI Channel A supports single-ended or differential mode of operation with external components. SCSI Channel B supports single-ended operation only. Both channels are capable of supporting up to 10 MByte/sec transfer rates. Channel A may also be operated as a Wide channel (16-bit data). In this case, data bits 8 through 15 plus parity are tied to data bits 0 through 7 plus parity on Channel B, and Channel B is not usable. When Channel A is in Wide mode the state of WIDEXFER in SXFRCTL0 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. Channel A controls allow Fast, Differential and Wide (at the expense of Channel B) transfers or any combination thereof. Channel B controls allow Fast single-ended transfers.

During the reset state, there is a mechanism to indicate that Channel B is not used or whether the device is connected to a Wide bus. This is accomplished by grounding certain pins on Channel B control signals according to the following table.

The data interface to the SCSI control logic within the device is 16 bits. 16-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 DFIFO 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 DFIFO 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 DFIFO 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. The following table illustrates how the decisions are made: (DIR 1=write, 0=read; CNT = bit 0 of STCNT; ADDR = bit 0 of SHADDR)

DIR*	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
*W = To SCSI R = To DFIFO				

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.

Initializing the SCSI Section

When the device is reset at power-on, the Sequencer is held in the paused state. This allows the driver to access the SCSI registers directly. Each SCSI channel 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 SCSIBUSL and SCSIBUSH, 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 data is presented on the SCSI bus and 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 (DFIFO) through the SCSI FIFO (SFIFO). 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 before hand. 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.

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

Bit 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

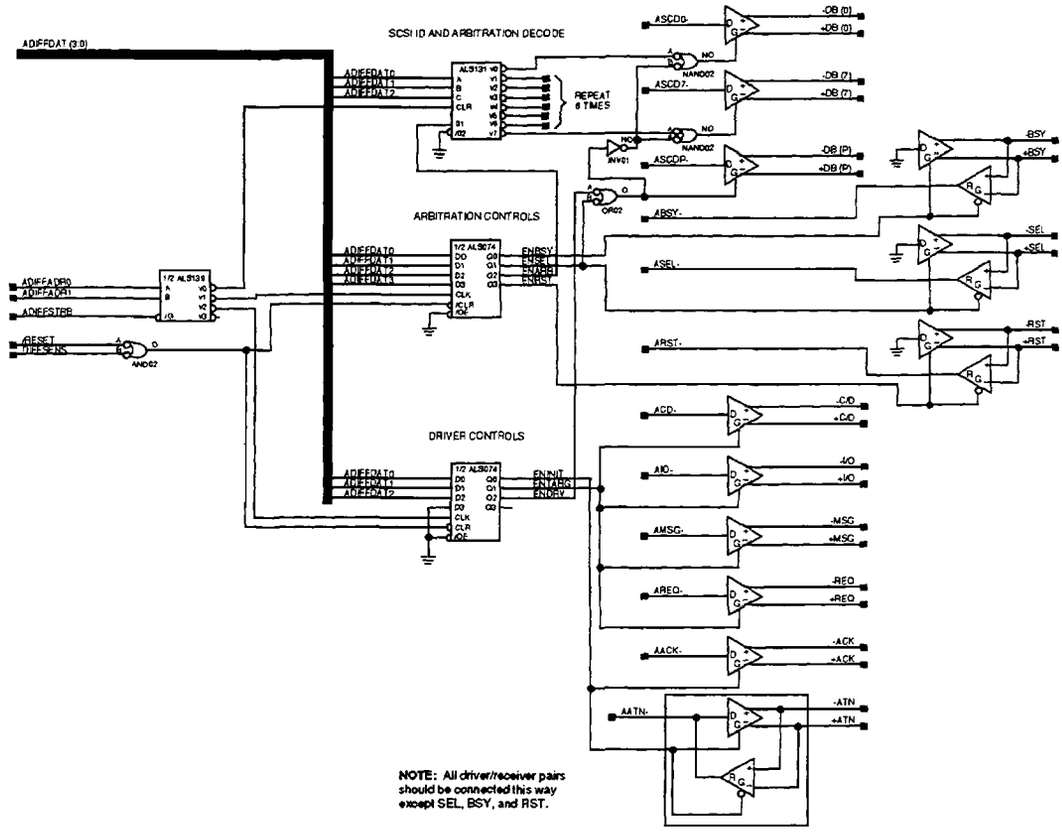


Figure 4-1. Differential Controls

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 Timeout. 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 & 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 Auto 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 HADDR0-3 (088-08Bh) is loaded. The current value of 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 non-zero transfer count or a zero transfer count with SWRAPEN set. (The event that both counters count 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 DFIFO. 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 SFIFO 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.

Chip Reset
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 address pointer HADDR(3:0) is loaded. HADDR 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. 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.

Pause should 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 pre-determined 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 IRQ pin will also be driven active. BRKADRINT and the interrupt may be cleared by setting CLRBRKADRINT (bit 3, CLRINT). A driver may do one of a number of things listed below:

- 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 a hardware failure detection mechanism. Illegal Opcodes, illegal address accesses, and Sequencer RAM Parity Errors will be detected and will cause a BRKADRINT interrupt which will pause the Sequencer and will drive the IRQ pin. The cause of the interrupt may be read from the ERROR register. If this condition occurs BRKADRINT may only be cleared by setting 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 instructions should be executed between the control write and the status read of the bit in interest to assure a valid status. Following is a list of status bits that fall into this category, the direction of the status bit change, and the action which causes the bit to change.

Status Bit	Control Causing Change
SDONE	Goes inactive after loading a non-zero value into STCNT, or changing SWRAPEN.
DMADONE	Goes inactive after loading a non-zero value into STCNT, changing SWRAPEN, or after loading a non-zero value in 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 DIRECTION (bit 2, DFCNTRL), if SCSIENACK, SDMAENACK, and HDMAENACK are zero.
FIFOFLUSHACK	Goes active after setting FIFOFLUSH (bit 1, DFCNTRL).
DWORDEMP	Goes active after reading DFDAT with the DFIFO containing 5 bytes, or after setting FIFORESET. Goes inactive after writing to DFDAT with the DFIFO containing 3 bytes.
MREQPEND	Goes active when setting HDMAEN=1, DIRECTION=1, and FIFOEMP=1; or HDMAEN=1, DIRECTION=0, and FIFOFULL=1; or FIFOFLUSH=1, DIRECTION=1, HDMAENACK=1, and FIFOEMP=0. Goes inactive when setting FIFORESET with HDMAENACK=0.
HDONE	Goes inactive after loading a non-zero value in HCNT.
DFTHRSH	Goes active or inactive with a write to or read from DFDAT.
FIFOFULL	Goes active with a write to DFDAT and the number of stored bytes in the DFIFO is 4 less than full. Goes inactive with a read from DFDAT and the number of stored bytes in the DFIFO is 3 less than full or by setting FIFORESET.
FIFOEMP	Goes active with a read from DFIFO one stored byte in the FIFO, or by setting FIFORESET. Goes inactive with a write to DFIFO with DFIFOEMP=1.

Host in ISA Mode

Device Identification

The EISA configuration registers are available in ISA mode for identification purposes. BID0, BID1, BID2, and BID3 are hard wired values.

Bus Time

The Bus-on and Bus-off time may be adjusted by writing to BUSTIME. Bus-on values are from 2 BCLKS or 1 to 15 usec and Bus-off values are from 2 BCLKS or 4 to 60 usec in increments of 4 usec. The bus time should be chosen to optimize system throughput. For a minimum configuration, a large Bus-on time with a small Bus-off time gives the best performance. When another I/O device is on the bus and used at the same time as the device, overall system performance may be improved by reducing Bus-on and/or increasing Bus-off time. The device must get off the bus every 15 us minimum to allow refresh to occur.

Bus-on time indicates the time that the device will stay on the Host bus before giving it up. It is measured from DACK asserted to DREQ de-asserted. Bus-on time may be longer than the value programmed in the case of a slow data transfer rate and a minimum Bus-on time. In addition, there is some time after the transfer equal to or less than one BCLK cycle in order to synchronize DREQ de-assertion.

Bus-off time is the time the device will stay off the bus before requesting it again. It is measured from DACK de-asserted to DREQ asserted.

Bus Speed

Bus speed may be adjusted to provide the maximum bus bandwidth allowed on the ISA bus. Care should be taken when choosing a bus speed value to consider the Bus-on time choice. For slower speeds, the on time may be over before the first transfer has completed. In any case, at least one transfer will take place before the bus is given up by the device. This will, however, extend the Bus-on time. The strobe widths and the time between strobes may be adjusted for maximum throughput. The active strobe time is set by setting the STBON value in the BUSSPD register. The time between strobes is set by loading the STBOFF value in BUSSPD.

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 HDMAEN , SDMAEN , and SCSIEN bits in DFCNTRL should be set. 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.

MEM16 Control

In order to support 8-bit video RAM transfers as a Bus Master, the hardware will determine whether to transfer in 8 or 16-bit mode by comparing the address of the transfer. If the starting address is between A0000h and BFFFFh, then MEM16 determines whether the transfer is 8 or 16-bit. All other starting addresses will transfer 16-bits at a time.

Host in EISA Mode

Configuration

The configuration ID will be hard-wired registers. All chip setup will take place by the controlling BIOS or driver at initialization time. The configuration ID will be ADP7770.

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.

Bus Master Modes

The device will implement four 32-bit Master modes of operation. These modes are transparent to all operations and the particular mode is chosen by a combination of control bits on the EISA bus summarized below. All modes are listed in order of decreasing performance:

- 32-bit Burst
- 32-bit 2 cycle
- 32-bit Downshift to 16-bit
- 32-bit System Translate

Slave	Burst	Non-Burst
ISA 8-bit	X	32-bit System Translate EX32- = 1 SLBURST- = 1
ISA 16-bit	X	32-bit System Translate EX32- = 1 SLBURST- = 1
EISA 16-bit	32-bit Downshift to 16-bit EX32- = 1 SLBURST- = 0	32-bit System Translate EX32- = 1 SLBURST- = 1
EISA 32-bit	32-bit Burst EX32- = 0 SLBURST- = 0	32-bit 2 cycle EX32- = 0 SLBURST- = 1

Reading and Writing the Data FIFO

The DFIFO 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 32-bit double words. Data is properly aligned in conjunction with the state of bits HADDR(00) and HADDR(01). 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 HADDR(00) and HADDR(01). Normal data transfer does not require any intervention from the Sequencer. The correct offset is set up automatically when the Host address is loaded in HADDR 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 HADDR, 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 HADDR and HCNT, resetting the FIFO, writing the data to DFDAT, and clearing 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 HADDR, 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 HADDR and HCNT, resetting the FIFO, and setting 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.

Below is a table of which port is active and in which direction with respect to the Data FIFO.

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

Data is aligned in the Data FIFO and the SCSI FIFO according to the offset of the starting address from a double word boundary. The alignments in the DFIFO and SFIFO are summarized below.

HADDR		DFIFO 1st Byte Offset			SCSI BLOCK
(01)	(00)	SEQ	SCSI	Host	Byte Offset
0	0	0	0	0	0
0	1	1	0	0	1
1	0	2	2	0	0
1	1	3	2	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 of the 286/386™ instruction set 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 the section SCB Definition under Application Notes 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 Queue In FIFO. 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 Queue In FIFO 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 Queue In FIFO. 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 Queue In FIFO. 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 Queue Out FIFO and will interrupt the driver with CMDCMPLT (bit 1, INTSTAT) set. The driver will then read the Queue Out FIFO 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 Queue Out FIFO. 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 Timeout. 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 pre-determined 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 IRQ 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, illegal I/O address, or Sequencer RAM Parity Error. This feature may be disabled by setting FAILDIS (bit 5, SEQCTL).

Software Interrupt

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

Interrupt Summary

The following table lists the conditions under which the host may be interrupted.

Description	Enable Conditions	Pause	INTSTAT Bit	Error Bit
Sequencer parity error	PERRORDIS=0 AND parity error detected during opcode read	Yes	BRKADRINT	PARERR
Illegal opcode	FAILDIS=0 AND illegal opcode detected	Yes	BRKADRINT	ILLOPCODE
Illegal sequencer address detected	FAILDIS=0 AND illegal sequencer address detected	Yes	BRKADRINT	ILLSADDR
Illegal host address detected	FAILDIS=0 AND illegal host address detected	Yes	BRKADRINT	ILLHADDR
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

Power Down

Power may be conserved by degating the clock to most of the chip. Setting POWRDN (bit 6, HCNTRL) will cause the entire chip with the exception of I/O decode logic to remain in a quiescent state. This will also disable any interrupts that may be generated independent from the clock. Interrupts pending in this case will drive IRQ as soon as POWRDN is cleared. The Sequencer should 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. The series of tests are listed in the following table.

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
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
FIFO check by Sequencer	Sequencer verifies Write/Read ability
Data path	Sequencer transfers data through FIFO/SCSI interface
Interrupts	Driver verifies proper interrupt operation
Queue In/Queue out	Driver verifies proper operation
Power-down mode	Driver verifies power down non-operation
EISA enable bit	Driver verifies enable bit when EISA bus is selected

5

Application Notes

Chip Initialization

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

- SCSI ID
- Bus release time - EISA bus
- Bus on/off time - ISA bus
- DFIFO thresholds
- Interrupt level

SCSI Phases

Arbitration/Selection

Arbitration and selection are automatic hardware sequences which are started by the Sequencer. 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, and 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-7770 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 non-zero, 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.

SCB Definition

Offset	Stack Contents	# Bytes
00	Control 7 Reject MDP message 6 Disconnect enable 5 Tag enable 4 Reserved 3 Waiting 2 Disconnected 1,0 00 - Simple queue 01 - Head of queue 10 - Ordered queue 11 - Illegal	1
01	Target ID, Chan, LUN	1
02	S/G Segment Count	1
03	S/G List Pointer	4
07	SCSI Command Pointer	4
11	SCSI Command Length	1
12	Reserved	1
13	Reserved	1
14	Target Status	1
15	Residual Data Count	3
18	Residual S/G Segment Count	1
19	Data Pointer	4
23	Data Count	3
26-31	Reserved	6
	Total	32

Scratch RAM Definition

The scratch RAM area contains general purpose RAM used during the execution of commands and to store configuration data which describes the system setup. The content of the RAM is defined by the firmware in the Sequencer.

Multi-threaded Operation

More than one Target device may have commands open but disconnected. The four SCBs are general purpose and may be used in any combination on either channel. The Sequencer will match the Target/Channel/LUN when a new SCB ID is received from the Queue In FIFO. 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 Queue In FIFO. If there is and the Target/Channel/LUN matches an open command, the ID will be pushed back on the Queue In FIFO. 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. If two Reselections happen at the same time, then a fairness algorithm is used to prevent one channel from being locked out.

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, SFIFO 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 SFIFO 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 busses will generally be different, and so the 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 and 7, BUSSPD) 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 EISA mode, a value of 1,1 will be chosen as the default. In ISA 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 ISA 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 non-zero 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 Queue In FIFO, 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 Queue In FIFO 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 non-zero 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

Target mode will not be implemented in the first version of the device, 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 will require additional Sequencer code.

Programming the Sequencer

The program for the Sequencer is written in an assembly-level language, very similar to the assembly language for the Intel 80x86 microprocessors. The instruction set for this language is defined in the following section. The instruction formats and mnemonics were chosen to provide a simple and familiar language for the programmer, while at the same time allowing a straightforward compiler architecture. The instructions specify one, or sometimes a combination, of the following operations:

- Move a source register or a constant to a destination register
- Perform a logical or arithmetic operation on a source register, and move the result to a destination register
- Shift or rotate the contents of a source register, and move the result to a destination register; conditionally branch the program execution, as determined by the contents of a register
- Unconditionally branch the program execution

The instruction set was optimized such that the most commonly performed operations are the most powerful, while less common operations can be performed, but maybe less efficiently. For example, a bit test and branch is very common, and can be performed in one instruction. On the other hand, a multiple-precision subtraction is not common, and therefore requires several instructions.

The Sequencer program is written as a list of instructions, which are transformed by a compiler to a set of *command lines* for the Sequencer. These command lines are loaded into the Sequence program RAM for execution by the Sequencer. The command lines were defined in such a way as to maximize information density, thereby minimizing chip area required for the Sequencer RAM. A consequence of a compact

command line is that programming with the command line set is very difficult, necessitating the higher level language for the programmer.

Program development for the Sequencer is done with the Microsoft® Macro Assembler. A series of instructions are defined and implemented in a macro include file which will compile to a linkable object module. The resulting module will be in a form which may be directly loaded into the Sequencer RAM. See Functional Description/Sequencer/Loading for more information.

Instruction Set Definition

The following is the definition of the instruction set to be used by programmers in generating Sequencer programs for the AIC-7770 chip. These instructions are compiled to multi-byte command lines which are loaded into the Sequencer.

Definitions

The following conventions are used throughout this section:

- **A:** accumulator
- **ret:** return
- [ ]: optional
- /: alternative
- **Z:** zero flag
- **CY:** carry flag
- →: move from one position to another

Instruction Set

There are six classes of instructions: byte moves, logical operations, arithmetic functions, shifts and rotates, branches, and flag operations. Each instruction is modeled by a list of operations, which gives the programmer an effective sequence of events performed by the instruction. The programmer can consider that these events actually occur, although in reality the Sequencer may perform equivalent but different operations. The operations performed by an instruction are performed in the same order as listed in the instruction definition. All instructions compile to one Sequencer command line, unless otherwise indicated.

Move

mov destination,source [ret]

Source → destination.
Return (optional).
Flags affected: Z

mvi destination,immediate [ret]

Immediate → destination.
Return (optional).
Flags affected: Z

Logical

not destination[,source] [ret]

Source → destination (optional).
One's complement of destination → destination.
Return (optional).
Flags affected: Z

and destination,immediate/A[,source] [ret]

Source → destination (optional).
Logical AND of destination and immediate/accumulator → destination.
Return (optional).
Flags affected: Z

or destination,immediate/A[,source] [ret]

Source → destination (optional).
Logical OR of destination and immediate/accumulator → destination.
Return (optional).
Flags affected: Z

xor destination,immediate/A[,source] [ret]

Source → destination (optional).
Logical Exclusive OR of destination and immediate/accumulator → destination.
Return (optional).
Flags affected: Z

nop

No operation performed.
No destinations altered.
Flags affected: Z

Arithmetic

add destination,immediate/A[,source] [ret]

Source → destination (optional).
Arithmetic ADD without carry of destination and immediate/accumulator → destination.
If immediate = 0:
    Destination prior to ADD → accumulator.
    Compiles to two instructions.
Return (optional).
Flags affected: Z, CY

adc destination,immediate/A[,source] [ret]

Source → destination (optional).
Arithmetic ADD with carry of destination and immediate/accumulator → destination.
If immediate = 0:
    Destination prior to ADD → accumulator.
    Compiles to two instructions.
Return (optional).
Flags affected: Z, CY

inc destination[,source] [ret]

Source → destination (optional).
Increment destination → destination.
Return (optional).
Flags affected: Z, CY

dec destination[,source] [ret]

Source → destination (optional).
Decrement destination → destination.
Return (optional).
Flags affected: Z, CY

Shifts, Rotates

shl destination[,source],number [ret]

Source → destination (optional).
Shift destination left by number bit positions → destination.
256 > number >= 0
n = bits 2-0 of number
CY ← bit 8-n; bits n-1,n-2,...,0 ← 0
Return (optional).
Flags affected: Z, CY

shr destination[,source],number [ret]

Source → destination (optional).
Shift destination right by number bit positions → destination.
256 > number >= 0
n = bits 2-0 of number
0 → bits 7,...,8-n; bit n-1 → CY
Return (optional).
Flags affected: Z, CY

rol destination[,source],number [ret]

Source → destination (optional).
Rotate destination left by number bit positions → destination.
256 > number >= 0
n = bits 2-0 of number
CY ← bit 8-n; bit 0 ← bit 8-n
Return (optional).
Flags affected: Z, CY

ror destination[,source],number [ret]

Source → destination (optional).
Rotate destination right by number bit positions → destination.
256 > number >= 0
n = bits 2-0 of number
bit n-1 → bits 7; bit n-1 → CY
Return (optional).
Flags affected: Z, CY

rcl destination[,source] [ret]

Source → destination (optional).
Destination → accumulator.
Rotate destination left through carry → destination.
CY ← bit 7; bit 0 ← CY
Compiles to 2 command lines.
Return (optional).
Flags affected: Z, CY

xchg destination[,source] [ret]

Source → destination (optional).
Exchange nibbles in destination → destination.
Return (optional).
Flags affected: Z, CY

Branches

jmp/jc/jnc/call address

Unconditional jump/jump on carry/jump on not carry/call to next address.
Flags affected: Z

mov source jmp/jc/jnc/call address

Source → source index register.
Unconditional jump/jump on carry/jump on not carry/call to next address.
Flags affected: Z

mvi immediate jmp/jc/jnc/call address

Immediate → source index register.
Unconditional jump/jump on carry/jump on not carry/call to next address.
Flags affected: Z

or source,immediate jmp/jc/jnc/call address

Logical OR of source and immediate → source index register.
Unconditional jump/jump on carry/jump on not carry/call to next address.
Flags affected: Z

test source,immediate/A jz/jnz address

Logical AND of source and immediate/accumulator.
Jump on zero/not zero to next address.
No destinations are altered.
Flags affected: Z

cmp source,immediate/A je/jne address

Compare source and immediate/accumulator.
Jump on equal/not equal to next address.
No destinations are altered.
Flags affected: Z

ret

Unconditional return from subroutine.
No destinations are altered.
Flags affected: Z

Flag Operations

clc [mov destination,immediate/A] [ret]

Clear carry flag.
Immediate or accumulator → destination (optional).
Return (optional).
Flags affected: Z, CY

stc [destination] [ret]

Set carry flag.
Clear destination (optional).
Return (optional).
Flags affected: Z, CY

Command Line Format

The instruction set is implemented by defining a command line as a 29-bit register which is loaded on each Sequencer clock.

In general, each Sequencer command line performs one of two types of operations:

- Fetch the contents of a source register, modify the contents, and save the modified contents in a destination register.
- Examine a register, and execute the next program instruction or branch to a different instruction, based on the contents of the register.

Each command line is subdivided into fields of bits. There are three command line formats:

Format 1

command line with immediate, but not branch address, specified:

ALU/branch control	ret	destination	source	immediate
4	1	8	8	8

Format 2

command line for rotate ALU function:

ALU/branch control	ret	destination	source	shift control
4	1	8	8	8

Format 3

command line with branch address specified:

ALU/branch control	next address	source	immediate
4	9	8	8

The number of bits in each command line field is shown beneath the field label. The total of all fields is always 29 bits for all three line formats. Each field will be defined in detail below, but first an overview of the function of each command line will be given.

Each of the three command line types was designed to perform a specific type of operation:

- **Format 1:** The content of the register specified by the source field is combined with the immediate field, as specified by the opcode in the ALU/branch control field. The result is moved to the register specified by the destination field.
- **Format 2:** The content of the register specified by the source field is rotated or shifted by some number of bit positions, as specified by the shift control field. The result is moved to the destination register.
- **Format 3:** The content of the register specified by the source field is examined as specified by the ALU/branch control field and the immediate field. The result determines whether or not program execution branches. If the branch is taken, the next address field specifies the address of the next command line to be executed. Otherwise, the next command line in the list is executed.

Command Line Field Definitions

Each command line field is defined below. The definitions will provide more detail of the function of each command line.

Source/Destination

The registers specified by the source and destination fields are any of the registers defined in the AIC-7770 device specification. Some of these registers are defined again here, since they are specifically referred to in this document.

Register	Description
allones	A source register only, with contents fixed at 0ffh (all ones).
allzeros	A source register only, with contents fixed at 00h (all zeroes).
none	A destination register only, which has no function. It serves as a dump for an unwanted ALU output.
accumulator	A source or destination register. When specified as a destination, it is loaded by the output of the Sequencer ALU. When not specified as a destination, the accumulator is not altered by the command line execution. When specified as a source, it is read like any other source register. But in addition, the accumulator is the only register which can be implicitly accessed as a source in the same command line that a different register is explicitly specified as a source by the source field. (The only way two source registers can be combined is when one of them is the accumulator. This use of the accumulator is described more fully below.)
sindex	Command lines of the format 3 type do not have a field specifying a destination register. The destination register for all such command lines is the source index register, sindex.

The source register is not altered by the command line operation, unless the same register is also specified as the destination. Only the destination register is altered by the operation.

Next Address

When a branch in program execution is to be taken, the next address specifies the address of the next command line to be executed.

Immediate

The immediate field is a constant operand used by format 1 and format 3 commands. Note that an immediate field equal to zero has a special meaning for some ALU opcodes, as explained below. For these special cases, the constant operand is replaced by the contents of the accumulator as an operand.

Ret

The single-bit ret field specifies whether or not a subroutine return will be executed at the end of the command line operation. When ret is set to one, a subroutine return will be executed after the other command line operation is complete. When ret is set to zero, the next command line executed will be the next line in the program list.

Ret is an element of the set of branch controls, which are described below. It is the only element which is specified in a separate field. All other branch controls are encoded with the ALU functions to define the ALU/branch control field.

ALU/Branch Control

The ALU/branch control field is a component of all three command line types. In fact, it is decoded to identify the particular format of the command line. The interpretation of the other fields on the command line is therefore defined by this field. In addition, the ALU/branch control field is an encodation of the ALU function and the branch control, which specify the primary operation to be performed by the command line. The sets of ALU functions and branch controls are defined separately below, followed by a table of the encodations of the two.

ALU Functions

The Sequencer module contains an Arithmetic/Logic Unit (ALU), which performs logical and arithmetic operations plus a rotate function. The two inputs to the ALU are the source register and an operand. The following table contains the complete set of ALU functions:

ALU Function	Description
OR	Source register contents ORed with operand
AND	Source register contents ANDed with operand
XOR	Source register contents exclusive ORed with operand
ADD	Source register contents ADDed with operand, without carry
ADC	Source register contents ADDed with operand, with carry
ORI	Source register contents ORed with immediate
ROL	Source register contents rotated left, as specified by shift control

All ALU functions affect the zero flag. Individual functions have their own set of attributes:

ALU Function	Description
OR, AND, XOR	These functions perform the normal logical operations. When the immediate field is zero, the operand is the accumulator. When the immediate field is non-zero, the operand is the immediate. The carry flag is not altered
ADD	This function performs a normal arithmetic addition. When the immediate field is zero, the operand is the accumulator. When the immediate field is non-zero, the operand is the immediate. The carry is not added to the source register contents and operand. The carry flag is set to one if the sum overflows, and to zero for no overflow.
ADC	This function performs a normal arithmetic addition. When the immediate field is zero, the operand is the accumulator. When the immediate field is non-zero, the operand is the immediate. The carry is added to the source register contents and operand. The carry flag is set to one if the sum overflows, and to zero for no overflow.
ORI	The source register is logically ORed with the immediate field for all values of the immediate. Destination of the result is the source index register, sindex. The carry flag is not altered.
ROL	Source register is rotated left, as specified by the shift control field. The carry flag is altered. (See definition of the shift control field below.)

Branch Control

A branch is a change in direction of the flow of Sequencer program execution. The ALU/branch control field specifies the type of branch to be executed, if any. The complete set of branches are summarized here:

Branch	Description
JMP	Unconditional jump to address in next address field
JZ	Conditional jump on zero to address in next address field
JNZ	Conditional jump on not zero to address in next address field
JC	Conditional jump on carry to address in next address field
JNC	Conditional jump on not carry to address in next address field
CALL	Unconditional call to subroutine at address in next address field (Push current address + 1 onto stack for the return.)
RET	Unconditional subroutine return
NB	No branch

Each type of branch is explained in more detail:

Mnemonic	Branch Name	Description
JMP	Unconditional jump	Program execution branches unconditionally to the address specified in the next address field.
JZ	Jump on zero	If the result of the operation in the current command line is zero, program execution branches to the address specified in the next address field. If the result is not zero, the next command line executed is the next command line in the program list. (The zero flag is altered by every command line, and therefore the zero flag state after a command cannot be tested by a following command.)
JNZ	Jump on not zero	If the result of the operation in the current command line is not zero, program execution branches to the address specified in the next address field. If the result is zero, the next command line executed is the next command line in the program list. (The zero flag is altered by every command line, and therefore the zero flag state after a command cannot be tested by a following command.)
JC	Jump on carry	If the last command which alters the carry flag has set the carry flag, program execution branches to the address specified in the next address field of the current command. If the last command reset the carry flag, the next command line executed after the current command is the next command line in the program list. (The carry flag is altered only by commands which require a destination field. The jc branch requires a next address field. Since the next address and destination fields are shared on the command line, the jc branch control cannot coexist on the same command line with an ALU function that alters the carry flag. Therefore, the jc branch is a function of the carry state defined by a previous command.)

Continued

Mnemonic	Branch Name	Description
JNC	Jump on not carry	If the last command which alters the carry flag has reset the carry flag, program execution branches to the address specified in the next address field of the current command. If the last command set the carry flag, the next command line executed after the current command is the next command line in the program list. (The carry flag is altered only by commands which require a destination field. The jnc branch requires a next address field. Since the next address and destination fields are shared on the command line, the jnc branch control cannot coexist on the same command line with an ALU function that alters the carry flag. Therefore, the jnc branch is a function of the carry state defined by a previous command.)
CALL	Subroutine call	Program execution branches unconditionally via a subroutine call to the address specified in next address field. The address of the current command line incremented by one is pushed onto the stack.
RET	Subroutine return	Program execution branches unconditionally via a subroutine return to the address saved on the top of the stack. The return address is popped off the stack. A next address field is not required for this branch. Ret is the only branch control which is not encoded in the ALU/branch control field. It is specified in the ret field.
NB	No branch	No branch in program execution is taken. The next command line executed after the current command is the next command line in the program list. Since no branch is taken, no next address field is required on the command line.

Encodation

All elements of the branch control set, with the exception of ret, are encoded with the set of ALU functions to construct the ALU/branch control field. Branch controls requiring a next address field cannot share the command line with ALU functions which require a destination field, since these two field share the same bits on the command line. The set of ALU/branch control field values contains 14 elements, leaving an additional two elements available for future definition. The set of field values is tabulated below:

ALU Function	BranchControl	Opcode	Line Format
OR	NB/RET	0	1
AND	NB/RET	1	1
AND	JZ	15	3
AND	JNZ	13	3
XOR	NB/RET	2	1
XOR	JZ	14	3
XOR	JNZ	12	3
ADD	NB/RET	3	1
ADC	NB/RET	4	1
ORI	JMP	8	3
ORI	JC	9	3
ORI	JNC	10	3
ORI	CALL	11	3
ROL	NB/RET	5	2
RESERVED		6	-
RESERVED		7	-

The table is organized in groups of ALU functions, listed in the first column. The second column shows the branch control paired with the ALU function. The third column lists the opcode assigned to each ALU function/branch control pair. These opcodes reside in the command line ALU/branch control field. The last column shows the command line format type, which is specified by the opcode.

Command Line Functions

To describe the functions of command lines, it is convenient to break the complete set of command lines into subsets. The first level of subsets is identified by the command line format. There are three such subsets. A second level of subsets is defined by breaking each first-level subset up into a number of subsets, each of which is identified by the ALU function and branch control. The general functionality of commands in each of these subsets is described below:

Mnemonic	Command Name	Description
or nb	bit set	Bits which are set in the immediate field (or accumulator, if immediate is zero) are set in the destination register. Bits which are set in the source register are also set in the destination register. The next command line to be executed immediately follows the current command line in the program list.
or ret	bit set	Same as or nb, except a subroutine return branch is executed after the logical OR operation. The logical OR and the return are both executed by one command line.
and nb/ret	bit reset	The source register is first moved to the destination register. Then bits which are reset in the immediate field (or accumulator, if immediate is zero) are reset in the destination register.
and jz	bit test and branch	A branch to the address in the next address field is taken if none of the bits set in the immediate field (or accumulator, if immediate is zero) are set in the source register.
and jnz	bit test and branch	A branch to the address in the next address field is taken if one or more of the bits set in the immediate field (or accumulator, if immediate is zero) are set in the source register.
xor nb/ret	bit complement	The source register is first moved to the destination register. Then, the bits set in the immediate field (or accumulator, if immediate is zero) are complemented in the destination register.
xor jz	byte comparison	The source register content is compared with the immediate field (or accumulator, if immediate is zero). If the two are equal, a branch to the address in the next address field is taken at the end of the command line.
xor jnz	byte comparison	The source register content is compared with the immediate field (or accumulator, if immediate is zero). If the two are not equal, a branch to the address in the next address field is taken at the end of the command line.
add nb/ret	addition of two bytes	The content of the source register is added to the immediate field (or accumulator, if immediate is zero). The sum is moved to the destination register. This command type is used for single-precision addition, or for adding the least significant bytes in multi-precision addition.

Mnemonic	Command Name	Description
adc nb/ret	addition of two bytes	The content of the source register and the carry are added to the immediate field (or accumulator, if immediate is zero). The sum is moved to the destination register. This command type is used for multi-precision addition of bytes other than the least significant.
ori jmp/call	source index register (sindex) load; branch	A single command line will load the source index register (sindex) with the logical OR of the source register and immediate field, and then branch unconditionally to the address in the next address field. This command type is useful for passing a constant or variable into a subroutine via register sindex, and calling the subroutine.
ori jc/jnc	source index register (sindex) load; branch	A single command line will load the source index register (sindex) with the logical OR of the source register and immediate field, and then branch conditionally to the address in the next address field, depending on the state of the carry flag. (This command type does not alter the carry flag.)
rol	shift or rotate	The source register is first moved to the destination register, and then the destination register is either shifted or rotated.

Shift Control

The shift control field is used only in conjunction with the ROL ALU function. Its bits specify whether the function is a rotate or shift, and how many positions the bits are moved. A rotate moves all bits to the left, with bit 7 moving to bit 0 for each step. All bits are preserved. A shift moves all bits to the left, but also masks out certain bits after the rotate. For both rotates and shifts, the carry flag is set to the previous bit 7 or bit 0 value after each step of the move.

Bit	Definition
bits 2-0	Specify the number of rotation steps to the left.
bit 3	Direction bit - = 0: Bit mask is right-justified. Carry is set from bit 7. = 1: Bit mask is left-justified. Carry is set from bit 0.
bits 6-4	Mask encode - Equals the binary value of the number of contiguous bits to be masked out. Bit 3 aligns the mask with the right or left byte boundary.
bit 7	To mask out all 8-bits, set bits 7-4 to 1. When fewer than 8-bits are to be masked, set bit 7 = 0.

The bit mask is generated by the Sequencer as specified by the shift control field. A zero in the bit mask indicates the bit to be masked out. An x indicates a dont care bit value. Some examples of shift controls and bit masks:

Function	Shift Control	Bit Mask
Rotate left by 2	0 000 0 010	1111 1111
Shift left by 1	0 001 0 001	1111 1110
Shift left by 7	0 111 0 111	1000 0000
Shift left by greater than 7	1 111 0 xxx	0000 0000
Shift right by 1	0 001 1 111	0111 1111
Shift right by 7	0 111 1 001	0000 0001

Command Line Applications

Some specific applications of command lines are described below. These applications are useful when writing a compiler to translate higher-level program instructions to Sequencer command lines.

To clear a destination:
source = allzeros AND immediate = dont care → destination

To move an immediate > 0 to a destination:
source = allzeros OR immediate > 0 → destination

To clear carry flag:
source = allzeros ADD immediate = dont care → destination = none

To set carry and clear destination:
source = allones ADD immediate = 1 → destination

Simple branch without register or carry flag alteration:
ret, jz, jnz: source = sindex AND immediate = 0ffh
jmp, jc, jnc, call: source = sindex ORI immediate = 00h

When calling a subroutine, a constant or variable can be passed in via the source index register in the same command as the call:
variable pass-in: source = variable ORI immediate = 00h
constant pass-in: source = allzeros ORI immediate = constant

To move a source to a destination:
source AND immediate = 0ffh → destination

To exclusive OR a source and the accumulator:
source XOR immediate = 0 → destination

To exclusive OR a source and a constant > 0:
source XOR immediate = constant → destination

To exclusive OR a source and a constant = 0:

source AND immediate = 0ffh → destination

To compare a source with a second source:

source → accumulator
source XOR immediate = 0
If equal, zero flag = 1. If not equal, zero flag = 0.

To compare a source with a constant > 0:

immediate = 2s complement of constant
source ADD immediate
If equal, zero flag = 1. If not equal, zero flag = 0.
If source > = constant, carry flag = 1.
If source < constant, carry flag = 0.

To compare a source with a constant = 0:

source AND immediate = 0ffh
If equal, zero flag = 1. If not equal, zero flag = 0.

To test bits in the source:

source AND immediate = bit test mask
If bit(s) set, zero flag = 0. If bit not set, zero flag = 1.

> Note
> Bit test mask must have at least one bit set.

To add a source with a second source:

source → accumulator
source ADD immediate = 0 → destination

To add a source with an immediate > 0:

source ADD immediate → destination

To add a source with an immediate = 0:

00 → accumulator
source ADD immediate → destination

To add a 3-byte source with a 1-byte immediate > 0:

source,lo ADD immediate > 0 → destination,lo
00 → accumulator
source,mid ADC immediate = 0 → destination,mid
source,hi ADC immediate = 0 → destination,hi

To add a 3-byte source with a 1-byte immediate > 0:

source,lo → accumulator
source,lo ADD immediate = 0 → destination,lo
source,mid → accumulator
source,mid ADC immediate = 0 → destination,mid
source,hi → accumulator
source,hi ADC immediate = 0 → destination,hi

To subtract a 1-byte constant > 0 from a 3-byte source:

immediate = 2s complement of constant
source,lo ADD immediate → destination,lo
source = allzeros OR immediate = 0ffh → accumulator
source,mid ADC immediate = 0 → destination,mid
source,hi ADC immediate = 0 → destination,hi

To subtract a 3-byte source from a 3-byte source:

set carry
source,lo XOR immediate = 0ffh → accumulator
source,lo ADC immediate = 0 → destination,lo
source,mid XOR immediate = 0ffh → accumulator
source,mid ADC immediate = 0 → destination,mid
source,hi XOR immediate = 0ffh → accumulator
source,hi ADC immediate = 0 → destination,hi

❐

6

Design Notes

Unusual or difficult requirement details are included in this section.

I/O Decodes

Most registers will be accessible to both the Sequencer 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:

- Board ID (BID0-3), read or write by Host only without pause
- Board Control(BCTL), read or write by Host only without pause
- Host Control (HCNTRL), read or write by Host only without pause
- Clear Interrupt (CLRINT), write by Host only without pause
- Interrupt Status (INTSTAT), read by Host only without pause, concurrent write by Sequencer
- Error Status (ERROR), read by Host only
- Queue Out FIFO (QOUTFIFO), read by Host only without pause, concurrent write by Sequencer
- Queue Out Count (QOUTCNT), read by Host only without pause, concurrent write by Sequencer
- SINDIR is not usable by the Host
- DINDIR is not usable by the Host

❐

7

Test Features

Overview

This section is a description of the hardware test modes and features used in the AIC-7770 to facilitate production testing. It contains a description of the special hardware configurations designed into the AIC-7770. A complete list of the functional and other tests written for this purpose is contained in the *AIC-7770 Test Definition Document*.

The AIC-7770 consists of six basic sections: SCSI, Sequencer, Host, FIFO, SCB RAM, and Scratch RAM. The Host interface may be configured for either the EISA or ISA bus. All registers are available to the Host computer and to the Sequencer (except for two) 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 FIFO, SCB RAM, Scratch RAM, and Sequencer RAM all have a special test mode which allows the detection of missing transistors in the RAM cell which otherwise may not be discovered. These tests are described in a later section.

The AIC-7770 contains special circuitry to help test input levels and output current levels.

Most of the signals connect to busses 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	
3	
2	
1	CNTRTEST
0	CMODE

7-2	(0)	Not Used	Always reads 0.
1	(0)	CNTRTEST	When set to a one the SCSI transfer counter STCNT and the Selection Timeout 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 Timeout Timer (SELTIMER)

Type: R
Address: I-0358h, E-zC18h, C-18h

This register is used to monitor the state of the hardware Selection Timeout timer.

0358 SELTIMER	R/W
7	
6	
5	STAGE 6
4	STAGE 5
3	STAGE 4
2	STAGE 3
1	STAGE 2
0	STAGE 1

7-6	(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).

Test Chip (TESTCHIP)

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

These bits select certain sections of the chip for test purposes. This register should not be written during the normal operation as test logic will be activated that will disrupt the operation. TESTCHIP register is cleared to 0 when input RESDRV is asserted or POR is active except when the HIC INPUT PAD test has been selected. In this case input CHRDY/EXRDY must be de-asserted while RESDRV is asserted to clear TESTCHIP register to 0.

TESTCHIP	R/W
7	TESTSEL2
6	TESTSEL1
5	TESTSEL0
4	TESTRAM
3	TESTHOST
2	TESTSEQ
1	TESTFIFO
0	TESTSCSI

7:5	(0)	TESTSEL(2:0)(2:0)	Used to select hardware sub-test features within a major cell. See Testchip register bits TEST(4:0)(4:0) for usage. TEST(2:0)(2:0) states only have meaning when one TESTCHIP register bit (4:0)(4:0) is in the active state.
4	(0)	TESTRAM	TESTRAM when set enables blocks that contain Adaptec RAM cells to utilize the Adaptec RAM test feature to verify that no weak or defective storage locations exist in the RAM cells. Currently the following associated blocks contain RAM to be tested: DFIFO, SCB Array, Scratch RAM and the Sequencer.
3	(0)	TESTHOST	Select the Host block for testing. HIC provides the following hardware test features for testing HIC counters, HIC timers, HIC input PADs, HIC output PADs, un-interlock STARTI-/CMDI- for IO and enable monitoring of the chip CSDAT(7:0) bus. These features are only active when the TESTHOST bit is in the active state. Note, to provide access to monitor the Adaptec HIC test features chip output pads LA(23:17)(23:17) and SA/LA16 are redefined from normal ISA/EISA definition. This redefinition may only be performed in a test environment where the redefinition will not affect system operation. TESTHOST is placed in the active state (=1) by the following non-normal input pin states by the part tester (de-assert input CHRDY / EXRDY then perform a write I/O cycle to the TESTCHIP register with TESTHOST asserted).

TEST HOST	TEST SEL 2	1	0	HTEST DECODE	USE of LA(23:17)(23:17) and SA/LA16
0	x	x	x	--	Normal operation of address outputs
1	0	0	0	0	CSDAT monitor, SEQ ALU and DPRAM
1	0	0	1	1	HIC input pad test or process check oscillator
1	0	1	0	2	HIC output pad voltage high
1	0	1	1	3	HIC output pad voltage low
1	1	0	0	4	HCNT counter (increment, carry, decode)
1	1	0	1	5	HADDR counter (increment, carry, decode), timers (BONT, BOFFT, BSPDT)
1	1	1	0	6	SEQ RAM parity
1	1	1	1	7	SEQ RAM data
1	x	x	x	1-7	LA(23:17)(23:17), SA/LA16 output selected internal signals instead of address.
1	x	x	x	1-7	STARTI- / CMDI- interlock is disabled (enables modified tester EISA IO).

Normal operation: Outputs LA(23:17)(23:17) and SA/LA16 output memory address data while HIC is in Bus Master state.

CSDAT(7:0) monitoring: Outputs LA(23:17), SA/LA 16 normal output replaced with CSDAT(7:0) current state.

CSDAT monitoring is active anytime TESTHOST is set. The CSDAT(7:0) bit relationship to LA(23:17), SALA 16 is as follows:

LA, SA/LA	CSDAT
23	7
22	6
21	5
20	4
19	3
18	2
17	1
16	0

2 (0) TESTSEQ — When active (=1) selects the Sequencer cell for testing. See Sequencer Tests for more information.

1 (0) TESTFIFO

Select the DFIFO block for testing. When TESTFIFO is active CIOBUS writes to DFDAT will cause 32-bit writes to occur in the DFIFO with the 8-bit data value on the CSDAT bus written in parallel to each byte of the 32-bit DFIFO location pointed to by DFWADDR value. The DFWADDR value will increment following each write. This test feature shortens the DFIFO RAM test data load time for doing RAM stress tests or for writing a constant data value to all RAM locations.

> Note
> DFDAT must be enabled for writing from the CIOBUS or an illegal address error will occur (ref DFCNTRL register).
>
> Reading DFDAT thru the CIOBUS remains at 8-bits per read.

0 (0) TESTSCSI

Select the SCSI block for testing.

> Note
> When TESTSCSI is active external SCSI bus pins may be redefined for test purposes (ref TESTCHIP register values 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) by the following non-normal input pin states by the part tester (assert input CHRDY/EXRDY then perform a write I/O cycle to the TESTCHIP register with D0 asserted). (See SCSI specification for pin information and additional tests.)
>
> The SCSI block also has a test register within it's own block that operates independently of TESTCHIP register.

Test Group Description

The following sections explain in greater detail about each of the test groups and configurations. Below is a table explaining the defined values which are loaded into TESTCHIP 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.

Data (7:0)	HTEST	REF*	CSDAT Monitor	Description
00	-	---	N	Normal operation.
02	-	ad	N	Enable DFIFO double-word write.
04	-	bd	N	Sequencer ALU output monitor test.
10	-	ad	N	Dual port RAM stress test (Scratch, SCB, DFIFO).
12	-	ad	N	Dual port RAM stress test with enable of DFIFO double-word write.
14	-	bd	N	Sequencer single port RAM stress test.
D4	-	bd	N	Sequencer RAM parity test.
F4	-	bd	N	Sequencer RAM data test.
08	0	a	Y	CSDAT monitor only test (no other TESTCHIP test feature active).
0C	0	ab	Y	Sequencer ALU output monitor test.
18	0	a	Y	Dual port RAM stress test (Scratch, SCB, DFIFO).
1A	0	a	Y	Dual port RAM stress test with enable of DFIFO double-word write.
1C	0	b	Y	Sequencer single port RAM stress test.
29	1	ac	Y	Input PAD test (HIC, SCSI).
2B	1	ac	Y	Process check oscillator test.
2C	1	ab	Y	Monitor illegal addresses test.
49	2	ac	Y	Output PAD high-voltage test (HIC, SCSI).
69	3	ac	Y	Output PAD low-voltage test (HIC, SCSI).
88	4	a	Y	HIC HCNT counter and decode test.
A8	5	a	Y	HIC HADDR counter, decode test and BONT, BOFFT, BSPDT timers.
DC	6	b	Y	Sequencer RAM parity test.
FC	7	b	Y	Sequencer RAM data test.

*a = HIC
b = Sequencer
c = SCSI
d = Test modes that are also usable in normal system environment.

Input Pad Testing

Input Pad Test

When an I/O write is performed to place HIC in this test mode, eight input pad ANDed strings are connected to output pads LA(23:17), SA/LA16 for input threshold testing. This allows eight inputs (one from each string) to be tested at the same time. When all inputs in a string are at an input high level the associated output will also be at an output high level. Placing any one input at an input low level will cause the associated output to also go to an output low level. While in HIC INPUT PAD test selection the reset of the TESTCHIP register caused by RESDRV has a modified action to allow testing of the RESDRV input and at the same time prevent clearing of the INPUT PAD test selection. This requires that input CHRDY / EXRDY be de-asserted before testing input RESTDRV. Also while in INPUT PAD test the input RESDRV is inhibited from affecting the state of it's added string until input CHRDY / EXRDY is de-asserted. SANDTO is the added input string associated with the SCSI logic block. An IO write or RESDRV may be used to exit from this test mode.

ANDed Input Pad Strings List	
LA23	CLK, SEL1/EX32-, IORC-/START-, RESDRV, IOWC-/CMD-, D5, D6, D7
LA22	SEL2/MBURST-, D0, D1, D2, D3, D4, SEL0/SBURST-, BRESET, BBSY, BSEL, BREQ, BMSG, BIO, BCD, BATN, BACK, BSCD0, BSCD1, BSCD2, BSCD3, BSCD4, BSCD5, BSCD6, BSCD7, BSCDP, ARESET, ABSY, ASEL, AREQ, AMSG, AIO, ACD, AATN, AACK, ASCD0, ASCD1, ASCD2, ASCD3, ASCD4, ASCD5, ASCD6, ASCD7, ASCDP
LA21	SA0/BE0-, SA1/BE1-, M16-/BE2-, CHRDY/EXRDY, BCLK, AEN, MWTC-/WR, MRDC-/MIO
LA20	SA8/LA8, SA9/LA9, SA10/LA10, SA11/LA11, SA12/LA12, SA13/LA13, SA14/LA14, SA15/LA15
LA19	D17, D16, SA2/LA2, SA3/LA3, SA4/LA4, SA5/LA5, SA6/LA6, SA7/LA7
LA18	D25, D24, D23, D22, D21, D20, D19, D18
LA17	D9, D8, D31, D30, D29, D28, D27, D26
SA16/LA16	ISAEISA, DACK/MAK-, D15, D14, D13, D12, D11, D10

Output Pad Testing

Output Pad High Voltage Test

When an I/O write is performed to place HIC in this test selection all output pads are enabled and controlled to place the outputs at an output high level following with the rising edge of input CHRDY/EXRDY after completion of the write to the TESTCHIP register.

> 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.
>
> I/O cycles are prohibited after the I/O write that placed HIC in this selection and assertion of RESDRV is required to exit from this test selection.

Output Pad Low Voltage Test

When an I/O write is performed to place HIC in this test selection all output pads are enabled and controlled to place the outputs at an output low level following with the rising edge of input CHRDY/EXRDY after completion of the I/O write to the TESTCHIP register.

> 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.
>
> I/O cycles are prohibited after the I/O write that placed HIC in this selection and assertion of RESDRV is required to exit from this test selection.

Process Check Oscillator

When an I/O write is performed to place HIC in this test selection the HIC located die will configure the input pad ANDed input strings (ref HIC 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 101 cell levels located over the die along with long metal interconnects. An output from the ring (reference to LA22 ANDed string) is muxed into the HADDR(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 HADDR(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 when it asserts CHRDY/EXRDY following the I/O write that entered this test mode with all other inputs (except RESDRV input) asserted as in INPUT PAD test. After a suitable delay (expected clock rate range is 1 - 4 usec) de-assert CHRDY/EXRDY to pause (internally synchronized) the ring oscillator and read the HADDR(3:0) value to determine the die speed value. The input ANDed string sections of the ring oscillator are

interconnected as shown below with each string output available on LA(23:17), LA/SA16 same as in INPUT PAD test. While PROCESS CHECK OSCILLATOR test is selected inputs RESDRV, BCLK and CLKIN in the assigned input strings are forced to appear asserted allowing RESDRV to be kept de-asserted while CHRDY / EXRDY input is asserted to enable the ring oscillator to free run. An IO write or RESDRV may be used to exit from this test mode.

Start of ring loop: string ASCDP through BRESET, to string SEL0SBURST through SEL2MBURST, to string DI7 through CLKIN, to string DI10 through ISAEISA , to string DI26 through DI9, to string DI18 through I25, to string SLAI7 through DI17, to string SLAI15 through SLAI8, to string MRDCMIOI through SABEI0, return to start of ring loop. The ring osc output to the HADDR counters is taken from SABEI0.

Monitor Illegal Address

When an I/O write is performed to place HIC in this test selection the outputs from the illegal I/O decode checking logic will be enabled to outputs LA(23:17) and LA/SALA16 as follows to shorten the test vector requirement for this logic. Note while this test is selected the ERROR register is held in a reset state. Pause is not forced active when an illegal address is accessed to allow the SEQ to cycle all illegal addresses without being stopped by PAUSE and the need to recover for each illegal address. This modifies the actions indicated in the Interrupt status summary table. An IO write or RESDRV may be used to exit from this test mode.

LA23:19:	Not used = 0
LA18:	Illhadr
LA17:	Illsaddr
SA16/LA16:	Illopcode

Host Tests

The Host Interface Cell (HIC) provides the following hardware test features for testing HIC counters, HIC timers, and monitoring of the chip CSDAT(7:0) BUS. These features are only active when the TESTHOST bit is set to the active state. Note that output pads LA(23:17) and LA16 are redefined from normal ISA/EISA definition when monitoring the CSDAT(7:0) bus. This reassignment may only be performed in a test environment where it will not affect system operation.

TESTHOST may only be placed in the active state by following a non-normal write sequence by the part tester. First, de-assert input CHRDY/EXRDY then perform an I/O write cycle to the TESTCHIP register with D3 asserted and the required values on D[7:5] for the desired HIC test. A write to the TESTCHIP register with EXRDY asserted will cause TESTHOST to become inactive disregarding the value on D3.

CSDAT(7:0) Monitoring

Outputs LA(23:17) and SA/LA16 normal output replaced with CSDAT(7:0) with the rest of the chip operation continuing in the normal state. 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	Pin Name	Signal Name
LA23	CSDAT7	LA19	CSDAT3
LA22	CSDAT6	LA18	CSDAT2
LA21	CSDAT5	LA17	CSDAT1
LA20	CSDAT4	LA16	CSDAT0

Host HCNT(2:0) and Decode Test

When an I/O write is performed to place HIC in this test selection, the normal outputs of LA(23:17), SA/LA16 and IRQ are replaced with the following logic outputs to allow testing of their logic with a smaller test vector set. The carry in logic between bytes and the logic which normally controls counting by 1, 2, 3, 4 in ISA/EISA Bus Master operation are not tested in this test selection. Halting HCLKM and performing I/O reads to the HCNT(2:0) will allow access to the counter values. An IO write or RESDRV may be used to exit from this test mode.

Pin Name	Signal Name	Pin Name	Signal Name
LA23	HCNTGTF decode	LA18	HCNTLTTGZ decode
LA22	HCNTF decode	LA17	HCNT carry out 1
LA21	HCNTLTFGZ decode	SA/LA16	HCNT carry out 0
LA20	HCNTGTT decode	IRQ	HCNTZ- decode
LA19	HCNT T decode		

HIC HADDR(3:0) and Decode Test

When this test is selected the HADDR address counter is segmented into bytes which are incremented with each rising edge of HCLKM (every other CLKIN). The carry in logic between bytes and the logic which normally controls counting by 1, 2, 3, 4 in ISA/EISA Bus Master operation are not tested in this test selection. Halting HCLKM and performing I/O reads to HADDR(3:0) will allow access to the counter values. Timers BONT, BOFFT and BSPDT are clocked by HCLKM in parallel. When the selected count values of BONT and BOFFT have been reached they will change state but will not recycle and need to be restarted by de-selecting HIC HADDR test selection, then changing to the next BONT and BOFFT selections then re-selecting HIC HADDR test. Timer BSPDT will continue to cycle while HIC HADDR test is active and IADLDEN- (ISA address and data load enable) will cycle with BSPDT. IADLDEN- timing changes with BSPDT selections (ref BUSSPD register). An IO write or RESDRV may be used to exit from this test mode.

Pin Name	Signal Name	Pin Name	Signal Name
LA23	BONT time selections	LA18	HADDR carry out 2
LA22	BOFFT time selections	LA17	HADDR carry out 1
LA21	BSPDT time selections	SA/LA16	HADDR carry out 0
LA20	VAR decode memory range	IRQ	IADLDEN- decode
LA19	OKB decode memory range		

Sequencer Tests

The Sequencer RAM is also tested. After basic operation, a stress test may also be run. This will test for missing transistors in the RAM cell. The Sequencer RAM also has parity bits internal to the cell which also are tested for proper operation.

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

Data is written to the Sequencer RAM and read back. The parity bit is available on CSDAT(0).

Sequencer RAM Data

Various patterns are written to the Sequencer RAM.

Sequencer RAM Leakage

A special test sequence is performed to test the RAM cell for missing transistors. An address is presented through the I/O write logic, and the write strobe is held in place by the tester. Data is presented on D(7:0) and then is complemented. CSDAT(0) shows the testout pin from the Sequencer RAM and will be driven low Xns after the data change and will go high again at least 145ns later. CSDAT(0) should be sampled low Xns after the data is changed, and sampled high Xns after the data change.

SCSI Tests

SCSI Counter Tests

The Selection Timeout counter may be tested by setting CMODE and CNTRTEST (bits 0,1, SCSITEST) and clocking the input clock. CNTRTEST clocks SELTIMER, SHADDR, and STCNT at the input clock rate. CMODE forces each stage to count with every clock. Proper functionality may be determined by examining the counts after each clock. The contents of SELTIMER may be examined for proper carry operation in all stages of the counter.

SCSI Data Path Tests

The data path from DFIFO to SFIFO and back again may be exercised by the proper sequencing of SDMAEN and SCSIEN.

FIFO Test

FIFO RAM Test

The FIFO test may be shortened by setting TESTFIFO in TESTCHIP. When set, a write to a FIFO location will store the data in four locations at one time. This can be done since the FIFO is arranged in 4-byte wide architecture. This feature is used to shorten RAM tests:

- Writing to DFDAT register will write the data value on CDDAT(7:0) to the four bytes pointed to by the bits (5:0) in the DFWADDR0 register. DFWADDR0 will increment following each write operation to minimize the I/O operations to perform RAM testing (see RAM tests for a complete description) with four bytes tested in parallel to minimize total test time. Each RAM byte is read for test verification as in normal operation.

- Reading DFWRADDR0 will access DFWADDR0 bits (5:0) as normal but will also access on bit 6 the state of DFWADDR0 bit 6 (used in DFIFO full/empty status generation) instead of a constant 0 value as in normal operation), and on bit 7 will access DFIFO RAM TESTLOCK state instead of a constant 0.

- Reading DFRADDR0 will access DFRADDR0 bits (5:0) as normal but will also access on bit 6 the state of DFRADDR0 bit 6 (used in DFIFO full/empty status generation) instead of a constant 0, and on bit 7 the state of DFIFO status DFSDH instead of a constant 0.

RAM Tests

There are two types of RAM in the AIC-7770, the Sequencer RAM and a Dual Port RAM used in various configurations. The Dual Port RAM is used in the FIFO, the Scratch RAM, and the SCB RAM. Both types of RAM have a special stress test which checks for missing transistors which may not otherwise be detected. These tests operate differently because of the different constructs of the RAM cells. The Sequencer RAM test is described in the Sequencer tests. All other RAMs use the following procedure for the stress test. Normal data pattern tests are run in addition to this test.

Dual Port RAM Stress Test

The following steps should be taken to test the Dual Port RAM. This procedure may be run at wafer and package levels or in the system at the diagnostic level:

➤ 1. With TESTRAM cleared, pre-load (write) a test data value to a RAM location. This a normal RAM write operation.

2. Place TESTRAM in the active state (set).

3. Write to the same RAM address the second time with the same data. Writing to a RAM address with TEST4 active will cause the CIOBUS logic in the cell containing the RAM to enter a RAM TESTLOCK state and internally latch the supplied CIOBUS CDDAT(7:0) data, CDADR(7:0)- address and (CDWEN- (active) to keep the RAM WE- input active) to start a stretched write operation. RAM TESTLOCK state also places the RAM cell TEST- input in the active state. Adaptec RAM cells when accessed in this manner will stress the addressed RAM location storage capability with the opposite value that is being supplied (and previously pre-loaded). The recommended minimum stress time period is 500ns.

4. 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 cell containing the RAM is disconnected from the destination side of the CIOBUS with normal operation continuing on the source side of the CIOBUS for all locations. Normal operation also continues on the destination side of the CIOBUS except for cells that are in the RAM TESTLOCK state.

5. Clearing TESTRAM will clear all RAM TESTLOCK states and return those cells to normal operation on the destination side of the CIOBUS. Multiple RAM(s) may have a RAM location stressed at the same time to shorten the overall test time. 8-bit RAM locations may be verified directly with the CSDAT bus and RAM that is wider than eight bits will need additional logic to provide the compare function, with the compare output status accessed by the CSDAT bus.

❒

8

Electrical Information

Absolute Maximum Ratings

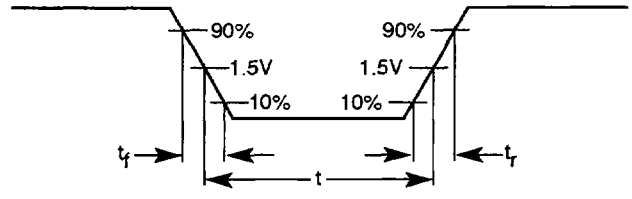
Parameter	Range
Storage Temperature	-55 to 125 degrees C
Power Supply Voltage	0 to 7 Volts
Voltage on any pin	-0.5 to VCC+0.5 Volts

Operating/Test Conditions

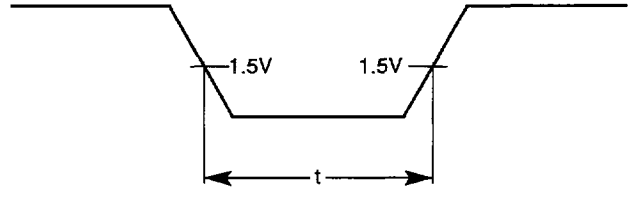
Parameter	Range
Ambient Temperature	0 to 70 degrees C
Supply Voltage	4.75 to 5.25 Volts
Supply Current:	
Active	50 mA
Paused	36 mA
Power-down	5 mA
Input Static Discharge	2000 Volts
tf	<5 ns
tr	<5 ns
CL	50 pf unless otherwise noted

Switching Waveforms

INPUT LEVELS

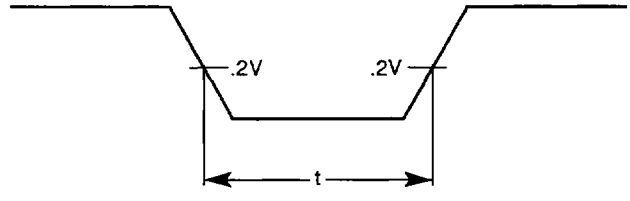


OUPUT LEVELS



C = 50 pf unless otherwise noted

OPEN DRAIN OUPUT LEVELS



C = 50 pf unless otherwise noted

Figure 8-1. Switching Waveforms

DC Parameters

Ta = 0 to 70 degrees C, VCC = 5 Volts +/- 5%, GND = 0 Volts

Symbol	Description	Min	Max	Units	Test Condition
Iil	Input Leakage		10	uA	Vin = .4 to VCC
Iol2	Output Leakage, SCSI open drain outputs		50	uA	Vout = .5 to VCC
Iol	Output Leakage, all outputs except SCSI		10	uA	Vout = .5 to VCC
Vih	Input High Voltage	2.0		Volts	
Vil	Input Low Voltage		.8	Volts	
Vihys	Input hysteresis: RESDRV, MASTER16-, NOWS-, D(31:0), LA(15:2), START-, BE(2:0)-, MIO, WR, MSBURST-, ASCD(7:0), ASCDP, ACD, AIO, AMSG, AREQ, AACK, ARESET, ASEL, ABSY, AATN, BSCD(7:0), BSCDP, BCD, BIO, BMSG, BREQ, BACK, BRESET, BSEL, BBSY, BATN	.2		Volts	
Voh	Output High Voltage	2.4		Volts	Io = -400uA
Vol1	Output Low Voltage: IRQ, ADIFFDAT(3:0), ADIFFADR(1:0), ADIFFSTRB		.5	Volts	Io = 4mA
Vol2	Output Low Voltage: MASTER16-, NOWS-, D(31:0), LA(31:24)-, LA(23:2), MREQ-, START-, BE(3:0)-, MIO, WR, MSBURST-		.5	Volts	Io = 24mA
Vol3	Output Low Voltage: ASCD(7:0), ASCDP, ACD, AIO, AMSG, AREQ, AACK, ARESET, ASEL, ABSY, AATN, BSCD(7:0), BSCDP, BCD, BIO, BMSG, BREQ, BACK, BRESET, BSEL, BBSY, BATN		.5	Volts	Io = 48mA
Cin	SCSI signals		10pf		Fc=1MHz
ICC_1	Supply Current at Full Operation		50	mA	25° C ambient temp.
Icc_2	Clock Enabled (Sequencer Paused)	59	68	uA	25° C ambient temp.
Icc_3	Powerdown Clock Degated		2	uA	25° C ambient temp.

Signal Test Loads

Pull up resistors are connected to VCC. Pull down resistors are connected to GND.

Signals	Circuit Values		
BRESET, BBSY, BSEL, BREQ, BMSG, BIO, BCD, BATN, BACK, BSCD0, BSCD1, BSCD2, BSCD3, BSCD4, BSCD5, BSCD6, BSCD7, BSCDP, ARESET, ABSY, ASEL, AREQ, AMSG, AIO, ACD, AATN, AACK, ASCD0, ASCD1, ASCD2, ASCD3, ASCD4, ASCD5, ASCD6, ASCD7, ASCDP	Capacitance Load 1	300	pf
	R1	402	Ohm
DRQ, IRQ	Capacitance Load 2	120	pf
	R1	402	Ohm
	R2	1300	Ohm
MREQ	Capacitance Load 2	120	pf
	R1	806	Ohm
	R2	4020	Ohm
D(31:0), START-, MSBURST-, LA(23:2), BE(3:0), MIO, WR, MASTER16-, NOWS-, LA(31:24)-	Capacitance Load 2	240	pf
	R1	402	Ohm
	R2	1300	Ohm
ADIFFDAT(3:0), ADIFFADR(1:0), ADIFFSTRB	Capacitance Load 2	50	pf
	R1	806	Ohm
	R2	4020	Ohm

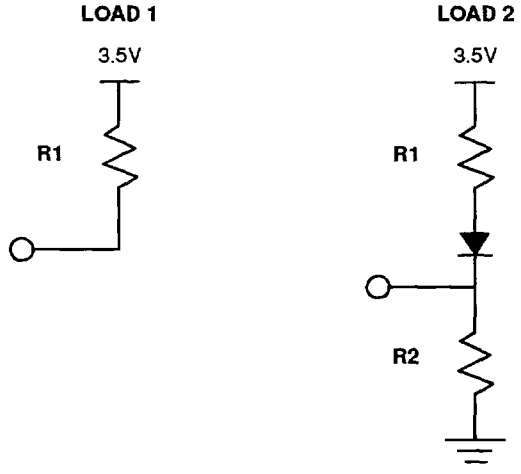


Figure 8-2. Signal Test Loads

9

System Timing

Clock Timing

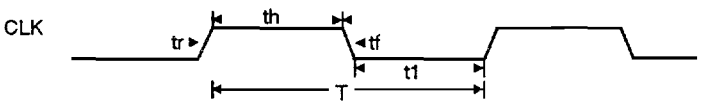


Figure 9-1. Clock Parameters

Parameter	Description	Time (ns)	
		min	max
th	Clock high time	7	
t1	Clock low time	8	
tr	Clock rise time		3
tf	Clock fall time		3
T	Clock period (nominal)	25	

SCSI Bus Timing

SCSI Data Transfers

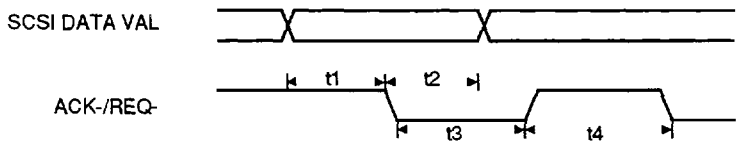


Figure 9-2. SCSI Data Transfers

Parameter	Description	Time (ns)	
		min	max
t1	SCSI Data setup to ACK/REQ active		
	Normal timing	55	
	Fast timing	25	
t2	SCSI Data hold from ACK/REQ inactive		
	Normal timing	95	
	Fast timing	35	
t3	ACK-/REQ- Pulse Width		
	Normal timing	90	
	Fast timing	30	
t4	ACK-/REQ- Negation Period		
	Normal timing	90	
	Fast timing	30	

EISA Master Bus Timing

EISA Arbitration

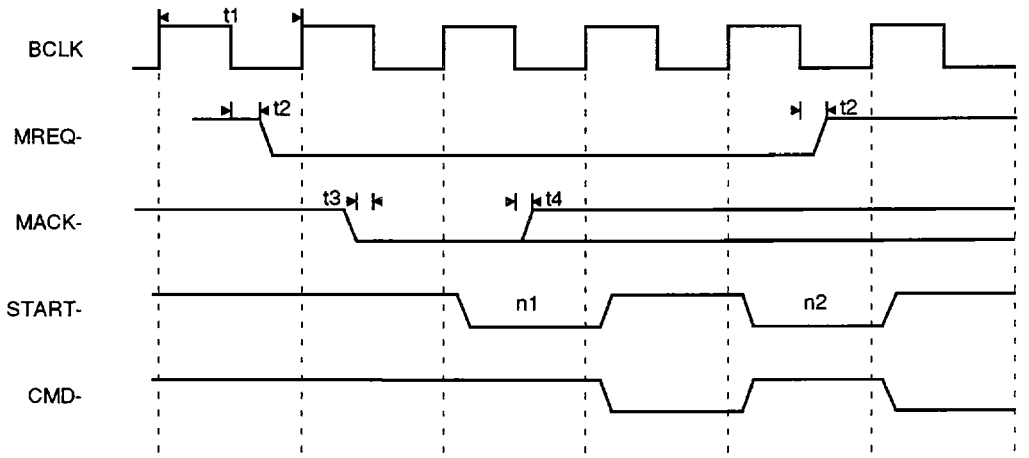


Figure 9-3. EISA Arbitration

Parameter	Description	Time (ns)	
		min	max
t1	BCLK period	120	
t2	MREQ- hold from BCLK falling	2	33
t3	MACK- setup BCLK falling	10	
t4	MACK- hold from BCLK falling	25	
n1	First START		
n2	Last START		

EISA Arbitration Burst Transfer

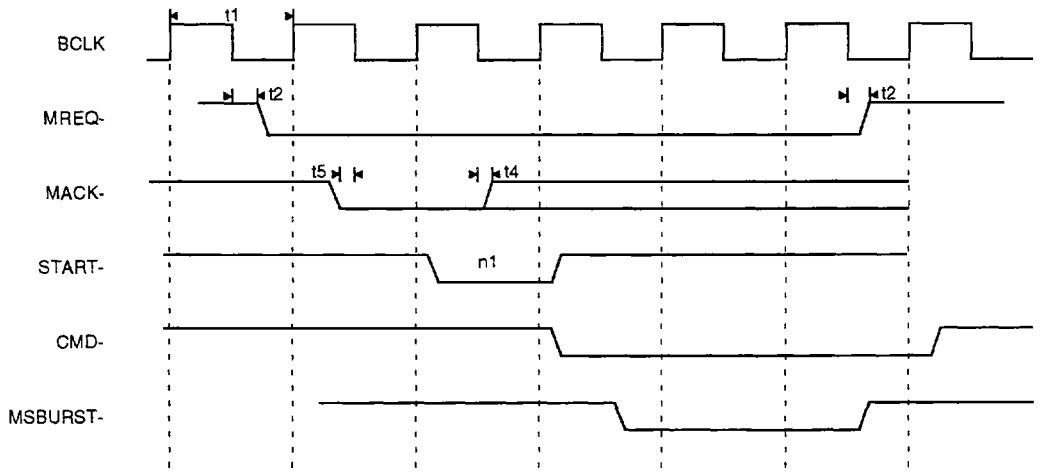


Figure 9-4. EISA Arbitration Burst Transfer

Parameter	Description	Time (ns)	
		min	max
t1	BCLK Period	120	
t2	MREQ- hold from BCLK falling	2	33
t3	MACK- setup to BCLK falling	10	
t4	MACK- hold from BCLK falling	25	
n1	First START		

EISA Arbitration Downshift Burst

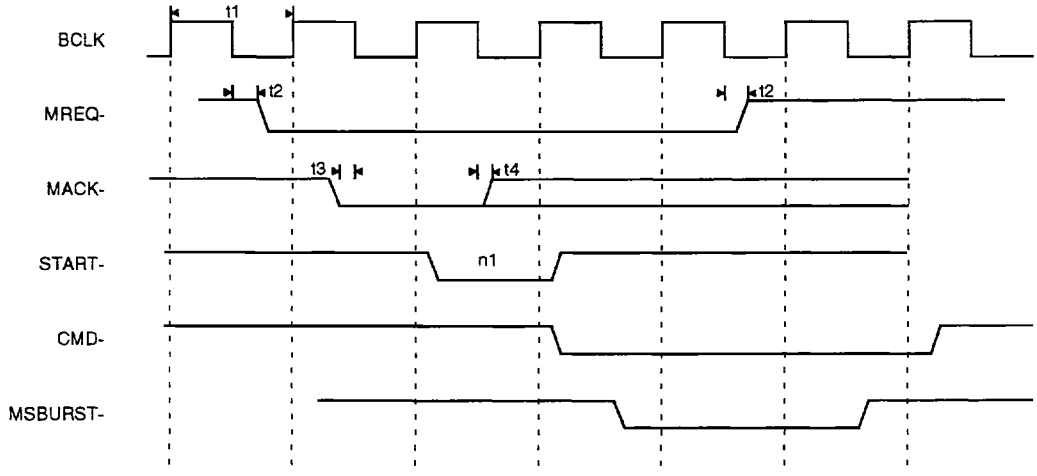


Figure 9-5. EISA Arbitration Downshift Burst

Parameter	Description	Time (ns)	
		min	max
t1	BCLK period	120	
t2	MREQ- hold from BCLK falling	2	33
t3	MACK- setup to BCLK falling	10	
t4	MACK- hold from BCLK falling	25	
n1	First START		

EISA Two Cycle Transfer – 32-bit

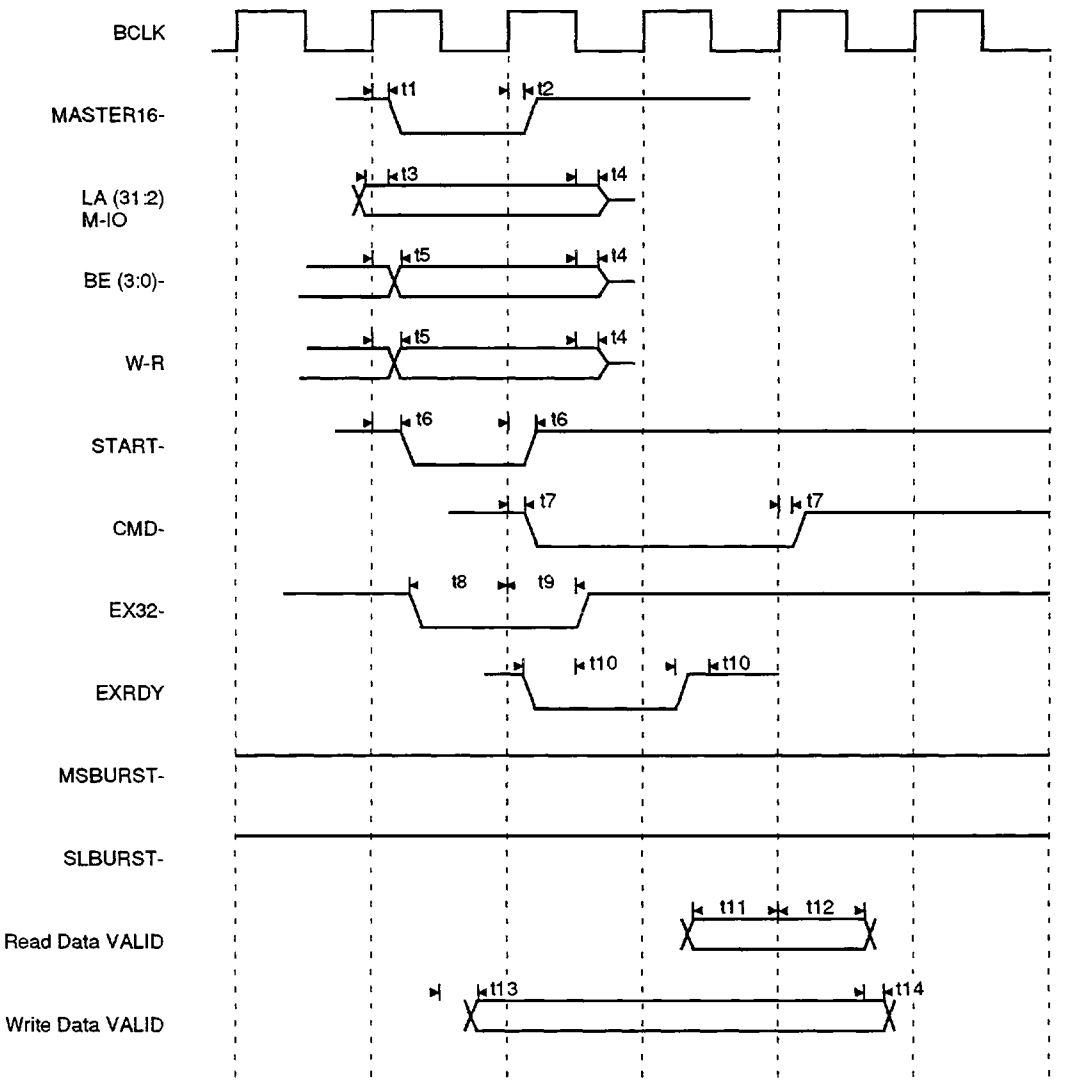


Figure 9-6. EISA Two Cycle Transfer – 32-bit

Parameter	Description	Time (ns)	
		min	max
t1	MASTER16- assert delay from BCLK rising	2	50
t2	MASTER16- float delay from BCLK rising	2	40
t3	LA, M-IO setup to START-	10	
t4	LA, M-IO float delay from BCLK falling	2	50
t5	BE-, W-R delay from BCLK rising	25	
t6	START- delay from BCLK rising	2	25
t7	CMD- delay from BCLK rising	2	25
t8	EX32- setup to BCLK rising	25	
t9	EX32- hold from BCLK rising	55	
t10	EXRDY setup to BCLK falling	15	
t11	Read data setup to BCLK rising	15	
t12	Read data hold from BCLK rising	4	
t13	Write data delay from BCLK falling	2	40
t14	Write data float delay from BCLK falling	2	50

EISA Burst Transfer – 32-bit Burst

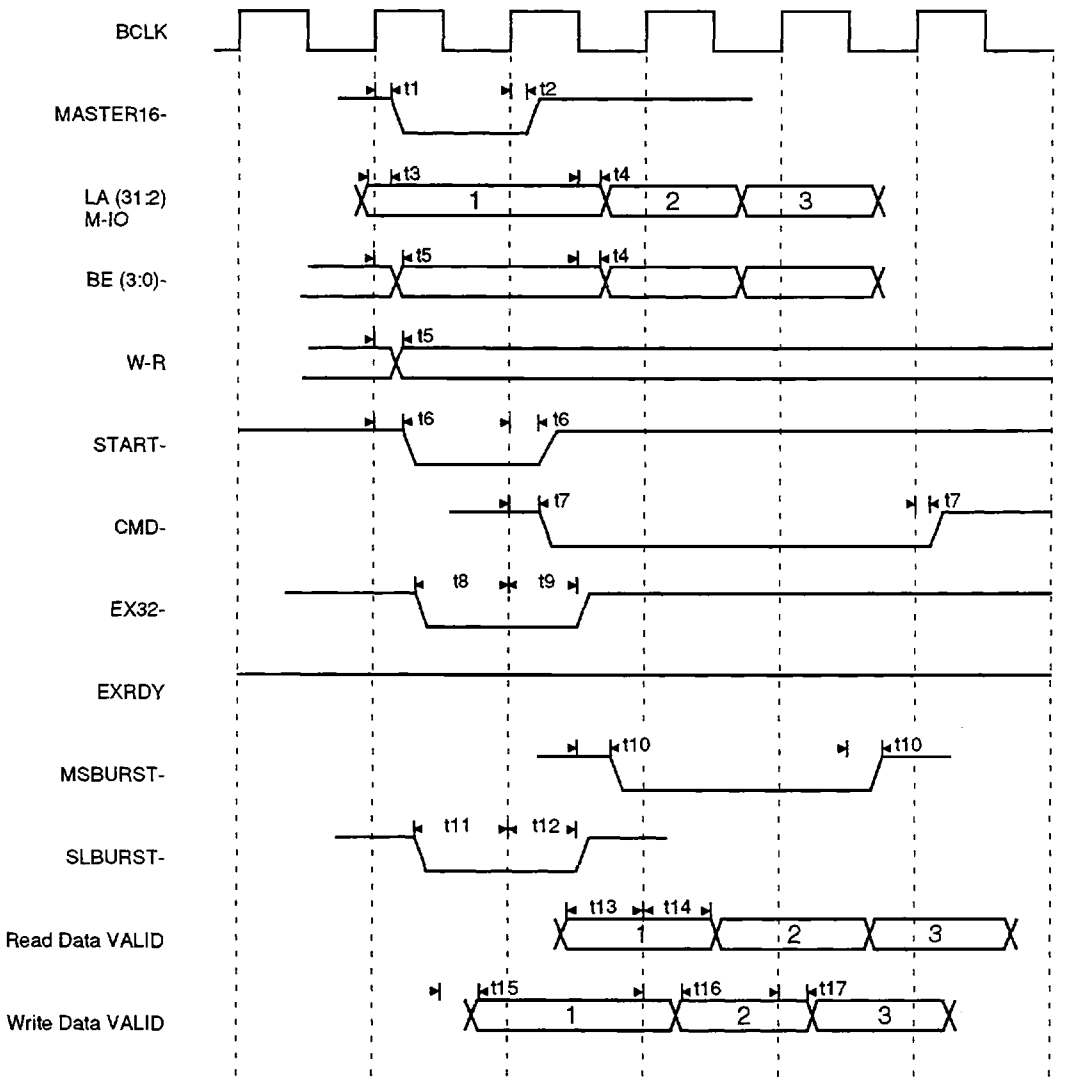


Figure 9-7. EISA Burst Transfer – 32-bit Burst

Parameter	Description	Time (ns)	
		min	max
t1	MASTER16- assert delay from BCLK rising	2	50
t2	MASTER16- float delay from BCLK rising	2	40
t3	LA, M-IO setup to START-	10	
t4	LA, BE- delay from BCLK falling	2	45
t5	BE-, W-R delay from BCLK rising		25
t6	START- delay from BCLK rising	2	25
t7	CMD- delay from BCLK rising	2	25
t8	EX32- setup to BCLK rising	25	
t9	EX32- hold from BCLK rising	55	
t10	MSBURST- delay from BCLK falling	2	35
t11	SLBURST- setup to BCLK rising	15	
t12	SLBURST- hold from BCLK rising	25	
t13	Read data setup to BCLK rising	15	
t14	Read data hold from BCLK rising	5	
t15	Write data delay from BCLK falling	2	40
t16	Write data delay from BCLK rising	5	40
t17	Write data hold from BCLK rising	5	

EISA Burst Transfer – 16-bit Downshift (No System Copy)

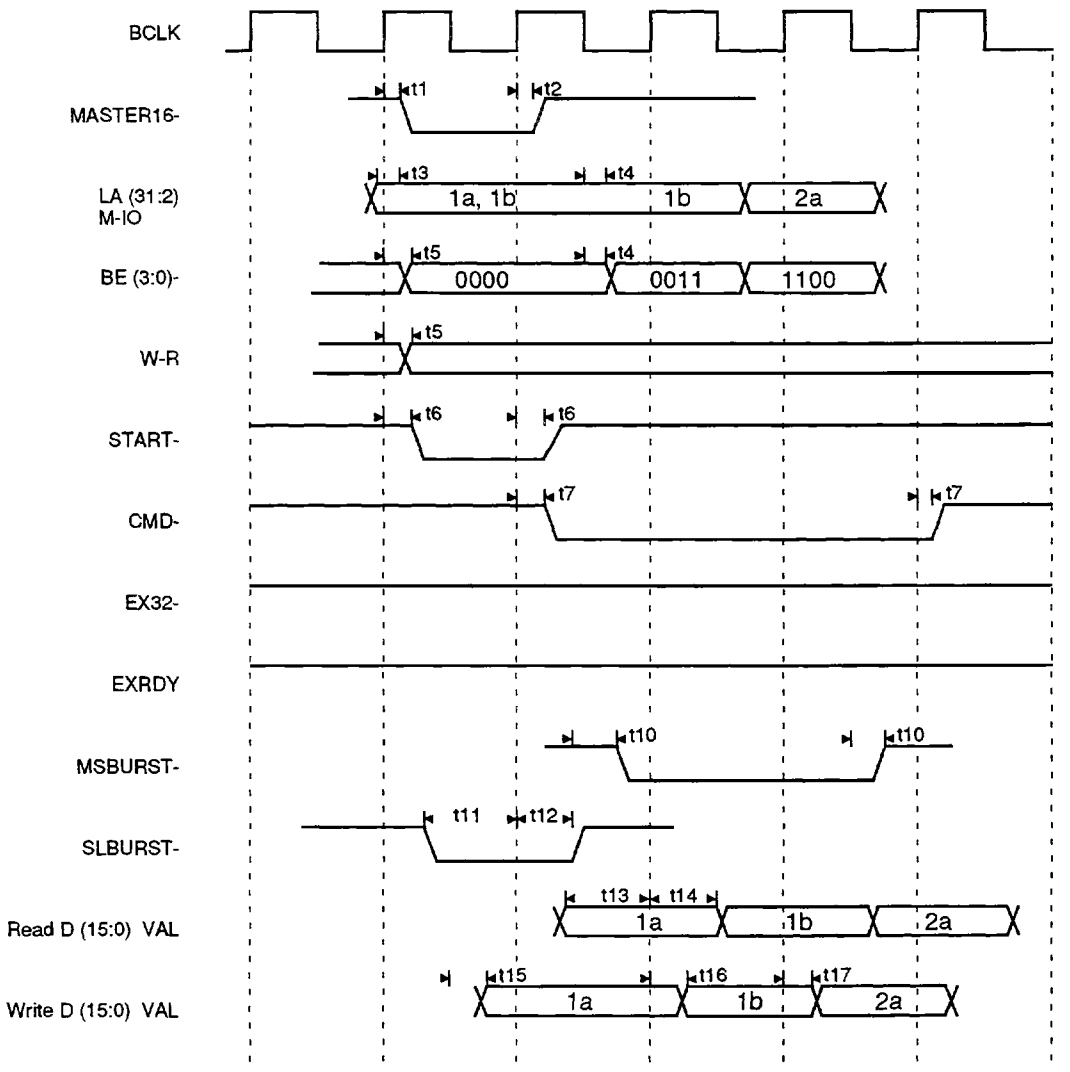


Figure 9-8. EISA Burst Transfer – 16-bit Downshift (No System Copy)

Parameter	Description	Time (ns)	
		min	max
t1	MASTER16- assert delay from BCLK rising	2	50
t2	MASTER16- float delay from BCLK rising	2	40
t3	LA, M-IO setup to START-	10	
t4	LA, BE- delay from BCLK falling	2	45
t5	BE-, W-R delay from BCLK rising		25
t6	START- delay from BCLK rising	2	25
t7	CMD- delay from BCLK rising	2	25
t8	EX32- setup to BCLK rising	25	
t9	EX32- hold from BCLK rising	55	
t10	MSBURST- delay from BCLK falling	2	35
t11	SLBURST- setup to BCLK rising	15	
t12	SLBURST- hold from BCLK rising	25	
t13	Read data setup to BCLK rising	15	
t14	Read data hold from BCLK rising	5	
t15	Write data delay from BCLK falling	2	40
t16	Write data delay from BCLK rising	5	40
t17	Write data hold from BCLK rising	5	

EISA Burst Transfer – 16-bit Downshift (System Copy)

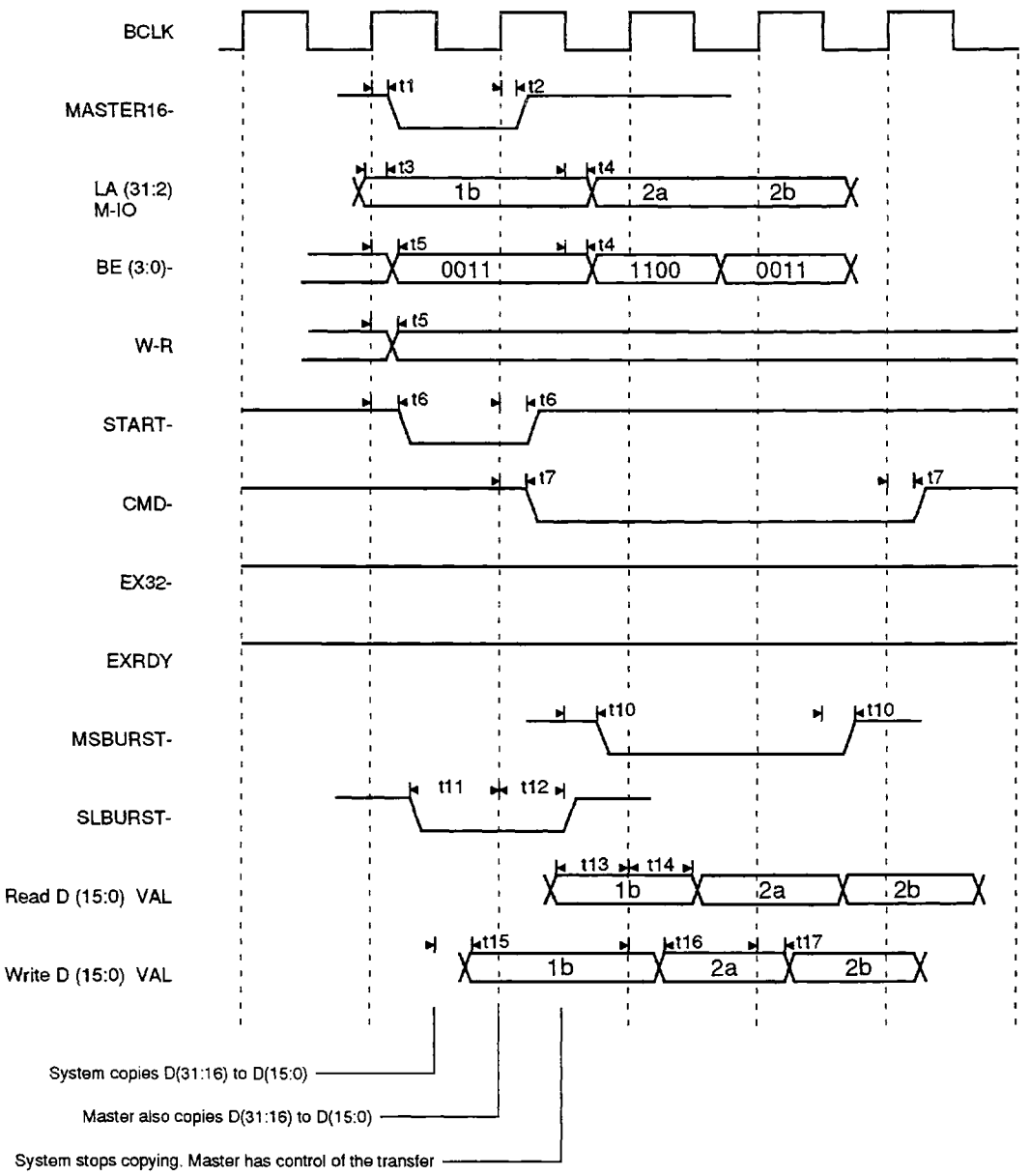


Figure 9-9. EISA Burst Transfer – 16-bit Downshift (System Copy)

Parameter	Description	Time (ns)	
		min	max
t1	MASTER16- assert delay from BCLK rising	2	50
t2	MASTER16- float delay from BCLK rising	2	40
t3	LA, M-IO setup to START-	10	
t4	LA, BE- delay from BCLK falling	2	45
t5	BE-, W-R delay from BCLK rising		25
t6	START- delay from BCLK rising	2	25
t7	CMD- delay from BCLK rising	2	25
t8	EX32- setup to BCLK rising	25	
t9	EX32- hold from BCLK rising	55	
t10	MSBURST- delay from BCLK falling	2	35
t11	SLBURST- setup to BCLK rising	15	
t12	SLBURST- hold from BCLK rising	25	
t13	Read data setup to BCLK rising	15	
t14	Read data hold from BCLK rising	5	
t15	Write data delay from BCLK falling	2	40
t16	Write data delay from BCLK rising	5	40
t17	Write data hold from BCLK rising	5	

EISA Two Cycle Transfer – 16-bit Translate

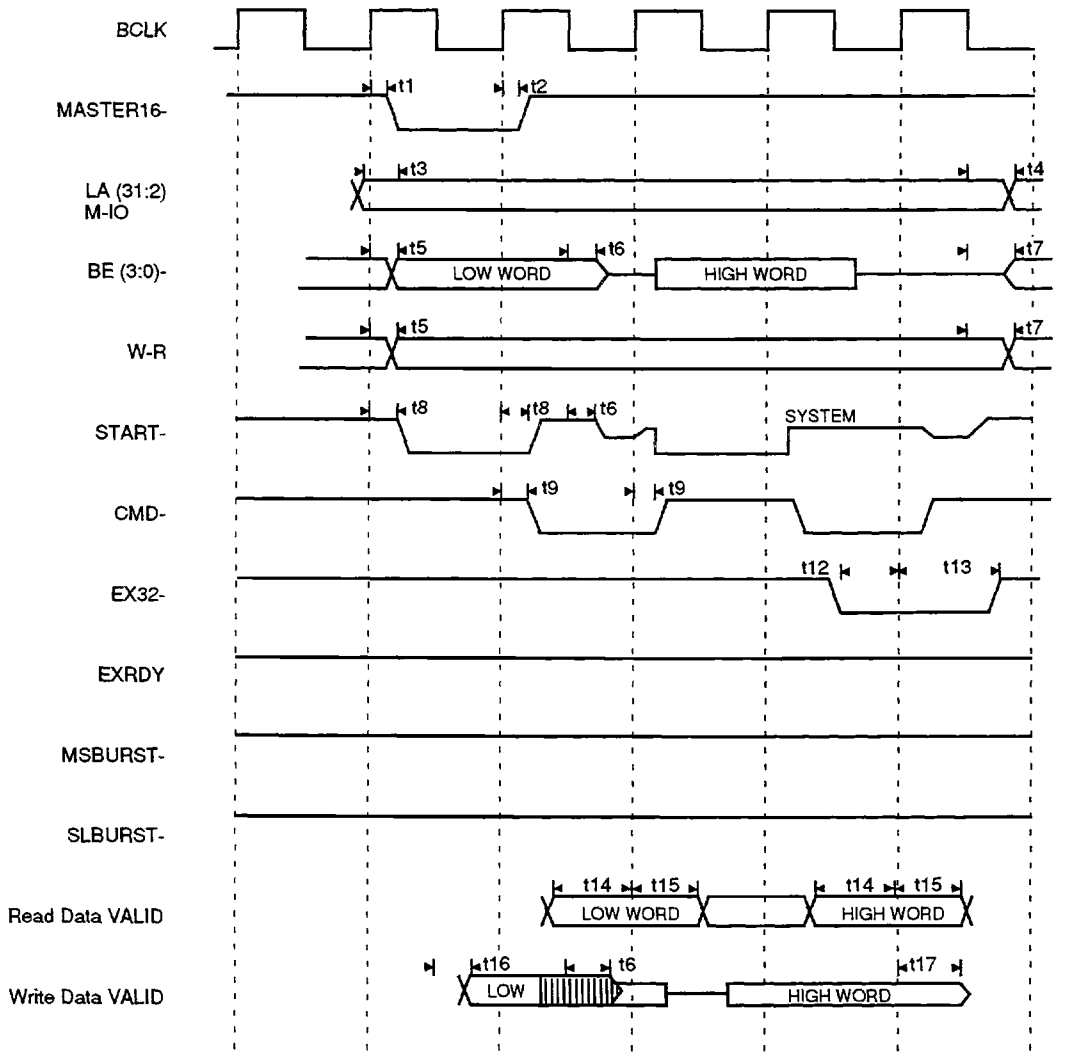


Figure 9-10. EISA Two Cycle Transfer–16-bit Translate

Parameter	Description	Time (ns)	
		min	max
t1	MASTER16- assert delay from BCLK rising	2	50
t2	MASTER16- float delay from BCLK rising	2	40
t3	LA, M-IO setup START-	10	
t4	LA, BE- delay from BCLK falling	2	45
t5	BE-, W-R delay from BCLK rising		25
t6	Float delay from BCLK falling	2	50
t7	BE-, W-R delay from BCLK falling	2	85
t8	START- delay from BCLK rising	2	25
t9	CMD- delay from BCLK rising	2	25
t10	---		
t11	---		
t12	EX32- setup to BLCK rising	15	
t13	EX23- hold from BCLK rising	50	
t14	Read data setup to BCLK rising	15	
t15	Read data hold from BCLK rising	4	
t16	Write data delay from BCLK falling	2	40
t17	Write data hold from BCLK rising	30	

EISA Slave Bus Timing

EISA I/O Slave – 8-bit Write

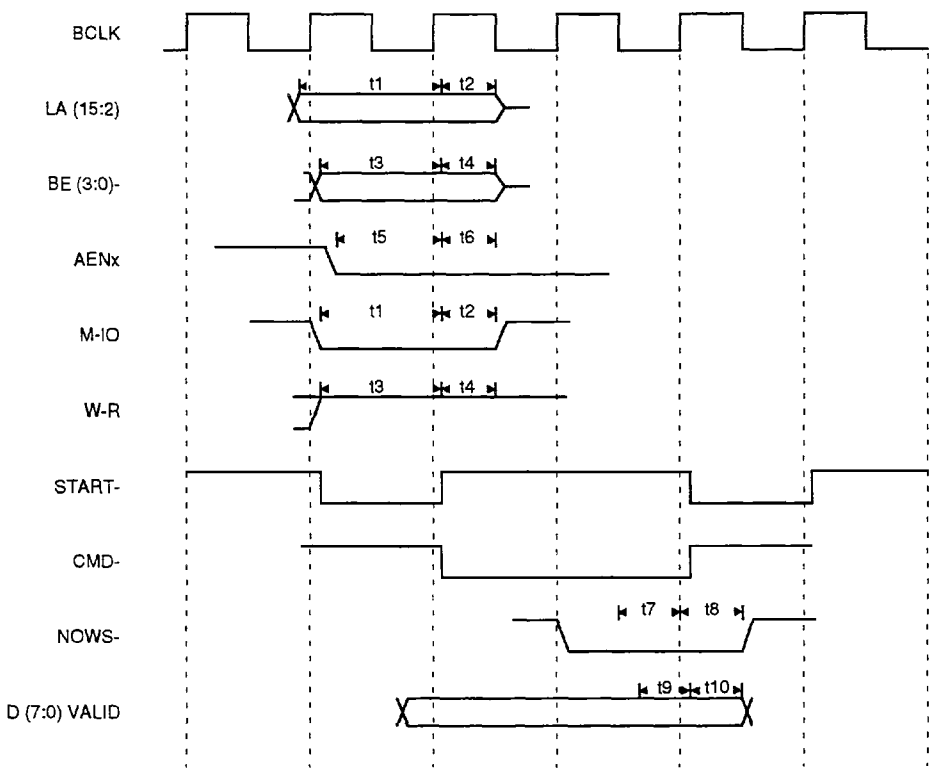


Figure 9-11. EISA I/O Slave – 8-bit Write

Parameter	Description	Time (ns)		
		min	typ	max
t1	LA, M-IO setup to START- negated	120		
t2	LA, M-IO hold from START- negated	15		
t3	BE(3:0), W-R setup to START- negated	80		
t4	BE(3:0), W-R hold from START- negated	15		
t5	AENx setup to START negated	95		
t6	AENx hold from START negated	25		
t7	NOWS- setup to BCLK rising edge	15		
t8	NOWS- hold from BCLK rising edge	5		
t9	DATA setup to CMD- negated	100		
t10	DATA hold from CMD- negated	25		

EISA I/O Slave – 8-bit Read

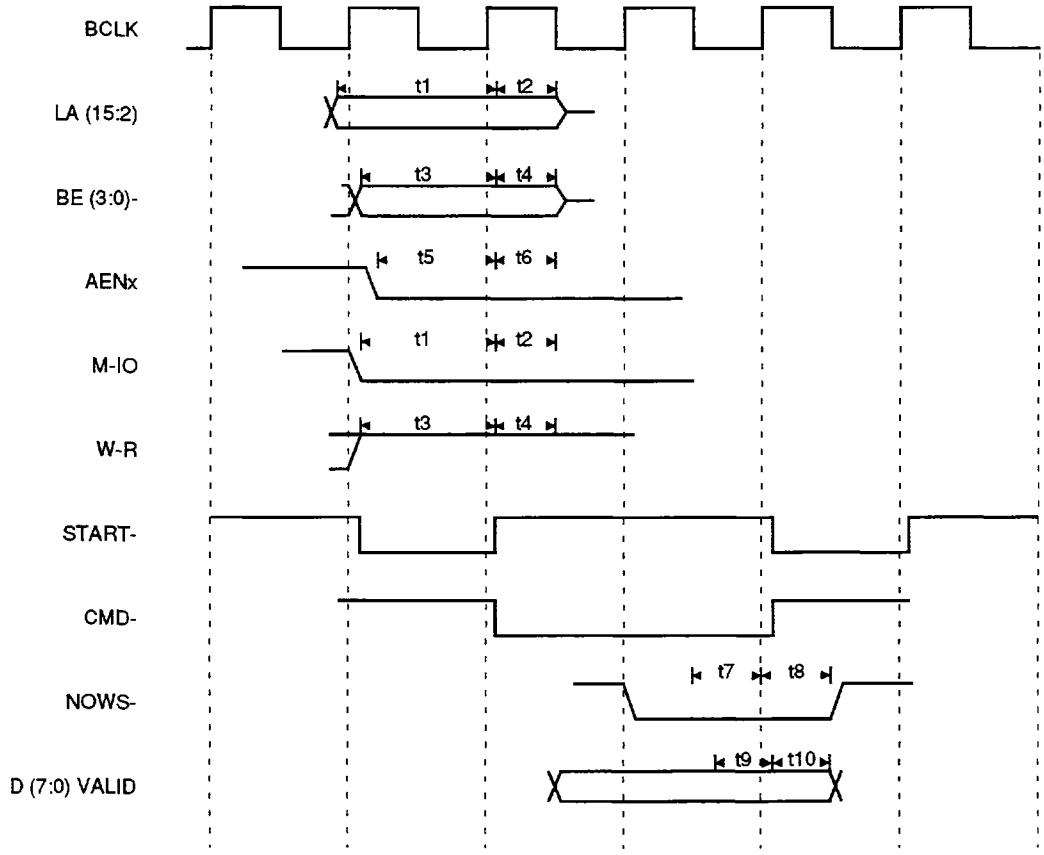


Figure 9-12. EISA I/O Slave – 8-bit Read

Parameter	Description	Time (ns)		
		min	typ	max
t1	LA, M-IO setup to START- negated	120		
t2	LA, M-IO hold from START- negated	15		
t3	BE(3:0), W-R setup to START- negated	80		
t4	BE(3:0), W-R hold from START- negated	15		
t5	AENx setup to START negated	95		
t6	AENx hold from START negated	25		
t7	NOWS- setup to BCLK rising edge	15		
t8	NOWS- hold from BCLK rising edge	5		
t9	DATA setup to BCLK rising edge	100		
t10	DATA hold from BCLK rising edge	25		

ISA Master Bus Timing

ISA Master Arbitration

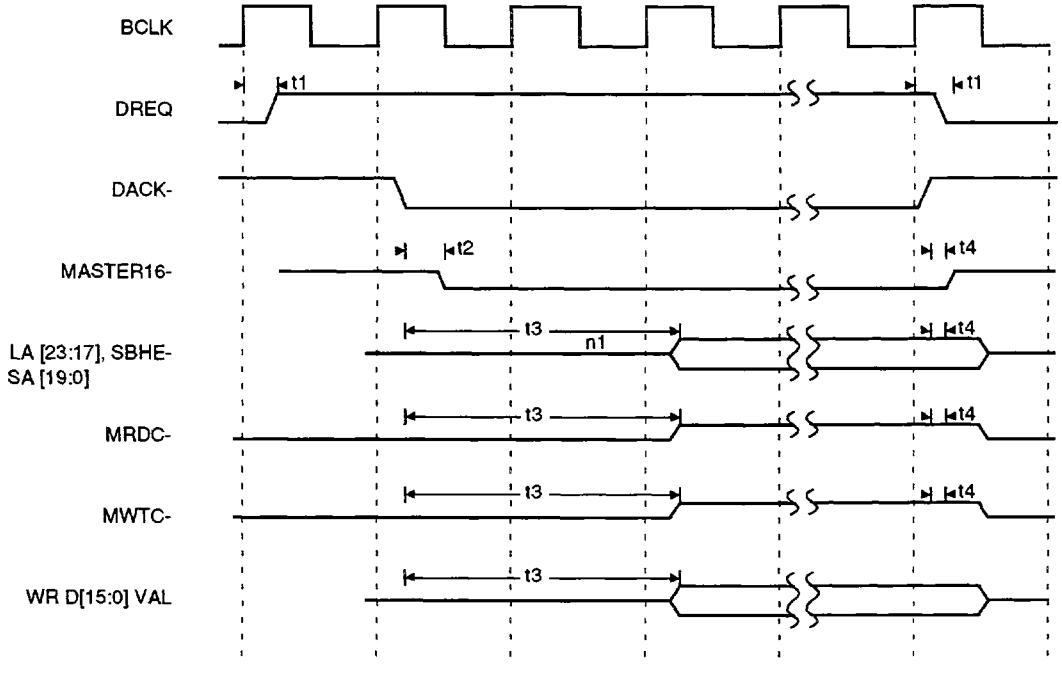


Figure 9-13. ISA Master Arbitration

Parameter	Description	Time (ns)	
		min	max
t1	Delay to DREQ valid from BCLK rising		45
t2	Delay to MASTER16- active from DACK-		60
t3	Delay to driving bus from DACK- falling	4T	6T
t4	Delay to releasing bus from DACK rising		50
n1	One BCLK min from MASTER16- active		

ISA Master – 16-bit Transfers

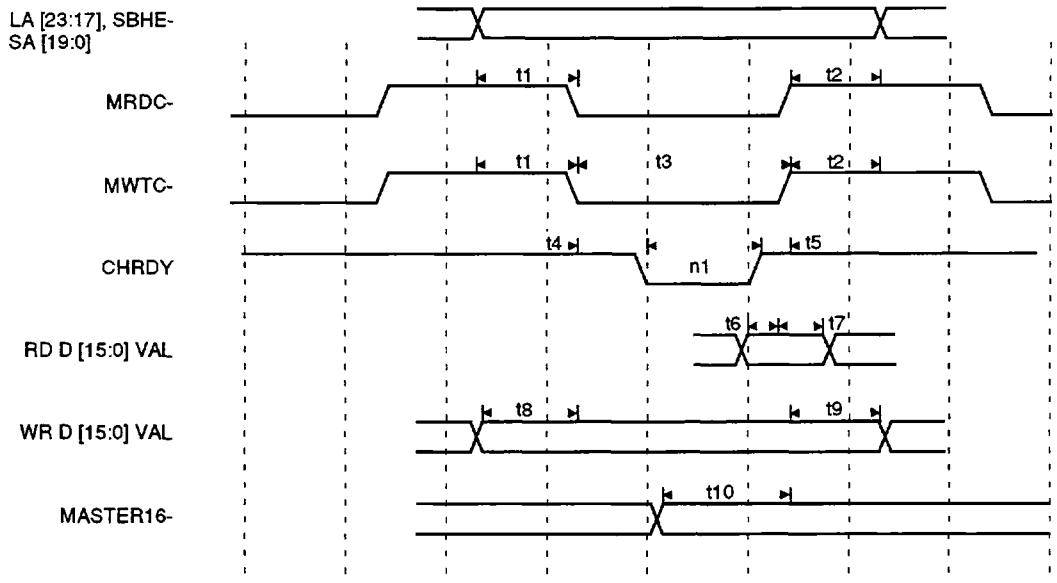


Figure 9-14. ISA Master – 16-bit Transfers

Parameter	Description	Time (ns)	
		min	max
t1	LA,SA,SBHE-, setup to MRDC-/MWTC- falling	112	
t2	LA,SA,SBHE-, hold from MRDC-/MWTC- rising	30	
t3	MRDC-, MWTC-, pulse width	230	2900
t4	CHRDY falling from MRDC-/MWTC- falling		98
t5	CHRDY rising to MRDC-/MWTC- rising	108	
t6	Read data setup to MRDC- de-asserted	36	
t7	Read data hold from MRDC- de-asserted	0	
t8	Write data valid to MWTC- falling	-40	
t9	Write data hold from MWTC- rising	22	
t10	MASTER16- setup to MRDC-/MWTC- rising	80	
n1	CHRDY is sample 2T before the de-assertion of MRDC- or MWTC-		
n2	M16- is sampled when A0000<addr<BFFFF		

ISA Slave Bus Timing

ISA I/O Slave – 8-bit Write/Read

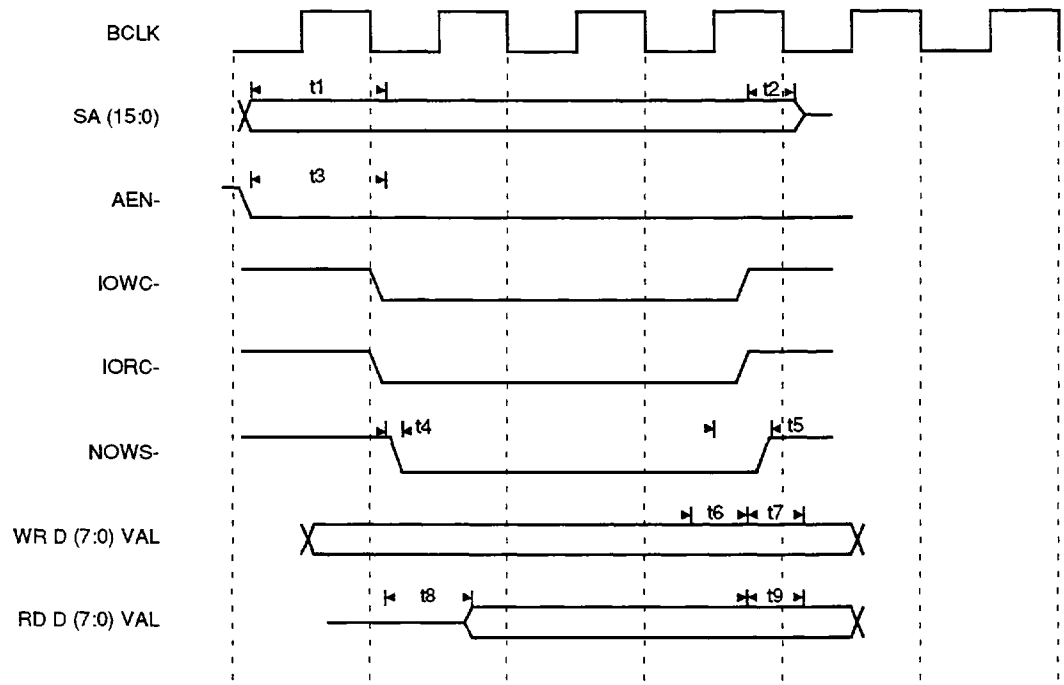


Figure 9-15. ISA I/O Slave – 8-bit Write/Read

Parameter	Description	Time (ns)	
		min	max
t1	SA(15:0) setup to IOWC- active	88	
t2	SA(15:0) hold form IOWC- negated	32	
t3	AEN setup to IOWC- active	100	
t4	NOWS- active delay from IOWC- active	?	
t5	NOWS- release delay from BCLK rising	?	
t6	Data setup to IOWC- negated	22	
t7	Data hold from IOWC- negated	25	
t8	Data delay from IORC- active	25	
t9	Data hold from IORC- negated	25	

Differential Timing

Initiator Arbitration Selection

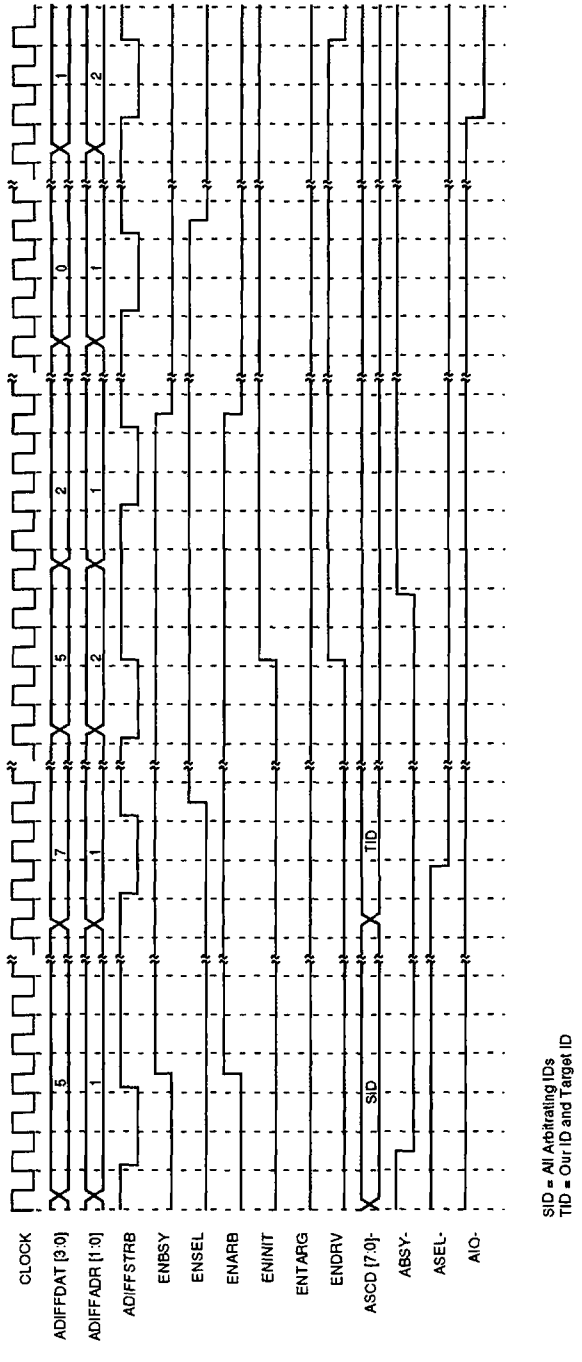


Figure 9-16. Initiator Arbitration Selection

Target Arbitration Reselection

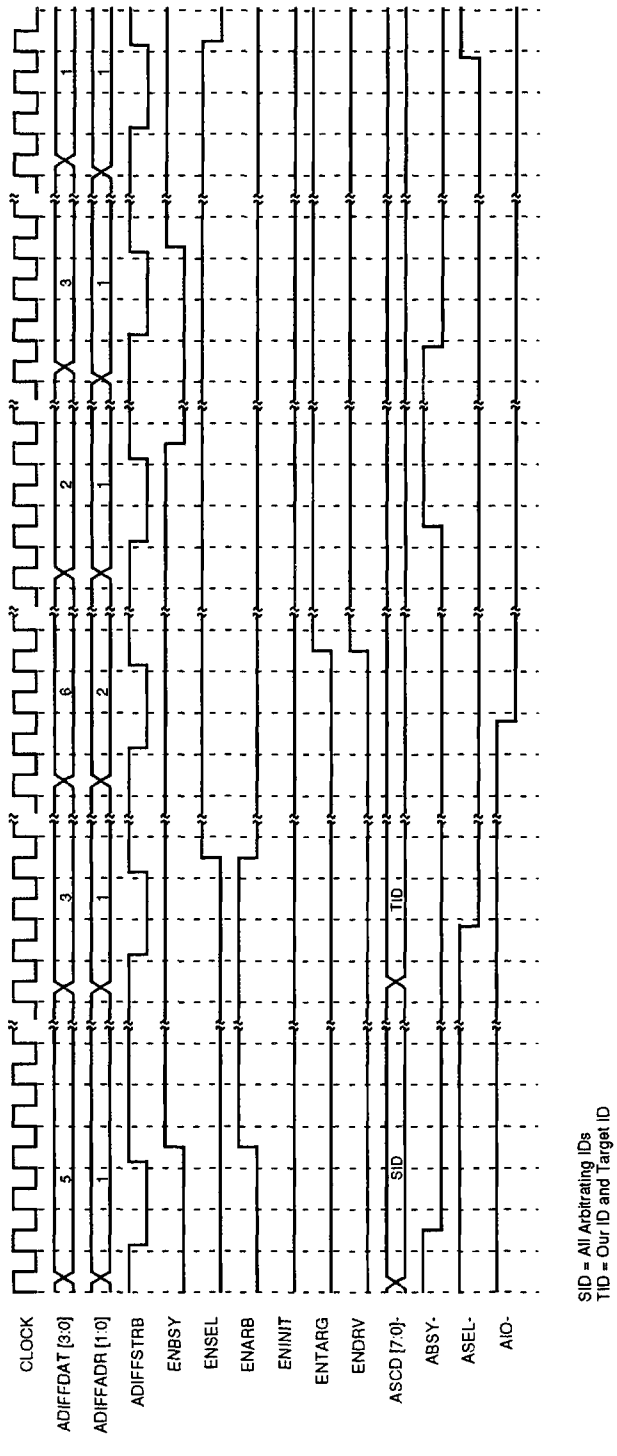


Figure 9-17. Target Arbitration Reselection

Initiator Reselection

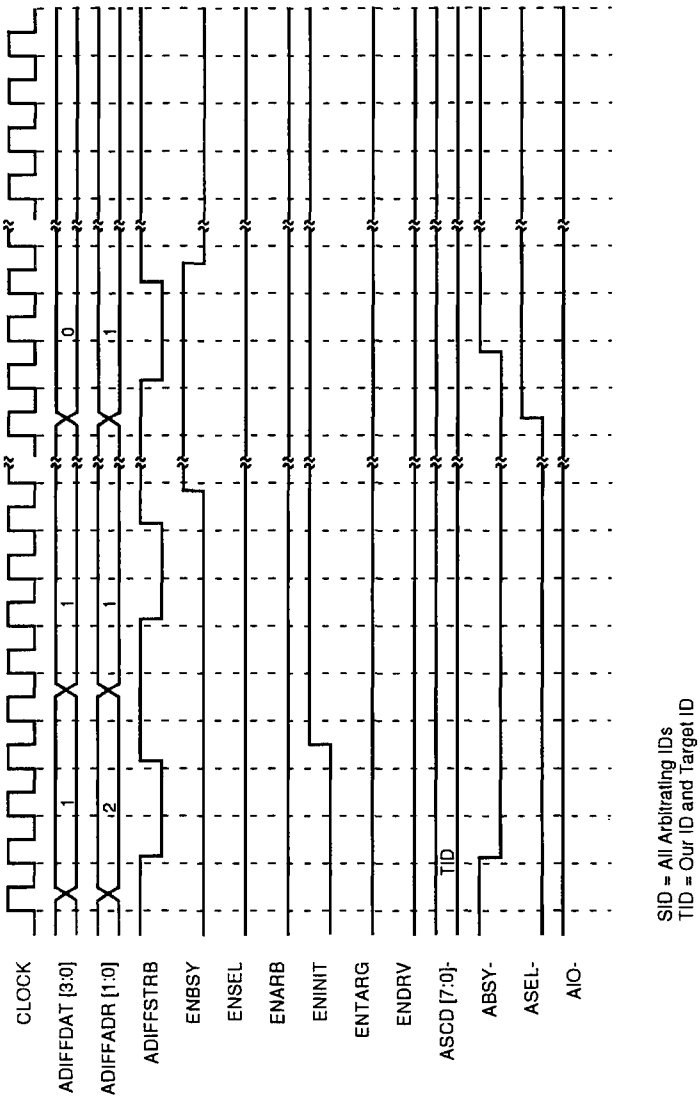


Figure 9-18. Initiator Reselection

Target Selection

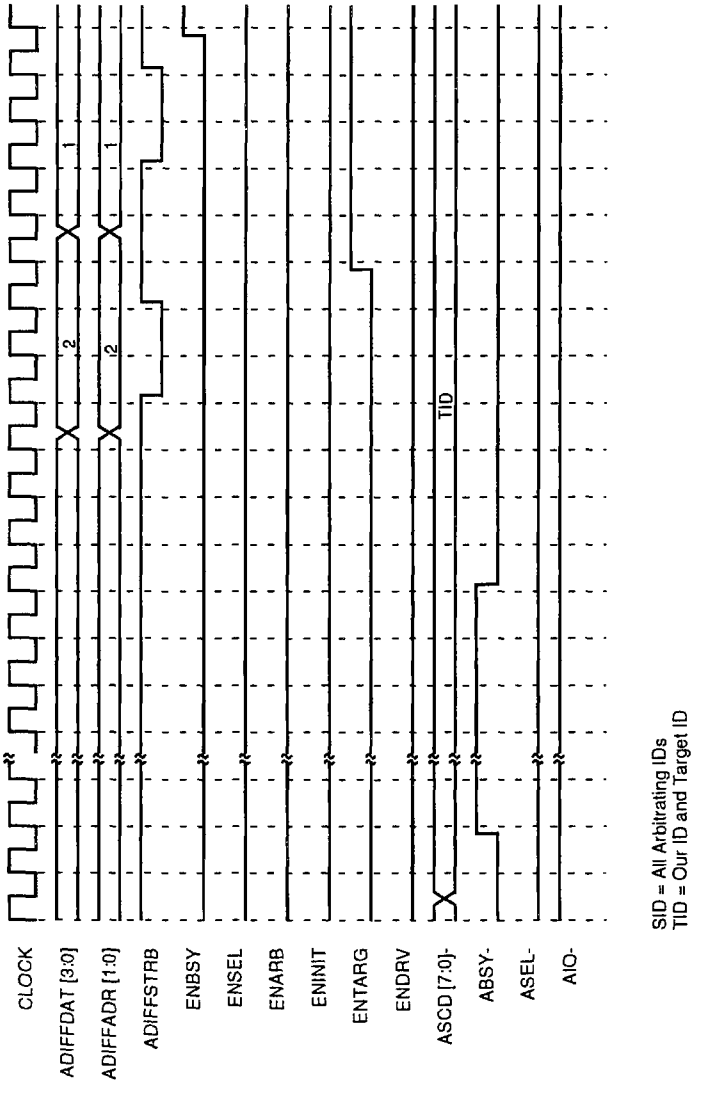
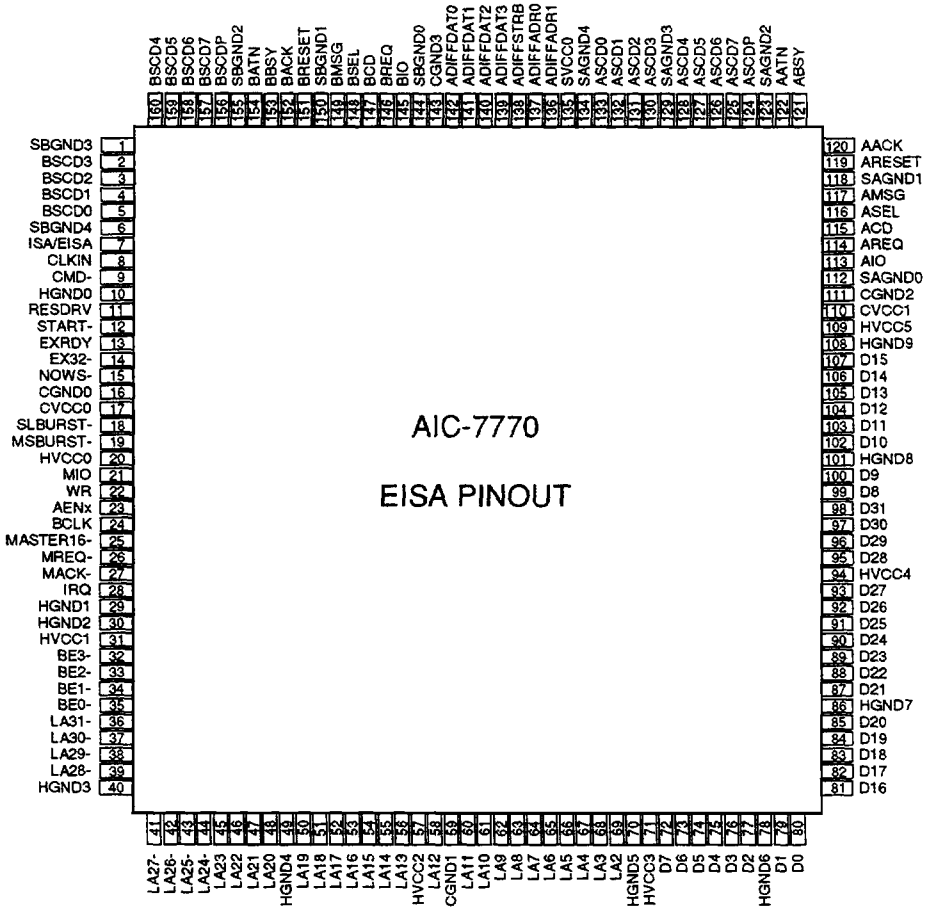


Figure 9-19. Target Selection

10

EISA Package Outline

160 pin, QFP, 28mm EIAJ standard.



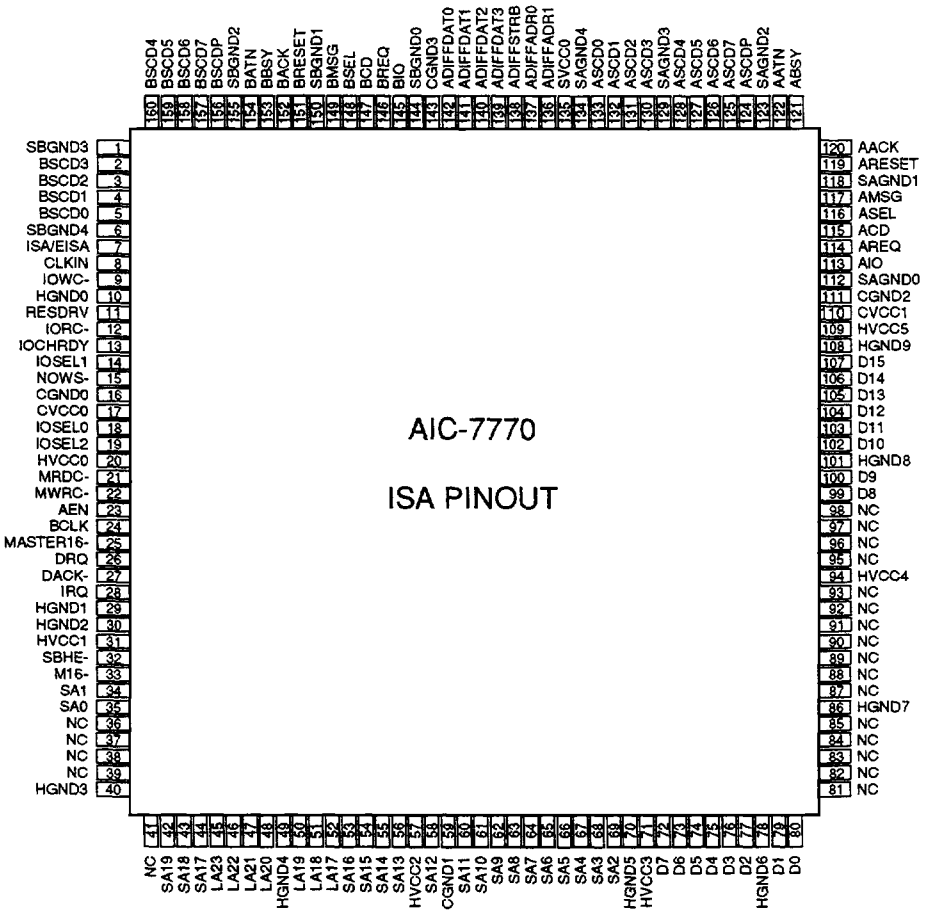
CVCCx = CHIP VCC
CGNDx = CHIP GND
SVCCx = SCSI VCC
SAGNDx = SCSI A BUS GND
SBGNDx = SCSI B BUS GND
HVCCx = HOST VCC
HGNDx = HOST GND
NC = NO CONNECTION

Figure 10-1. EISA Package Outline

11

ISA Package Outline

160 pin, QFP, 28mm EIAJ standard.



CVCCx = CHIP VCC
CGNDx = CHIP GND
SVCCx = SCSI VCC
SAGNDx = SCSI A BUS GND
SBGNDx = SCSI B BUS GND
HVCCx = HOST VCC
HGNDx = HOST GND
NC = NO CONNECTION

Figure 11-1. ISA Package Outline