Section One General Description

Adaptec's AIC-6110 is a high-performance, single-chip synchronous SCSI storage controller, which supports all the functions necessary to build a high-performance controller, except data separation, for the *embedded drive* manufacturers including but not limited to, hard, floppy, tape, and optical drives. The AIC-6110 single-chip synchronous SCSI storage controller offers the industry's highest synchronous data transfer rate of 5 Mbytes/second and asynchronous transfer rates of 3 Mbytes/second over the full SCSI cable length. The AIC-6110 also supports high NRZ data transfer rates up to 20 Mbits/sec. Adaptec's AIC-6110 is a single-chip synchronous SCSI storage controller intended for embedded tape and drive applications. The AIC-6110 transfers data between a local buffer or SCSI host and serial bit stream at rates up to 20 Mbits/sec.

The AIC-6110 incorporates the functions of four chips in one. Functions of the Adaptec AIC-011 high performance programmable controller, an enhanced version of the AIC-300 buffer manager, the AIC-270F 2,7 RLL encoder/decoder, and the AIC-6250 SCSI protocol chip have been integrated. Data separation can be provided with Adaptec's high-performance, high-precision data separator, the AIC-6225. Its high data transfer rates match the AIC-6110's high performance, and no glue logic is needed to connect the two chips. Figure 1 shows the block level diagram of the components required for an embedded SCSI controller.

Other features of the AIC-6110 is the support of DRAM or SRAM. The AIC-6110 supports DRAM with internal refresh counters and does not require additional logic to support refresh. The AIC-6110 also supports programmable precomp allowing for a greater choice of media capability. The integrated 48 mA drivers in the AIC-6110 connect the chip directly to the SCSI bus. For the SCSI bus, the AIC-6110 incorporates an 8-byte FIFO buffer, that maximizes synchronous and asynchronous SCSI performance.

Developed in CMOS, the AIC-6110 offers the minimum space requirements, low-power consumption and increased reliability, and flexibility for a high performance embedded system.

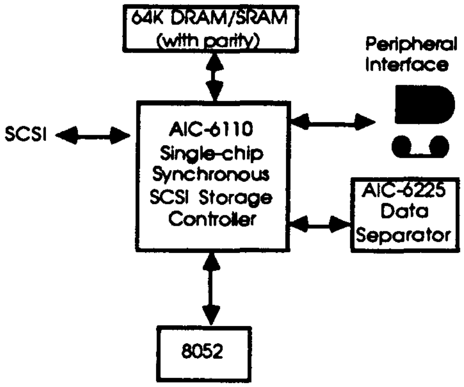

64K DRAM/SRAM (with parity)
Peripheral Interface
SCSI
AIC-6110 Single-chip Synchronous SCSI Storage Controller
AIC-6225 Data Separator
8052

FIGURE 1. EMBEDDED CONTROLLER DESIGN WITH THE AIC-6110

Some of the key features of the AIC-6110 are summarized below:

- Up to 5 Mbytes/sec. Synchronous SCSI transfer rate, and 3 Mbytes/sec. Asynchronous SCSI transfer rate over full SCSI cable length
- NRZ data rates up to 20 Mbits/sec.
- Software compatible to Adaptec's AIC-010/AIC-300 architecture
- Supports industry standard SRAMs and DRAM
- Separate buffer clock and disk clock for speed matching
- Choice of parity on buffered data
- 64K buffer addressing
- Direct SCSI signal interface with 48 mA drivers
- 8-byte offset when in synchronous mode
- 8-byte FIFO between SCSI port and buffer memory
- DMA and Auto PIO transfer modes for SCSI data
- Separate microprocessor and SCSI data paths to minimize overhead
- 32-bit and 48-bit programmable polynomial for EDAC, and fixed 16-bit CRC polynomial for detection
- Data to/from disk can be either NRZ or 2,7 RLL encoded
- Programmable delay control for precompensation
- +5V supply, 68-pin PLCC package or 80-pin Quad Flat package, low power and high-speed CMOS process

Functional Description

AIC-6110 is a software configurable VLSI chip that provides the buffer management, SCSI bus control, encode/decode, and disk control functions for an intelligent disk, floppy, optical or tape controller. The AIC-6110 chip simplifies and increases the throughput of block-oriented high-performance peripheral controllers. Adaptec's AIC-6110 is intended for use in intelligent controllers utilizing a low cost microprocessor for supervision.

The AIC-6110 is an advanced VLSI chip fabricated in CMOS technology that allows for operation with data rates up to 20 Mbits/sec. NRZ or 15 Mbits/sec. in the 2,7 RLL mode. The CMOS design significantly helps the systems designer because of reduced power consumption.

The serializer/deserializer and sequencer in AIC-6110 is similar to the AIC-010F/AIC-011F architecture. This allows software compatibility with firmware being used in today's products that use the Adaptec architecture. This section provides the interface between the serial data path to a parallel buffer memory. The fully programmable RAM-based sequencer controls the various operations for formatting, reading, and writing. The controller section also provides for a 32-bit or 48-bit programmable ECC polynomial for error detection and correction and also a fixed 16-bit CRC-CCITT polynomial for error detection purposes on the ID field.

The architecture of AIC-6110 is demonstrated by the functional block diagram in Figure 2. The AIC-6110 can be separated into five functional blocks for discussion:

- The Microprocessor Interface
- The SCSI Interface
- The Disk Controller
- The Buffer Controller
- The 2,7 RLL ENDEC

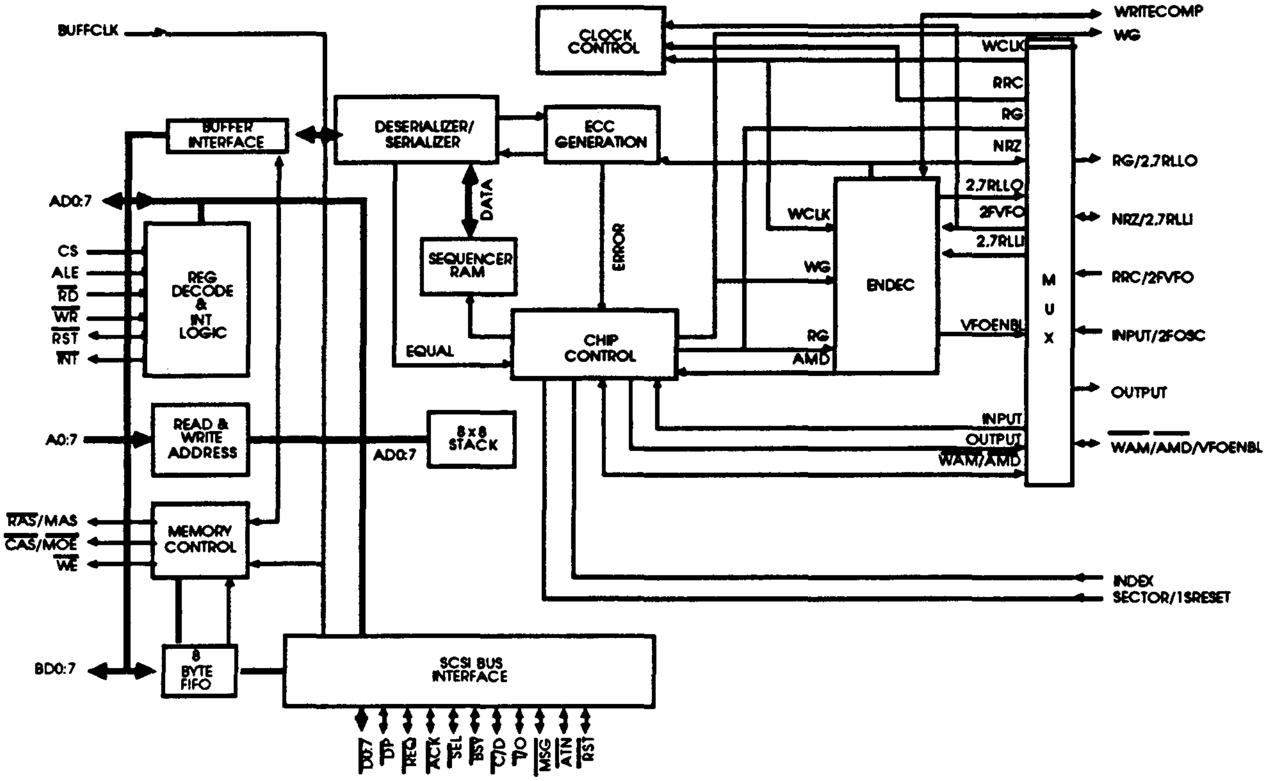


FIGURE 2. AIC-6110 BLOCK DIAGRAM

3.1 MICROPROCESSOR INTERFACE

The AIC-6110 requires microprocessor control to initiate operations and commands, and to check the status of the operation. With the help of automatic arbitration, selection and reselection on the SCSI bus, the programmable sequencer RAM, and the interrupts available the microprocessor housekeeping has been reduced to a minimum.

The AIC-6110 has a separate high-speed 8-bit multiplexed bus for interface to the microprocessor; this allows the microprocessor to access and update registers, even while the AIC-6110 is transferring data.

All registers in the AIC-6110 appear as unique memory or I/O locations and can be randomly accessed and operated upon. These register definitions are also compatible with the register definitions of the individual Adaptec chips which are now integrated in the AIC-6110.

3.2 SCSI INTERFACE

The SCSI interface block in the AIC-6110 provides all the functions necessary to implement a high-performance SCSI interface for the peripheral device. This block contains an eight-byte FIFO, direct SCSI bus signal drivers, SCSI bus control of all phases, and support of both target and initiator roles.

Both asynchronous and synchronous methods of information transfer on the SCSI bus are supported. Asynchronous transfer is a two-wire handshake protocol where the target device drives the SCSI *REQ signal, and the initiator device drives the SCSI *ACK signal.

Synchronous transfer is similar to asynchronous transfer, but the number of *REQ and *ACK signals can be offset by an offset count. The eight-byte FIFO serves as a speed matching buffer between the host memory and the SCSI bus. A performance advantage is realized here since the multiple bus line delay does not exist for every request/acknowledge handshake.

If the synchronous offset is set to zero, the asynchronous method of DMA transfer is automatically enabled. The transfer takes place automatically until the desired number of bytes have been transferred. This is established by a match between the Read or Write Address Pointer, and the Stop Pointer.

If the offset count is set to any number other than zero, the synchronous mode of DMA transfer is enabled. This mode is similar to the asynchronous mode with the exception that the number of requests and acknowledges can be offset by an amount equal to the offset count. The DMA cycle ends when all the data has successfully transferred, that is the offset is zero and the appropriate Address Pointer matches the Stop Pointer.

In addition to the synchronous and asynchronous methods of DMA data transfer, automatic PIO, and microprocessor controlled PIO methods of transfer are also implemented. Automatic PIO requires the microprocessor to intervene once each byte time. The microprocessor transfers a byte of data between the memory and the AIC-6110 register and then the SCSI handshake is automatically handled by the chip. After completion of the information transfer, an interrupt will be generated, or the microprocessor can mask out the interrupt and poll for the completion of transfer. This mode should not be used with synchronous transfer and does not utilize the FIFO in the asynchronous mode.

Microprocessor controlled PIO gives the microprocessor complete control over the SCSI *REQ and *ACK signal (depending on initiator or target mode of operation). This mode also requires the microprocessor to read or write data to or from the SCSI bus through a SCSI data register. This mode should not be used with synchronous transfer and does not utilize the FIFO in the asynchronous mode. The SCSI bus interface includes the 48 ma drivers and receivers for the single-ended SCSI bus option. This eliminates the need for external drivers and receivers.

The Parity Logic generates and optionally checks SCSI bus odd parity during information transfer on the SCSI bus, as well as automatically generating and optionally checking odd parity on memory data. While parity generation and checking is automatically performed when parity is enabled, the generation of an interrupt in the event of a parity error is optional. The interrupt mask may be used to enable an interrupt on SCSI parity error, or memory parity error, or both. If parity checking is enabled and an error is detected on the SCSI bus, the SCSI controller interrupts the microprocessor and continues data transfer. If parity checking is enabled and an error is detected during memory transfer, an interrupt is generated to the microprocessor and the transfer is halted.

The SCSI Select and Reselect control provide the logic to perform the SCSI bus arbitration scheme, implemented by most SCSI systems, as well as automatic response to Selection and Reselection. The controller may be set up to automatically look for the SCSI Bus Free phase. When the Bus Free phase is detected, the controller will arbitrate for control of the SCSI bus. If arbitration is won, it will proceed to the selection or reselection phase.

The selection timeout may be accomplished by starting a timer from an interrupt which occurs when arbitration has been won and the controller asserts the SCSI *SEL signal. The completion of the automatic arbitration and selection/reselection process is indicated by a Command Done interrupt.

The SCSI controller's interface to the buffer controller is simply an internal two-wire request/acknowledge handshake. Upon initiation of DMA transfer from the microprocessor, the buffer controller will transfer data to or from the eight-byte FIFO as long as there is enough room or the FIFO is not empty respectively.

3.3 DISK CONTROLLER

The disk controller block is specifically designed to be software compatible with Adaptec's popular AIC-6110/AIC-011 product line. This programmable controller provides the major portion of hardware for disk control and data path interface between buffer memory and serial disk data. The disk controller has a fully user programmable RAM-based sequencer that can be loaded in order to change its mode of operation.

The sequencer RAM can be easily modified to achieve full compatibility with a variety of drive interfaces. The controller also has a user-programmable 32-bit or 48-bit ECC polynomial, and a fixed 16-bit CRC polynomial, thus offering full flexibility in implementing error detection and correction. The controller provides the necessary serialization/deserialization, formatting, function sequencing, and ECC processing, in addition to search and verify capabilities.

The controller is designed to be used with a low-cost microprocessor. The processor is used to maintain loose synchronization with the real time on the device. The controller on the other hand maintains a close synchronization of data to and from the disk, and provides the signals necessary to control this path. The basic operation of this block is controlled by the contents of the sequencer RAM. The RAM consists of 96 bytes, organized as 24 words that are 4 bytes wide. The four bytes can be broken down into Data, Count, Control, and Next Address Fields.

The operation of the sequencer revolves around the BRANCH/NEXT ADDRESS REGISTER and the STATUS/START ADDRESS REGISTER. The START ADDRESS REGISTER is first loaded with the address where the sequencer is to begin execution. Thereafter, the chip sequences through the RAM and the next address executed is based on the contents of BRANCH/NEXT ADDRESS REGISTER if a successful branch condition occurs. Otherwise it is based on the contents of the next address field at that address. Thus, by setting up different branch conditions which are based on external or internal events, the sequencer can be made to sequence through different operations.

The controller also has a stack that is eight bytes deep. By enabling the stack during a read operation, the information read from the drive can be pushed onto the stack. These bytes can later be popped by the microprocessor at a lower speed to examine ID fields and similar information.

During a read operation, the controller also has the ability to compare the data being received on a byte-for-byte basis with information found in the data field of the sequencer RAM, or with information stored in the external buffer. This allows the controller to search for the correct ID field, or to search for a desired data pattern.

If an ECC error is detected after a read data operation, the syndrome is saved in the ECC register and will not be reset until a new read operation is started. By employing internal registers, the microprocessor can determine if the error is correctable and can calculate the error pattern and displacement from the beginning of the sector.

The interface between the disk controller and the buffer controller is through an internal asynchronous handshake. Since the disk controller and buffer controller run with independent clocks the asynchronous handshake signals are synchronized with the buffer clock and serviced as soon as possible. A one-byte FIFO between the disk controller and buffer controller insures that the data will not be lost if the buffer controller cannot immediately respond to the disk request.

The role of the sequencer along with the extensive data handling operation of the controller warrant a detailed discussion on its operation. Further discussion on the operation of the controller block can be found by referring to the Functional Operation Section (7.0).

Finally, the AIC-6110 employs several interrupts to ease the supervision task of the processor and keep the overhead minimum. Occurrences of Address Mark Detect, ECC Error, Sequencer Stopped, Data Xfer Active, Branch Active and Compare Equal conditions, all can initiate an interrupt. The designer can also mask out any and all interrupts if it is so desired.

3.4 BUFFER CONTROLLER

The buffer controller is specifically designed to simplify the buffering and increase the throughput of the controller function. Its primary function is to allow low cost static or dynamic RAMs to be utilized as a dual port circular FIFO, service transfer requests to and from the host or drive, provide all the necessary signals to fully control various industry standard RAMs. It is operated from a separate buffer clock and hence provides a higher throughput. The buffer controller contains logic for resolving disk/host requests by giving priority to the peripheral. The processor may also access the buffer data through a Memory PIO operation. However, the processor must insure that no data transfer to or from the host is taking place.

The buffer controller is capable of handling buffer sizes from 256 bytes to 64K bytes. In the DRAM mode, the buffer controller can access 64K of DRAM directly with the support of CAS before RAS automatic DRAM refresh. The chip can be configured with the parity option on the buffer data bus for full parity checking through the SCSI bus. There are a total of eight address signals which are multiplexed to provide the full 64K address space. The Buffer controller operates with a wraparound buffer managed with pointers indicating the start and stop locations in the buffer. Refer to Section 7.3 for more details.

3.5 ENDEC

The encoder/decoder (ENDEC) block within AIC-6110 provides an efficient data interface to a 2,7 Run Length Limited (RLL) recorded disk drives. The user can bypass the ENDEC altogether through a software programmable mode bit, in which case the AIC-6110 generates NRZ data. This can then be a suitable interface to ESDI type drives, or the output can be encoded with additional external circuitry to provide an MFM or 1,7 RLL type of encoding schemes. The onboard ENDEC can generate two types of preamble patterns, two possible address mark patterns.

By enabling precompensation through a programmable register bit, the 2,7 RLLO output data can be precompensated. The amount of precompensation is determined by two programmable delay registers for Early and Late precompensation. The output signal Write Precomp (WPCOMP) is an open drain output which must have an external resistor capacitor network to establish a desired ramp rate for precompensation.

Section Four Pin Description

SYMBOL	*PIN*	*PIN*	*TYPE*	*DESCRIPTION*
	PLCC	*QFP*		
CS	10	64	I	CHIP SELECT: Active during processor bus cycles.
*INT	14	68	O	INTERRUPT: When active, this signal indicates that an interrupt is pending. The interrupt condition is described in the internal status registers. This is an open drain output buffer.
ALE	11	65	I	ADDRESS LATCH ENABLE: Signal used to latch the address from the multiplexed address/data bus. The address is latched on the falling edge of ALE.
*POR	15	69	I	POWER ON RESET: When this signal is driven low the chip is reset. The internal software reset latch will remain active until it is cleared by the microprocessor.
*RD	12	66	I	READ: Read and chip select active causes the data from an internal register to be enabled onto the AD0:7 bus.
*WR	13	67	I	WRITE: Write and chip select active causes the data on the AD0:7 to be written to an internal register.
AD0:7	2:9	53,54,56,57,59, 34,35,36	I/O	MULTIPLEXED ADDRESS/DATA LINES interfacing to the control processor.
BD0:7	49:56	29,30,32,32,33, 34,35,36	I/O	BUFFER DATA BUS: Byte parallel tristatable data lines to and from the buffer memory. These pins along with the BDP pin are normally tristated, and are driven only during a memory write cycle.
BDP	57	37	I/O	BUFFER DATA PARITY: Odd parity bit generated from the buffer data when writing to buffer memory but checking is optional when reading from buffer memory.
A0:7	61:68	41,42,44,45,47, 48,50,51	O	BUFFER ADDRESS BUS: Multiplexed low and high order address bytes.

SYMBOL	*PIN* *PLCC*	*PIN* *QFP*	*TYPE*	*DESCRIPTION*
*CAS/ *MOE	59	39	O	COLUMN ADDRESS STROBE in DRAM mode, MEMORY OUTPUT ENABLE for enabling the memory's output during a read cycle in SRAM mode.
*WE	58	38	O	WRITE ENABLE: Asserted when a buffer write operation is desired. Note timing difference in SRAM and DRAM modes.
INPUT/ 2FOSC	16	70	I	General purpose INPUT signal: as a branch condition to the sequencer RAM., if used in the NRZ mode of operation. In the RLL mode, this pin is used as the write oscillator clock source which must be twice the frequency of the RLL data.
OUTPUT	17	71	O	General purpose OUTPUT port derived from the control field of the sequencer RAM in the NRZ and RLL modes.
INDEX	18	72	I	INDEX pulse input from disk.
SECTOR/ RESET1S	19	73	I	SECTOR pulse input from disk in the hard sectored mode, One's Reset input from an external one shot in the softsectored mode. (The desired mode can be set through bit 4 in Register 71.)
RG/ 2,7RLLO	20	74	O	READ GATE in the NRZ mode, encoded 2,7 RLL DATA OUTPUT in the RLL mode. The encoded data will be compensated if the precompensation mode is enabled.
WPCOMP	21	75	O	WRITE PRECOMPENSATION ramp control open drain output. An external resistor and capacitor network will establish the desired ramp rate on this output. The output is redirected into the chip where a set of comparators establish the desired delay for precompensation. The incremental precompensation delay is equal 0.05 x RC.
WG	22	76	O	WRITE GATE output signal to disk.
NRZ/ 2,7RLLI	23	77	I/O	NRZ input/output data signal in the NRZ mode. 2,7 RLL input signal from the data separator in the RLL mode.
*RAS/ MAS	60	40	O	ROW ADDRESS STROBE in DRAM mode, MEMORY ADDRESS STROBE for latching the low byte of a 16 bit address in SRAM mode.

SYMBOL	*PIN PLCC*	*PIN QFP*	*TYPE*	*DESCRIPTION*
*WAM/*AMD/ VFOENBL	24	78	I/O	WRITE ADDRESS MARK/ADDRESS Mark Detect in the NRZ mode, VFO Enable output to VFOEN the data separator in the RLL mode.
RRC/ 2FVFO	25	79	I	READ REFERENCE CLOCK in the NRZ mode, and 2FVFO oscillator input from the data separator in the RLL mode.
BUFFCLK	26	80	I	BUFFER CLOCK is the primary clock used by the Buffer controller and the SCSI controller blocks.
*D0:7	38:42 44:46	16,18,19,21,22, 24,25,26	I/O	DATA BUS: These signals carry the eight-bit bidirectional data on the SCSI bus.
*DP	47	27	I/O	DATA PARITY: Odd parity for SCSI data bus. Parity is always generated, but checking is optional.
*C/D	32	9	I/O	COMMAND/DATA: This line is asserted when command information is on the SCSI bus, and is tristated when data information is on the bus. In the initiator mode this signal is an input, and in the target mode it is an output.
*I/O	33	10	I/O	INPUT/OUTPUT: This signal is driven by the target on the SCSI bus and determines the direction of data transfer with respect to the initiator. The signal is low when data is input to the initiator, and disabled when the information is output from the initiator.
*MSG	34	12	I/O	MESSAGE: The message line is low during a message phase on the SCSI bus. In the initiator mode it is an input, and in the target mode an output.
*ATN	35	13	I/O	ATTENTION: The attention is driven low by the initiator to indicate to the target that there is a message to be transferred. In the target mode this signal is an input.
*REQ	28	3	I/O	REQUEST: This signal is used with the acknowledge signal to perform the SCSI bus handshake protocol. This signal is an input when in initiator mode, and an output when in target mode.

SYMBOL	*PIN* *PLCC*	*PIN* *QFP*	*TYPE*	*DESCRIPTION*
*ACK	29	4	I/O	ACKNOWLEDGE: This signal is used with the request signal to perform the SCSI bus handshake protocol. It is an input when in target mode, and an output when in the initiator mode.
*SEL	30	6	I/O	SELECT: This bidirectional signal carries the SCSI selection signal. This signal is used in the SCSI selection and reselection phase.
*BSY	31	7	I/O	BUSY: This bidirectional signal carries the SCSI busy signal. When busy is active, the SCSI bus is being used.
*RST	37	15	I/O	SCSI BUS RESET: This bidirectional signal will drive reset low on the SCSI bus when the appropriate bit in an internal control register is set. When it is being driven low externally, it indicates that another device is forcing reset. This condition is internally latched by the chip and may be cleared by asserting *POR or a chip software reset.
V_{DD}	1	52	PWR	+5 Volt Supply
V_{SS}	27,36 43	2,5,8,11,14 17,20,23,28	GRD	Ground potential with an isolated ground bus for 48 ma SCSI drivers and all other output pins. - Noisy ground.
V_{SS}	48	1,43,46,49 55,58,61	GRD	Ground potential for internal logic.

50 RESERVED	51 RESERVED	52 RESERVED
7 RESERVED	7 RESERVED	7 RESERVED
6 RESERVED	6 RESERVED	6 RESERVED
5 RESERVED	5 RESERVED	5 RESERVED
4 RESERVED	4 RESERVED	4 RESERVED
3 RESERVED	3 RESERVED	3 RESERVED
2 RESERVED	2 RESERVED	2 RESERVED
1 RESERVED	1 RESERVED	1 RESERVED
0 RESERVED	0 RESERVED	0 RESERVED

	(W)	(W)
53 RESERVED	54 BUFFER SIZE	55 LATCH RAP/WAP
7 RESERVED	7 64K	7 Latch RAP/WAP
6 RESERVED	6 32K	6 Latch RAP/WAP
5 RESERVED	5 16K	5 Latch RAP/WAP
4 RESERVED	4 8K	4 Latch RAP/WAP
3 RESERVED	3 4K	3 Latch RAP/WAP
2 RESERVED	2 2K	2 Latch RAP/WAP
1 RESERVED	1 1K	1 Latch RAP/WAP
0 RESERVED	0 512	0 Latch RAP/WAP

(W)	(W) EARLY PCOMP	(W) LATE PCOMP
56 MODE SELECT REG	57 DELAY 0:7	58 DELAY 0:7
7 RESERVED	7 EDELAY7	7 LDELAY7
6 RESERVED	6 EDELAY6	6 LDELAY6
5 RESERVED	5 EDELAY5	5 LDELAY5
4 *SRAM/DRAM	4 EDELAY4	4 LDELAY4
3 *NRZ/RLL	3 EDELAY3	3 LDELAY3
2 ENBLCOMP	2 EDELAY2	2 LDELAY2
1 VFO OPT	1 EDELAY1	1 LDELAY1
0 AM OPT	0 EDELAY0	0 LDELAY0

(W) RESET ADDRESS 59 POINTERS	(R/W) READ ADDRESS 5A POINTER 0:7	(R/W) READ ADDRESS 5B POINTER 8:15
7 RESET7	7 RAP7	7 RAP15
6 RESET6	6 RAP6	6 RAP14
5 RESET5	5 RAP5	5 RAP13
4 RESET4	4 RAP4	4 RAP12
3 RESET3	3 RAP3	3 RAP11
2 RESET2	2 RAP2	2 RAP10
1 RESET1	1 RAP1	1 RAP9
0 RESET0	0 RAP0	0 RAP8

(R/W) WRITE ADDRESS 5C POINTER 0:7	(R/W) WRITE ADDRESS 5D POINTER 8:15	(R/W) STOP ADDRESS 5E POINTER 0:7
7 WAP7	7 WAP15	7 STOP7
6 WAP6	6 WAP14	6 STOP6
5 WAP5	5 WAP13	5 STOP5
4 WAP4	4 WAP12	4 STOP4
3 WAP3	3 WAP11	3 STOP3
2 WAP2	2 WAP10	2 STOP2
1 WAP1	1 WAP9	1 STOP1
0 WAP0	0 WAP8	0 STOP0

STOP ADDRESS 5F POINTER 8:15	(W) 60 INT MSK REG 0	(R) 60 STATUS REG 0
7 STOP15	7 EN REQ ON INT	7 SCSI REQ ON
6 STOP14	6 EN MEM PERR INT	6 MEM PARITY ERROR
5 STOP13	5 EN SCSI PERR INT	5 SCSI PARITY ERROR
4 STOP12	4 EN BUS FREE	4 BUS FREE DETECT
3 STOP11	3 EN PHSE ERR/ATN ON INT	3 PHASE ERR/ ATN ON
2 STOP10	2 EN SEL OUT INT	2 SELECT OUT
1 STOP9	1 EN RESELECTED	1 RESELECTED
0 STOP8	0 EN SELECTED	0 SELECTED

(W)	(R)	(W)
61 INT MSK REG 1	61 STATUS REG 1	62 CNTRL REG 0
7 EN AM ACTIVE INT	7 RESERVED	7 EN SCSI PARITY
6 EN DATA XFER INT	6 RESERVED	6 EN MEM PARITY
5 EN BRCH ACT INT	5 RESERVED	5 SOFTWARE RESET
4 EN STOPPED INT	4 RESERVED	4 20 MHz CLOCK
3 EN ECC ERROR INT	3 DISK PARITY ERR	3 10 MHZ CLOCK
2 EN COMP EQ INT	2 SCSI RESET IN	2 SCSI ID 2
1 EN CMD DONE	1 CMD DONE	1 SCSI ID 1
0 EN RST OCCR INT	0 SCSI RST OCCURRED	0 SCSI ID 0

(W)	(R)	(W)
63 CNTRL REG 1	64 SCSI SIG	64 SCSI SIG
7 EN MEM PIO	7 SCSI C/D IN	7 SCSI C/D OUT
6 MEM PIO *WR/RD	6 SCSI I/O IN	6 SCSI I/O OUT
5 CLR RST OCCRD	5 SCSI MSG IN	5 SCSI MSG OUT
4 *WOP/ROP	4 SCSI ATN IN	4 SCSI ATN OUT
3 EN DMA	3 SCSI SEL IN	3 SCSI SEL OUT
2 EN SCSI PIO	2 SCSI BSY IN	2 SCSI BSY OUT
1 EN AUTO ATN	1 SCSI REQ IN	1 SCSI ACK(I)/REQ(T) OUT
0 EN ARB/SEL START	0 SCSI ACK IN	0 SCSI RST OUT

(R/W)	(R)	(R) SCSI
65 SCSI DATA	66 SOURCE/DEST ID	67 LATCHED DATA
7 SCSI DATA 7	7 ID 7	7 SCSI PIO IN 7
6 SCSI DATA 6	6 ID 6	6 SCSI PIO IN 6
5 SCSI DATA 5	5 ID 5	5 SCSI PIO IN 5
4 SCSI DATA 4	4 ID 4	4 SCSI PIO IN 4
3 SCSI DATA 3	3 ID 3	3 SCSI PIO IN 3
2 SCSI DATA 2	2 ID 2	2 SCSI PIO IN 2
1 SCSI DATA 1	1 ID 1	1 SCSI PIO IN 1
0 SCSI DATA 0	0 ID 0	0 SCSI PIO IN 0

(W) 67 SCSI BSY RST	(W) 68 OFFSET CNTR	(R) 69 FIFO STATUS
7 SCSI BSY RST	7 RESERVED	7 RESERVED
6 SCSI BSY RST	6 SYNC RATE 2	6 RESERVED
5 SCSI BSY RST	5 SYNC RATE 1	5 OFFSET CNT ZERO
4 SCSI BSY RST	4 SYNC RATE 0	4 FIFO EMPTY
3 SCSI BSY RST	3 OFFSET BIT 3	3 FIFO FULL
2 SCSI BSY RST	2 OFFSET BIT 2	2 FIFO CNT BIT 2
1 SCSI BSY RST	1 OFFSET BIT 1	1 FIFO CNT BIT 1
0 SCSI BSY RST	0 OFFSET BIT 0	0 FIFO CNT BIT 0

6A RESERVED	6B RESERVED	6C RESERVED
7 RESERVED	7 RESERVED	7 RESERVED
6 RESERVED	6 RESERVED	6 RESERVED
5 RESERVED	5 RESERVED	5 RESERVED
4 RESERVED	4 RESERVED	4 RESERVED
3 RESERVED	3 RESERVED	3 RESERVED
2 RESERVED	2 RESERVED	2 RESERVED
1 RESERVED	1 RESERVED	1 RESERVED
0 RESERVED	0 RESERVED	0 RESERVED

6D RESERVED	6E RESERVED	6F RESERVED
7 RESERVED	7 RESERVED	7 RESERVED
6 RESERVED	6 RESERVED	6 RESERVED
5 RESERVED	5 RESERVED	5 RESERVED
4 RESERVED	4 RESERVED	4 RESERVED
3 RESERVED	3 RESERVED	3 RESERVED
2 RESERVED	2 RESERVED	2 RESERVED
1 RESERVED	1 RESERVED	1 RESERVED
0 RESERVED	0 RESERVED	0 RESERVED

(R/W)	(W)	(R)ECC (32-39)
70 BUFFER DATA	71 ECC CONTROL	72 ECC (16-23)
7 BD7	7 32/*48 BIT ECC	7 ECC 23/39
6 BD6	6 RESET/*PRESET ECC	6 ECC 22/38
5 BD5	5 RESERVED	5 ECC 21/37
4 BD4	4 EN SECTOR BRCH	4 ECC 20/36
3 BD3	3 CLEAR ECC	3 ECC 19/35
2 BD2	2 DISABL FEEDBACK	2 ECC 18/34
1 BD1	1 SHIFT ECC	1 ECC 17/33
0 BD0	0 SERIAL ECC IN	0 ECC 0-16/0-32

(W)	(R) ECC (40-47)	(W)
72 POLY (1-8)	73 ECC (24-31)	73 POLY (9-16)
7 POLY 8	7 ECC 31/47	7 POLY 16
6 POLY 7	6 ECC 30/46	6 POLY 15
5 POLY 6	5 ECC 29/45	5 POLY 14
4 POLY 5	4 ECC 28/44	4 POLY 13
3 POLY 4	3 ECC 27/43	3 POLY 12
2 POLY 3	2 ECC 26/42	2 POLY 11
1 POLY 2	1 ECC 25/41	1 POLY 10
0 POLY 1	0 ECC 24/40	0 POLY 9

(W) POLY (1-8)	(W) POLY (9-16)	(W) POLY (17-24)
74 POLY (17-24)	75 POLY (25-32)	76 POLY (33-40)
7 POLY 8/24	7 POLY 16/32	7 POLY 24/40
6 POLY 7/23	6 POLY 15/31	6 POLY 23/39
5 POLY 6/22	5 POLY 14/30	5 POLY 22/38
4 POLY 5/21	4 POLY 13/29	4 POLY 21/37
3 POLY 4/20	3 POLY 12/28	3 POLY 20/36
2 POLY 3/19	2 POLY 11/27	2 POLY 19/35
1 POLY 2/18	1 POLY 10/26	1 POLY 18/34
0 POLY 1/17	0 POLY 9/25	0 POLY 17/33

(W) POLY (25-31)	(R/W)	(R)
77 POLY (41-47)	78 BRANCH/NA	79 DISK STATUS
7 RESERVED	7 RESERVED	7 AM ACTIVE
6 POLY 31/47	6 RESERVED	6 DATA XFER
5 POLY 30/46	5 RESERVED	5 BRCH ACTIVE
4 POLY 29/45	4 BRCH/NA 4	4 STOPPED
3 POLY 28/44	3 BRCH/NA 3	3 RESERVED
2 POLY 27/43	2 BRCH/NA 2	2 ECC ERR
1 POLY 26/42	1 BRCH/NA 1	1 COMPARE LOW
0 POLY 25/41	0 BRCH/NA 0	0 COMPARE EQUAL

(W)		(R/W)	(W)
79 START ADDRESS		7A OP CONTROL	7B WAM CONTROL
7 RESERVED	(R/W)	7 INHIBIT CARRY	7 WAM AT BR7
6 RESERVED	(R/W)	6 RESERVED	6 WAM AT BR6
5 RESERVED	(R/W)	5 SUPRES XFER	5 WAM AT BR5
4 SEQUEN ADD 4	(R/W)	4 SRCH OP	4 WAM AT BR4
3 SEQUEN ADD 3	(R/W)	3 RESERVED	3 WAM AT BR3
2 SEQUEN ADD 2	(R)	2 NRZ DATA IN	2 WAM AT BR2
1 SEQUEN ADD 1	(R)	1 SECTOR PAST	1 WAM AT BR1
0 SEQUEN ADD 0	(R)	0 INDEX PAST	0 WAM AT BR0

(W)		(R) GENERAL
7C SYNC COMP VAL	7D RESERVED	7E PURPOSE I/O
7 SYNC VAL 7	7 RESERVED	7 RESERVED
6 SYNC VAL 6	6 RESERVED	6 RESERVED
5 SYNC VAL 5	5 RESERVED	5 OUTPUT
4 SYNC VAL 4	4 RESERVED	4 BRANCH INPUT
3 SYNC VAL 3	3 RESERVED	3 RESERVED
2 SYNC VAL 2	2 RESERVED	2 RESERVED
1 SYNC VAL 1	1 RESERVED	1 RESERVED
0 SYNC VAL 0	0 RESERVED	0 RESERVED

(R)	(W) SYNC COMPARE	(R/W) SEQUENCER
7F STACK READ	7F CNTRL	80:97 NEXT ADDR FIELD
7 STACK BIT 7	7 RESERVED	7 BRNCH COND
6 STACK BIT 6	6 RESERVED	6 BRNCH COND
5 STACK BIT 5	5 RESERVED	5 BRNCH COND
4 STACK BIT 4	4 RESERVED	4 NEXT ADDRS
3 STACK BIT 3	3 RESERVED	3 NEXT ADDRS
2 STACK BIT 2	2 SYNC CMPR CNTRL 2	2 NEXT ADDRS
1 STACK BIT 1	1 SYNC CMPR CNTRL 1	1 NEXT ADDRS
0 STACK BIT 0	0 SYNC CMPR CNTRL 0	0 NEXT ADDRS

(R/W) SEQUENCER	(R/W) SEQUENCER	(R/W) SEQUENCER
A0:B7 CONTROL FIELD	C0:D7 COUNT FIELD	E0:F7 DATA FIELD
7 RG/WG CTL 1	7 COUNT/DATA TYPE 1	7 DATA
6 RG/WG CTL 0	6 COUNT/DATA TYPE 0	6 DATA
5 CRC/ECC SELECT	5 COUNT	5 DATA
4 STACK ENABLE	4 COUNT	4 DATA
3 INVALID NRZ	3 COUNT	3 DATA
2 OUTPUT	2 COUNT	2 DATA
1 COMPRE ENABLE	1 COUNT	1 DATA
0 DATA XFER	0 COUNT	0 DATA

54 BUFFER SIZE REG

7 (W)
6 (W)
5 (W)
4 (W)
3 (W)
2 (W)
1 (W)
0 (W)

The contents of this register establish the size of buffer memory being used. If for example a 256 byte buffer is required the comparison between the Read or Write Address Pointers and the Stop Pointer is performed on the lower byte count only. The upper byte compare is masked out. This feature allows the user to utilize the memory like a segmented FIFO buffer. The table below illustrates how the bits must be programmed.

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Buffer size
0	0	0	0	0	0	0	0	256 byte
0	0	0	0	0	0	0	1	512 byte
0	0	0	0	0	0	1	1	1K byte
0	0	0	0	0	1	1	1	2K byte
0	0	0	0	1	1	1	1	4K byte
0	0	0	1	1	1	1	1	8K byte
0	0	1	1	1	1	1	1	16K byte
0	1	1	1	1	1	1	1	32K byte
1	1	1	1	1	1	1	1	64K byte

55 LATCH RAP/WAP REG

7 (W)
6 (W)
5 (W)
4 (W)
3 (W)
2 (W)
1 (W)
0 (W)

During a DMA cycle a microprocessor write to this register will, after getting synchronized with BUFFCLK, latch the contents of the RAP & WAP pointers. This will insure that the read value is not changing during a microprocessor read operation of the RAP / WAP pointers. The latcheswill reopen at the end of a microprocessor read of any RAP or WAP byte.

56 MODE SELECT REG

7 Reserved bit. *Note: During register write operation,*
6 Reserved bit. *these bits must be set*
5 Reserved bit. *to zero only for proper operation.*
4 (W) *SRAM/DRAM: If this bit is low the memory control signals provide the appropriate signals and timing for industry standard static RAMs. If this bit is high dynamic RAMs with *CAS before *RAS refresh with an internal refresh counter are supported.
3 (W) *NRZ/RLL: If this bit is low, the serial data output to disk is not encoded and is simply a non-return-to-zero type data. All the appropriate disk interface signals for supporting NRZ format data are multiplexed on chip's external pins. If this bit is high, data output to disk is encoded with a 2,7 Run Length Limited encoding scheme. Data from disk goes to the 2,7RLLI input pin and is internally decoded.
2 (W) ENBLCOMP: The assertion of this bit will enable precomp.
1 (W) VFO OPT: A control signal bit that selects one of two different VFO sync field patterns during RLL encoding. If VFO OPT is low, a 100100100100100... RLL output is generated. If VFO OPT is high, a 1000100010001000... RLL output is generated.
0 (W) AM OPT: A control signal bit that selects one of two address mark patterns used by the ENDEC. If AM OPT is low, a FFH pattern for syncing is used, if AM OPT is high, a 5EAxH pattern for syncing is used.

NOTE: All the pins on the part will remain tristated after a Power On Reset until this register is programmed. The proper pins will then be enabled as outputs according to the mode selected.

57 EARLY PCOMP DELAY

7 (W)
6 (W)
5 (W)
4 (W)
3 (W)
2 (W)
1 (W)
0 (W)

The contents of this register establish the precompensation delay required for an early write of encoded data. If precompensation is enabled the data on the 2,7RLLO output is written earlier by a delay amount which depends on the ramp rate on the WPCOMP pin and the programmed value in this register. The following table illustrates how these bits are programmed. The delay "D" is equal to an RC time constant. The rising edge of the RLL data output comes sooner by an amount equal to the delay values. The table illustrates programming examples:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	delay
----	----	----	----	----	----	----	----	-----
0	0	0	0	0	0	0	1	0.00*D
0	0	0	0	0	0	1	0	0.05*D
0	0	0	0	0	1	0	0	0.10*D
0	0	0	0	1	0	0	0	0.15*D
0	0	0	1	0	0	0	0	0.20*D
0	0	1	0	0	0	0	0	0.25*D
0	1	0	0	0	0	0	0	0.30*D
1	0	0	0	0	0	0	0	0.35*D

58 LATE PCOMP DELAY

7 (W)
6 (W)
5 (W)
4 (W)
3 (W)
2 (W)
1 (W)
0 (W)

The contents of this register establish the precompensation delay required for a late write of encoded data. If precompensation is enabled the data on the 2,7RLLO output is written late by a delay amount which depends on the ramp rate on the WPCOMP pin and the programmed value in this register. The table below illustrates how these bits are programmed. The delay "D" is equal to and RC time constant. The rising edge of the RLL data output comes later by an amount equal to the delay values. The table illustrates programming examples:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	delay
----	----	----	----	----	----	----	----	-----
0	0	0	0	0	0	0	1	0.00*D
0	0	0	0	0	0	1	0	0.05*D
0	0	0	0	0	1	0	0	0.10*D
0	0	0	0	1	0	0	0	0.15*D
0	0	0	1	0	0	0	0	0.20*D
0	0	1	0	0	0	0	0	0.25*D
0	1	0	0	0	0	0	0	0.30*D
1	0	0	0	0	0	0	0	0.35*D

59 RESET ADDRESS POINTERS REG

7 (W)
6 (W)
5 (W)
4 (W)
3 (W)
2 (W)
1 (W)
0 (W)

Writing any value to this register location will reset the Read, Write, and Stop Address Pointers to zero. Its purpose is to quickly clear these registers without any software chip reset commands.

5A Read Address Pointer 0:7

7	(R/W)	The contents of the Read Address Pointer appears on the RAM address bus if a Read operation is in process and the host has memory access. Read Address Pointer appears on the RAM address bus if a Write operation is in process and disk has memory access. In a typical memory cycle the low address byte is output first along with MAS or *RAS, while the high address byte is place on the RAM address bus next, along with *MOE or *CAS.
6	(R/W)	
5	(R/W)	
4	(R/W)	
3	(R/W)	
2	(R/W)	
1	(R/W)	
0	(R/W)	

5B Read Address Pointer 8:15

7	(R/W)	The contents of the Read Address Pointer appears on the RAM address bus if a Read operation is in process and the host has memory access. Read Address Pointer appears on the RAM address bus if a Write operation is in process and disk has memory access. In a typical memory cycle the low address byte is output first along with MAS or *RAS, while the high address byte is place on the RAM address bus next, along with *MOE or *CAS.
6	(R/W)	
5	(R/W)	
4	(R/W)	
3	(R/W)	
2	(R/W)	
1	(R/W)	
0	(R/W)	

5C Write Address Pointer 0:7

7	(R/W)	The contents of the Write Address Pointer appears on the RAM address bus if a Write operation is in process and the host has memory access. Write Address Pointer appears on the RAM address bus if a Read operation is in process and disk has memory access. In a typical memory cycle the low address byte is output first along with MAS or *RAS, while the high address byte is place on the RAM address bus next, along with *CAS.
6	(R/W)	
5	(R/W)	
4	(R/W)	
3	(R/W)	
2	(R/W)	
1	(R/W)	
0	(R/W)	

5D Write Address Pointer 8:15

7	(R/W)	The contents of the Write Address Pointer appears on the RAM address bus if a Write operation is in process and the host has memory access. Write Address Pointer appears on the RAM address bus if a Read operation is in process and disk has memory access. In a typical memory cycle the low address byte is output first along with MAS or *RAS, while the high address byte is place on the RAM address bus next, along with *CAS.
6	(R/W)	
5	(R/W)	
4	(R/W)	
3	(R/W)	
2	(R/W)	
1	(R/W)	
0	(R/W)	

5E Stop Address Pointer 0:7

7	(R/W)	The Stop Address Pointer is compared with the appropriate Address pointer (Read Address Pointer during a read operation or Write Address Pointer during a write operation) to end a DMA operation upon the transfer of the desired number of bytes. The Read and Write addresses are incremented for every byte of transfer and then compared with the Stop Pointer. If a match is found the DMA operation halts. The upper bytes of the comparator logic can be masked out by the buffer size register.
6	(R/W)	
5	(R/W)	
4	(R/W)	
3	(R/W)	
2	(R/W)	
1	(R/W)	
0	(R/W)	

5F Stop Address Pointer 8:15

7	(R/W)	The Stop Address Pointer is compared with the appropriate Read or Write Address pointer to end a DMA operation upon the transfer of the desired number of bytes. The Read and Write addresses are incremented for every byte of transfer and then compared with the Stop pointer. If a match is found the DMA operation halts. The upper bytes of the comparator logic can be masked out by the buffer size register.
6	(R/W)	
5	(R/W)	
4	(R/W)	
3	(R/W)	
2	(R/W)	
1	(R/W)	
0	(R/W)	

NOTE: The number of bytes that will be transferred before the DMA halts is equal to SP-AP+1 (SP=Stop Pointer and AP=Appropriate Address Pointer). Thus, setting SP=AP will enable a one byte DMA transfer.

60 INT MSK REG 0

7 (W) EN REQ ON INT: This bit is used in the initiator mode during microprocessor controlled PIO. When set to 1, this bit causes an interrupt to be generated when the SCSI *REQ signal is asserted. The interrupt generated when this event occurs is reflected by the SCSI REQ ON bit in STATUS REG 0 (Bit 7, Reg 60).

6 (W) EN MEM PERR INT: When this bit is set to 1, an interrupt will be generated if the EN MEM PARITY bit in CNTRL REG 0 (Bit 6, Reg 62) is set and a parity error is detected when reading from memory. The error condition causing the interrupt is reflected in the MEM PARITY ERROR bit in STATUS REG 0 (Bit 6, Reg 60), or in the DISK PARITY ERROR bit in STATUS REG 1 (Bit 3, Reg 61).

5 (W) EN SCSI PERR INT: When this bit is set to 1, an interrupt will be generated if the EN SCSI PARITY bit (Bit 7, Reg 62) is set and a parity error is detected when receiving data from the SCSI bus. The error condition causing the interrupt is reflected in the SCSI PARITY ERROR bit in STATUS REG 0 (Bit 5, Reg 60).

4 (W) EN BUS FREE: When this bit is set to 1, detection of a SCSI Bus Free Phase will set the BUS FREE DETECT bit of STATUS REG 0 (Bit 4, Reg 60) and generate an interrupt. Setting this bit to 0 will disable the BUS FREE DETECT bit in STATUS REG 0 (Bit 4, Reg 60) as well as the generation of an interrupt when a Bus Free Phase occurs.

3 (W) EN PHSE ERR/ATN ON INT: In the initiator mode if this bit is set to 1, an interrupt will be generated if the SCSI *REQ signal is asserted and the SCSI phase does not match the expected phase set in the SCSI SIG register. In the target mode if the this bit is set to 1, an interrupt will be generated if the SCSI *ATN line is asserted and the EN AUTO ATN bit is set to 0 (Bit 1, Reg 63). The interrupt generated when this event occurs is reflected by the WRONG PHASE bit in STATUS REG 0 (Bit 3, Reg 60).

2 (W) EN SEL OUT INT: When this bit is set to 1 before an arbitration phase, an interrupt will be generated when the SCSI *SEL is asserted by the chip after it has won arbitration. The interrupt may be used to start a timer for the SCSI selection time-out.

1 (W) EN RESELECTED: When this bit is set to 1, the chip will automatically respond to the reselection phase if the SCSI *SEL signal is asserted, the SCSI bus ID matches the chip's ID, there is no parity error, *I/O signal asserted and no more than two ID's are on the bus, . After the reselection is completed an interrupt is generated indicating a successful reselection. When this bit is set to 0, the chip will not respond to the reselection phase. In this mode, the processor may respond to the SCSI *SEL signal by polling the SCSI SIG register and responding appropriately.

0 (W) EN SELECTED: When this bit is set to 1, the chip will automatically respond to the selection phase if the SCSI *SEL signal is asserted, the SCSI bus ID matches the chip's ID, there is no parity error, *I/O signal not asserted and no more than two ID's are on the bus. After the selection is completed an interrupt is generated indicating a successful selection. When this bit is set to 0, the chip will not respond automatically to the selection phase. In this mode, the microprocessor may respond to the SCSI *SEL signal by polling the SCSI SIG register and responding appropriately.

60 STATUS REG 0

7 (R) SCSI REQ ON: In the initiator mode, when found to be 1, this bit indicates the SCSI *REQ signal has been asserted. If the initiator wants to start transfer after the SCSI phase has been determined, or use microprocessor controlled PIO, this bit may be used for checking the SCSI *REQ signal. This bit is reset to 0 when the SCSI *ACK signal is asserted and *REQ is deasserted. In Target mode this bit is not used.

6 (R) MEM PARITY ERROR: When this bit is found to be 1, a parity error on transfer from memory to the SCSI bus or the microprocessor has been detected. When a parity error is detected, information transfer from buffer is halted, this bit is set, and an interrupt is generated if the EN MEM PERR INT bit in INT MASK REG 0 (Bit 6, Reg 60) is set. This bit is reset, by disabling the EN MEM PARITY bit of CNTRL REG 0 (Bit 6, Reg 62). After disabling the interrupt bits they may be re-enabled again at any time.

5 (R) SCSI PARITY ERROR: When this bit is found to be 1, a parity error on transfer from SCSI has been detected. When a parity error is detected, this bit is set, and an interrupt is generated if the EN SCSI PERR INT bit in INT MASK REG 0 (Bit 5, Reg 60) is set. This bit is reset by disabling the EN SCSI PARITY bit of CNTRL REG 0 (Bit 5, Reg 62). After disabling the interrupt bits they may be re-enabled again at any time.

4 (R) BUS FREE DETECT: When this bit is found to be 1, SCSI Bus Free phase has been detected. This bit is reset to 0 when the EN BUS FREE INT bit in INT MSK REG 0 (Bit 4, Reg 60) is set to 0. In the target mode this bit is not used.

3 (R) PHASE ERR/ATN ON: In the initiator mode, if this bit is found to be 1, it indicates the SCSI *REQ signal has been asserted and the SCSI phase did not match the expected phase in the SCSI signal register. In the target mode, if this bit is found to be 1, it indicates the *ATN line is asserted and the EN AUTO ATN bit in CNTRL REG 1 (Bit 1, Reg 63) is set to 0.

2 (R) SELECT OUT: When found to be 1, the chip is driving the SCSI *SEL signal. This bit may be used to start the selection time-out timer. If the SCSI device to be selected does not respond within the selection timeout, there are two methods of cancelling the attempted selection: Setting the SCSI RST OUT bit in SCSI SIG REG (Bit 0, Reg 64), or by writing a 0 to SCSI DATA REG(Reg 65), waiting a selection abort time plus two deskew delays (~ 200 µs), checking for *BSY not asserted by reading the SCSI SIG REG, then resetting the EN ARB/SEL START bit of CNTRL REG 1 (Bit 0, Reg 63). If *BSY is asserted the SCSI RST OUT bit in SCSI SIG REG (Bit 0, Reg 64) must be set.

1 (R) RESELECTED: When set to 1, this bit indicates that the reselection phase is completed and that the chip has been reselected by another device. Masking the EN RESELECTED INT bit of INT MSK REG 0 (Bit 1, Reg 60) resets this bit.

0 (R) SELECTED: When set to 1, this bit indicates that the selection phase is completed and that the chip has been selected by another device. Masking the EN SELECTED bit of INT MSK REG 0 (Bit 0, Reg 60) resets this bit.

61 INT MSK REG 1

7 (W) EN AM ACTIVE INT: If this bit is set the corresponding bit in DISK STATUS REG (Bit 7, Reg 79) will set the interrupt flag.

6 (W) EN DATA XFER INT:If this bit is set the corresponding bit in DISK STATUS REG (Bit 6, Reg 79) will set the interrupt flag.

5 (W) EN BRCH ACT INT:If this bit is set the corresponding bit in DISK STATUS REG (Bit 5, Reg 79) will set the interrupt flag.

4 (W) EN STOPPED INT:If this bit is set the corresponding bit in DISK STATUS REG (Bit 4, Reg 79) will set the interrupt flag.

3 (W) EN ECC ERR INT:If this bit is set the corresponding bit in DISK STATUS REG (Bit 2, Reg 79) will set the interrupt flag.

2 (W) EN COMP EQ INT:If this bit is set the corresponding bit in DISK STATUS REG (Bit 0, Reg 79) will set the interrupt flag.

1 (W) EN CMD DONE: When set to 1, completion of any one of the following commands will generate an interrupt and set the CMD DONE bit in STATUS REG 1 (Bit 1, Reg 61): MEMORY or SCSI PIO, DMA transfer, or the successful completion of a Selection or Reselection Phase following an Arbitration Phase. When this bit is set to 0, both the CMD DONE bit in STATUS REG 1 (Bit 1, Reg 61) and the generation of an interrupt by the CMD DONE condition will be disabled.

0 (W) EN RST OCCR INT: When set to 1, an interrupt will be generated when a SCSI *RST condition occurs. A SCSI reset must be at least 500 ns in duration. The SCSI *RST signal will be latched in the SCSI RESET OCCURRED bit of STATUS REG 1 (Bit 0, Reg 61) and may only be cleared by asserting *POR, doing a software reset, or setting the CLR RST OCCRD bit in CNTRL REG 1 (Bit 5, Reg 63).

61 STATUS REG 1

7 Reserved.

6 Reserved.

5 Reserved.

4 Reserved.

3 (R) DISK PARITY ERR: When this bit is found to be 1, a parity error on transfer from memory to disk has been detected. When a parity error is detected, this bit is set and an interrupt is generated if the EN MEM PERR INT bit in INT MASK REG 0 (Bit 6, Reg 60) is set. This bit is reset by setting the EN MEM PARITY bit in CNTRL REG 0 (Bit 6, Reg 62) to 0.

2 (R) SCSI RESET IN: When set to 1, this bit indicates that the SCSI *RST signal is asserted. This bit is set to 1 only as long as the SCSI *RST signal is asserted on the SCSI bus. When the SCSI reset occurs, this bit must be checked before chip initialization begins, to verify that the SCSI RST signal is still not asserted on the bus.

1 (R) CMD DONE: When set to 1, this bit indicates that one of the three following commands has been completed: DMA transfer, a memory/SCSI PIO, Arbitration followed by a Selection or Reselection sequence. This bit will be set to 0 by setting EN CMD DONE in INT MSK REG 1 (Bit 1, Reg 61) to 0. The EN CMD DONE bit should then be set back to 1 to re-enable the command done function.

0 (R) SCSI RESET OCCURRED: When set to 1, this bit indicates that a SCSI *RST condition has occurred. The reset condition will be latched, resetting the chip's SCSI interface circuitry and releasing all SCSI bus signals (except for the SCSI *RST signal, which will be asserted if the SCSI RST OUT bit in the SCSI SIG REG (Bit 0, Reg 64) is set to 1). The latched reset condition can be cleared by setting the CLR RST OCCRD bit in CNTL REG 1 (Bit 5, Reg 63), asserting *POR or doing a software reset.

62 CNTRL REG 0

7 (W) EN SCSI PARITY: When set to 1 the parity checking logic on the data from SCSI to the chip is enabled. If the corresponding interrupt mask bit is also enabled, an interrupt is generated upon the detection of an error.

6 (W) EN MEM PARITY: When set to 1 the parity checking logic on the data from memory to the chip is enabled. If the corresponding interrupt mask bit is also enabled, an interrupt is generated upon the detection of an error.

5 (W) SOFTWARE RESET: When set to 1, the whole chip will be reset. This bit is automatically set after a power on reset, *POR, and will remain set until set to 0 by the external microprocessor.

4 (W) 20 MHz BUFF CLK: This bit must be set to 1 when the BUFFCLK frequency is greater that 20 MHz. The chip uses the BUFFCLK to determine the timing on the SCSI bus during the arbitration and selection/reselection phases. During synchronous data transfer the data transfer rate is determined by the BUFFCLK frequency.

3 (W) 10 MHz BUFF CLK: This bit must be set to 1 when the BUFFCLK frequency is greater that 10 MHz. The chip uses the BUFFCLK to determine the timing on the SCSI bus during the arbitration and selection/reselection phases. To insure proper timing and maximum performance during arbitration and selection/reselection, this bit should be set to 0 when the BUFFCLK is 10 MHz, and to 1 when the BUFFCLK is 20 MHz. If the BUFFCLK is greater than 20 MHz, the 20 MHz BUFFCLK bit must be set to 1.

2 (W) SCSI ID 2: chip's own encoded SCSI ID bit 2.

1 (W) SCSI ID 1: chip's own encoded SCSI ID bit 1.

0 (W) SCSI ID 0: chip's own encoded SCSI ID bit 0.

63 CNTRL REG 1

7 (W) EN MEM PIO: When this bit is set to 1, an automatic memory transfer cycle occurs between the buffer and the BUFFER DATA REG (Reg 70). This gives the microprocessor access to the buffer memory through an internal data register (BUFFER DATA REG (Reg 70)). Upon the completion of a memory PIO transfer the CMD DONE bit (Bit 1, Reg 61) is set and an interrupt is generated if the EN CMD DONE bit in INT MASK REG 1 (Bit 1, Reg 61) is set , and this bit is automatically reset.

6 (W) MEM PIO *WR/RD: This bit establishes the direction of memory PIO transfer. If set to 0, the contents of the Write Address Pointer are used to write the data in BUFFER DATA REG to the buffer. If set to 1, the Read Address Pointer is used to read the data from memory and latch it into BUFFER DATA REG (Reg 70).

5 (W) CLR RST OCCRD: Setting this bit will clear the SCSI RST OCCURRED bit in STATUS REG 1 (Bit 0, Reg 61), and the corresponding interrupt. This bit must be reset to 0 to resume normal operation of the SCSI RST OCCURRED status bit and the corresponding interrupt.

4 (W) ROP/*WOP: This bit establishes the direction of DATA transfers with respect to the disk. If set to 0, a write operation to disk is selected. The data from the SCSI bus (host) is written to the buffer using the Write Address Pointer, while the data is read from the buffer and written to the disk using the Read Address Pointer. If set to 1, a read operation from disk is desired. The data is written to the buffer from the disk using the Write Address Pointer, and read from the buffer and output to the SCSI bus using the Read Address Pointer.

3 (W) EN DMA: This bit is set to initiate the DMA operation automatically. The DMA transfer will be completed when the Read Address Pointer matches the Stop Pointer during a read operation or Write Address Pointer matches the Stop Pointer during a write operation. Upon completion of the DMA transfer, the CMD DONE bit (Bit 1, Reg 61) is set and an interrupt is generated if the EN CMD DONE bit in INT MASK REG 1 (Bit 1, Reg 61) is set , and this bit is automatically reset.

2 (W) EN SCSI PIO: This bit is similar to the EN MEM PIO bit. When it is set to 1, an automatic SCSI handshake is performed. This bit is reset after the handshake is completed. Automatic PIO requires the microprocessor to read data from the SCSI LATCHED DATA REG (Reg 67) and write data to the SCSI DATA REG (Reg 65). Upon completion of a SCSI PIO transfer the CMD DONE bit (Bit 1, Reg 61) is set and an interrupt is generated if the EN CMD DONE bit in INT MASK REG 1 (Bit 1, Reg 61) is set , and this bit is automatically reset.

1 (W) EN AUTO ATN: When this bit is set to 1 in the initiator mode, any of the following conditions will cause the SCSI *ATN signal to be automatically asserted: *SEL is asserted, a SCSI parity error exists, or the SCSI ATN signal has been set in the SCSI SIG REG (Reg 64). The SCSI *ATN signal is reset when this bit is turned off and the SCSI *REQ is asserted.

0 (W) EN ARB/SEL: When set to 1, the chip will wait for a bus free phase, then automatically proceed to an arbitration phase. If arbitration is won, it will automatically proceed to the SCSI selection phase if the SCSI I/O bit in the SCSI SIG REG (Bit 6, Reg 64) is set to 0, or the reselection phase if the SCSI I/O bit is

set to 1. If arbitration is lost, it will go back to waiting for a bus free phase, then repeat the same sequence of events. This bit is reset to zero if a selection/reselection phase is completed successfully, or the microprocessor resets this bit.

64 SCSI SIG REG

7 (R/W) SCSI C/D: When this bit is set in the target mode, the SCSI *C/D signal will be asserted by the chip. When this bit is read and found to be 1, the SCSI *C/D signal is asserted on the bus.

6 (R/W) SCSI I/O: When this bit is set in the target mode, the SCSI *I/O signal will be asserted by the chip. When this bit is read and found to be 1, the SCSI *I/O signal is asserted on the bus.

5 (R/W) SCSI MSG: When this bit is set in the target mode, the SCSI *MSG signal will be asserted by the chip. When this bit is read and found to be 1, the SCSI *MSG signal is asserted on the bus.

4 (R/W) SCSI ATN: When this bit is set in the Initiator mode, the SCSI *ATN signal will be asserted by the chip if the EN AUTO ATN bit in CNTRL REG 1 (Bit 1, Reg 63) is set to 1. When this bit is read and found to be 1, the SCSI *ATN signal is asserted on the bus.

3 (R/W) SCSI SEL: When this bit is set, the SCSI *SEL signal will be asserted by the chip. When this bit is read and found to be 1, the SCSI *SEL signal is asserted on the bus.

2 (R/W) SCSI BSY: When this bit is set, the SCSI *BSY signal will be asserted by the chip. When this bit is read and found to be 1, the SCSI *BSY signal is asserted on the bus.

1 (R) SCSI REQ: When this bit is read and found to be 1, the SCSI *REQ signal is asserted on the bus.

1 (W) SCSI REQ/ACK: When this bit is set in the target mode, the SCSI *REQ signal will be asserted by the chip. When the bit is set in the initiator mode, the SCSI *ACK signal will be asserted by the chip.

0 (R) SCSI ACK: When this bit is read and found to be 1, the SCSI *ACK signal is asserted on the bus.

0 (W) SCSI RST OUT: When this bit is set to 1, the SCSI *RST signal will be asserted on the SCSI bus, all other SCSI signals will be deasserted and the SCSI RST OCCURRED condition will be internally latched. The SCSI *RST signal will continue to be asserted on the bus until this bit is set to 0.

65 SCSI DATA REG

7 (R/W) The microprocessor must store both source and
6 (R/W) destination IDs in this register before the SCSI
5 (R/W) arbitration and selection/reselection phases. After the
4 (R/W) arbitration and selection or reselection sequence is
3 (R/W) complete, this register may be used to read or write
2 (R/W) data from or to the SCSI bus. Using the microprocessor
1 (R/W) controlled PIO method of data transfer, this register
0 (R/W) would be used in conjunction with the SCSI signal register to transfer data on the SCSI bus.

66 SOURCE/DEST ID REG

7 (R) After the chip gets Selected or Reselected,
6 (R) this register contains the source and the destination
5 (R) IDs. The microprocessor should read this register after
4 (R) a selection (Target) or reselection (Initiator) phase
3 (R) to determine the source ID.
2 (R)
1 (R)
0 (R)

67 SCSI LATCHED DATA REG

7 (R) Reading this register will read the data which is
6 (R) latched during the SCSI *REQ/*ACK cycle. In the target
5 (R) mode, the data is clocked when the *ACK signal is
4 (R) asserted. In the initiator mode, the data is clocked
3 (R) when the *REQ signal is asserted. This register can be
2 (R) read to find out the data transferred from the SCSI bus
1 (R) during an automatic PIO. The data to be sent on the
0 (R) SCSI bus during an automatic PIO must be in the SCSI DATA register.

67 SCSI BSY RST REG

7 (W) In the target mode, writing anything to this register will reset the SCSI
6 (W) *BSY signal and the SCSI bus will enter the bus free phase.
5 (W)
4 (W)
3 (W)
2 (W)
1 (W)
0 (W)

68 OFFSET CNTR REG

7 (W) Reserved.
6 (W) SYNC RATE 6:4 bits in combination with the BUFFCLK
5 (W) frequency establish the SCSI bus synchronous transfer
4 (W) rate. The formula below determines the transfer rate:
If BUFFCLK < = 20 MHz
Synchronous speed = (4 * T) + (SYNC RATE 2:0 * T)
If BUFFCLK > 20 MHz
Synchronous speed = (6 * T) + (SYNC RATE 2:0 * T)

For example a 20 MHz clock with a period of 50 ns, and a SYNC RATE equal seven yields a speed of 550 ns/byte, or an equivalent transfer rate of 1.82 Mbytes/second.

3 (W) OFFSET BIT 3:0 bits set the DMA transfer offset count.
2 (W) An offset count of zero allows asynchronous transfer
1 (W) only. An offset count of one to seven provides SCSI bus
0 (W) synchronous transfer. The offset count must always be set to distinguish between asynchronous and synchronous transfer when DMA is enabled.

69 FIFO STATUS REG

7 Reserved.
6 Reserved.
5 (R) OFFSET CNT ZERO: When this bit is set to 1, it indicates that there are no pending SCSI *ACK's. In conjunction with the FIFO EMPTY status bit, this bit can be used to qualify a disconnection from the SCSI bus.
4 (R) FIFO EMPTY: This bit indicates that the FIFO is empty and no other data is present in the FIFO.
3 (R) FIFO FULL: This bit indicates that the FIFO is full with eight bytes of data and can not take in any more data.
2 (R) FIFO CNT BIT 2:0 give the binary count of the number of bytes remaining in the
1 (R) FIFO. This count in conjunction with the Address pointer value will determine the
0 (R) number of bytes transferred across the SCSI bus. NOTE that these three bits will be zero when FIFO is full or empty.

70 BUFFER DATA REG

7 (R/W) To access the buffer memory, the microprocessor must go
6 (R/W) through the Buffer Data register. To write data to memory, this register is
5 (R/W) loaded with the appropriate data and the memory PIO
4 (R/W) write command issued. Upon writing the data to memory,
3 (R/W) the CMD DONE status alerts the microprocessor of a
2 (R/W) successful transfer. To read data from the memory, the
1 (R/W) PIO read command is issued. Upon completion of the transfer, the CMD DONE
0 (R/W) status bit is set and the microprocessor must read this register to access to the data.

71 ECC CONTROL REG

7 (W) 32/*48 BIT ECC: If this bit is 0, the 48-bit programmable ECC polynomial is enabled. If this bit is 1, the 32-bit programmable ECC polynomial is enabled.

6 (W) RESET/*PRESET ECC: This bit defines the initial state of the ECC and CRC polynomials during read or write operations. If set to 0, all the registers are preset, if set to 1, all the registers are reset.

5 Reserved.

4 (W) EN SECTOR BRCH: This bit controls the multiplexing on the SECTOR/RESET1S input pin. It should be set to 1 in the hard-sectored mode and set to 0 in the soft-sectored mode. When set to 1, this bit enables branching on sector by ORing together the SECTOR and INDEX inputs to the Disk Controller block, and disables the RESET1S input to the ENDEC block. When set to 0, this bit enables the RESET1S input to reset the internal counter in the ENDEC block, and disables the SECTOR input to the Disk Controller block.

3 (W) CLEAR ECC: While this bit is on, the ECC shift register will be reset or preset based on the state of bit 6 in this register.

2 (W) DISABLE FEEDBACK: When set, causes the ECC polynomial to function as a simple shift register.

1 (W) SHIFT ECC: Each time this bit is set, a single shift pulse will be sent to the ECC registers. This bit is automatically cleared after the shift pulse occurs.

0 (W) SERIAL ECC IN: ECC bit 0 will be loaded with the contents of this bit when the shift control bit is set. Note that the Read Reference Clock (RCC) must be cycling. RG and WG must be inactive and feedback must be disabled.

72 ECC (32-39) ECC(16-23) REG

7 (R) Bits 1 through 7 contain the 48 bit ECC syndrome bits
6 (R) 33 through 39 respectively, if in the 48 bit ECC mode.
5 (R) In the 32 bit ECC mode these bits contain syndrome bits
4 (R) 17 through 23 respectively.
3 (R)
2 (R)
1 (R)
0 (R) Bit 0 is an OR of the ECC syndrome bits 0 through 32 in the 48 bit ECC mode. In the 32 bit ECC mode, this bit is an OR of the ECC syndrome bits 0 through 16.

72 POLY (1-8) REG

7 (W) This register contains bits 1 to 8
6 (W) for the 48 bit ECC polynomial. Bit 7 of this register
5 (W) corresponds to bit 8 of the 48 bit polynomial.
4 (W) Each bit corresponds to a feedback path being enabled.
3 (W) If for example, if bit 0 in this register (i.e. POLY Bit 1) is set to 1, the
2 (W) feedback (ECC output bit 47 XORed with serial data)
1 (W) will be XORed with ECC bit 0 and the result is the
0 (W) input to ECC bit 1.

73 ECC(40-47) ECC(24-31) REG

7 (R) In the 48 bit ECC mode, ECC syndrome bits 40 through 47
6 (R) are contained in bits 0 through 7 of this register
5 (R) respectively. In the 32 bit ECC mode, ECC syndrome bits
4 (R) 24 through 31 are contained in bits 0 through 7 of this
3 (R) register respectively.
2 (R)
1 (R)
0 (R)

73 POLY (9-16) REG

7 (W) This register contains bits 9-16
6 (W) for the 48 bit ECC polynomial. Bit 7 of this register
5 (W) corresponds to bit 16 of the 48 bit polynomial.
4 (W) Each bit corresponds to a feedback path being enabled.
3 (W) If for example, if bit 0 in this register (i.e. POLY Bit 9) is set to 1, the
2 (W) feedback (ECC output bit 47 XORed with serial data)
1 (W) will be XORed with ECC bit 8 and the result is the
0 (W) input to ECC bit 9.

74 POLY (1-8) POLY(17-24) REG

7 (W) In the 32 bit ECC mode, this register represents
6 (W) polynomial bits 1 through 8 corresponding to bits 0
5 (W) through 7 of this register respectively.
4 (W) In the 48 bit ECC mode, this register represents
3 (W) polynomial bits 17 through 24 corresponding to bits 0
2 (W) through 7 of this register respectively.
1 (W)
0 (W)

75 POLY (9-16) POLY(25-32) REG

7 (W) In the 32 bit ECC mode, this register represents
6 (W) polynomial bits 9 through 16 corresponding to bits 0
5 (W) through 7 of this register respectively.
4 (W) In the 48 bit ECC mode, this register represents
3 (W) polynomial bits 25 through 32 corresponding to bits 0
2 (W) through 7 of this register respectively.
1 (W)
0 (W)

76 POLY (17-24) POLY(33-40) REG

7 (W) In the 32 bit ECC mode, this register represents
6 (W) polynomial bits 17 through 24 corresponding to bits 0
5 (W) through 7 of this register respectively.
4 (W) In the 48 bit ECC mode, this register represents
3 (W) polynomial bits 33 through 40 corresponding to bits 0
2 (W) through 7 of this register respectively.
1 (W)
0 (W)

77 POLY (25-31) POLY(41-47) REG

7 Reserved.
6 (W) In the 32 bit ECC mode, this register represents
5 (W) polynomial bits 25 through 31 corresponding to bits 0
4 (W) through 6 of this register respectively.
3 (W) In the 48 bit ECC mode, this register represents
2 (W) polynomial bits 41 through 47 corresponding to bits 0
1 (W) through 6 of this register respectively.
0 (W)

78 BRANCH/NA REG

7 Reserved.
6 Reserved.
5 Reserved.
4 (R/W) Writing bits 0 through 4 will cause the sequencer RAM
3 (R/W) to jump to this address when a branch condition is met.
2 (R/W) A read of this register gives the next address the
1 (R/W) sequencer will execute.
0 (R/W)

79 DISK STATUS REG

7 (R) AM ACTIVE: This bit is set by reading or writing an AM or SYNC byte, and is reset by reading or writing the ECC bytes. This bit is also reset by a stopped condition.

6 (R) DATA XFER: This bit is on whenever data is being transferred either to or from the buffer memory. This is the data transfer enable bit of the sequencer RAM.

5 (R) BRCH ACTIVE: This bit is set whenever a branch condition is met. This bit is reset by a read of this register.

4 (R) STOPPED: This bit is set when the sequencer is stopped. In the stopped state, the sequencer RAM is at address $1F_H$, the ECC contents have not been reset, RG and WG are reset, and the bit ring is running.

3 Reserved (Will be zero).

2 (R) ECC ERR: After the last bit of ECC is read, this bit is set to 1 if any of the ECC bits is 1, otherwise it is set to 0.

1 (R) COMPARE LOW: The state of the compare operation, as a result of all bytes that were read from the disk while the COMPARE ENABLE bit in the CONTROL FIELD of the sequencer RAM was set to 1. The comparison is done between the memory data or DATA FIELD byte of the sequencer RAM, and the read data from the disk. A COMPARE LOW condition occurs when the byte that is read from the disk is less than the sequencer RAM DATA FIELD byte (if DATA XFER is off) or the data from memory (if DATA XFER is on). This bit is valid only after the ECC bytes have been read in.

0 (R) COMPARE EQUAL: The state of the compare operation, as a result of all bytes that were read from the disk while the COMPARE ENABLE bit in the CONTROL FIELD of the sequencer RAM was set to 1. The comparison is done between the memory data or DATA FIELD byte of the sequencer RAM, and the read data from the disk. The final result is based on whether all enabled bytes compare. The value of this bit is valid only after the ECC bytes have been read in.

79 START ADDRESS REG

7 Reserved.
6 Reserved.
5 Reserved.
4 (W) Sequencer start address bits 0-4 : A write to this
3 (W) register will start the sequencer at the specified
2 (W) address. This register may only be set when the STOPPED
1 (W) bit of register 79_H is active, indicating that the
0 (W) sequencer is in the correct initial state.

NOTE: The starting or the stopping of the sequencer may take from 0 to 8 RRC cycles. In normal operation it should not be necessary to stop the sequencer by setting a $1F_H$ to the START ADDRESS REG (Reg 79_H). However, when a sync character is missed and the microprocessor timer expires, the AIC-6110 should be stopped and restarted to retry the operation. The recommended way to do this is in a loop that sets a $1F_H$ to the START ADDRESS REG and examine the STOPPED Bit (Bit 4, Reg 79_H). If the sequencer is not stopped, continue to loop. The AIC-6110 will typically stop the first time, but two or three loop cycles may be required.

7A OP CONTROL REG

7 (R/W) INHIBIT CARRY: When set, the carry/load of the sequencer during a DATA TRANSFER will be inhibited. After a carry has occurred this bit will be reset. This is used to cause the sequencer to execute the present address again. For this time, however, the count field will start at 00_H and count down (256 cycles will be executed). For example, if the sequencer is executing a DATA TRANSFER having a COUNT FIELD of 03_H, that instruction will be executed 4 times if INHIBIT CARRY = Ø, or 260 times if INHIBIT CARRY = 1.

6 (R) Reserved.

5 (R/W) SUPPRESS XFER: When this bit is set, it will overide the DATA TRANSFER bit in the Sequencer RAM and disable transfers to/from memory. Thus during WG, the data field will be written with the contents of the sequencer RAM data, or during RG compared with the contents of sequencer RAM data.

4 (R/W) SRCH OP: This bit must be set whenever a DATA FIELD compare between memory data and the read data from the disk is required. For proper functioning of the search operation, the buffer controller should be set up for a write operation by setting the *WOP/ROP bit in CNTRL REG 1 (Bit 4, Reg 63) to 0.

3 (R) Reserved. Will be high at all times.

2 (R) NRZ DATA IN: An input of 1 on the NRZ input to the disk controller has occurred since the last time this register was accessed.

1 (R) SECTOR PAST: Sector pulse has been received from the device since the last read of this register. Reading this bit while sector is present does not reset this bit.

0 (R) INDEX PAST: Index point from the device has been detected since the last time this register was read. Reading this bit while index is present does not reset this bit.

7B WAM CONTROL REG

7 (W) Write Address Mark Control 0-7: The *WAM/*AMD pin will go
6 (W) active for each bit cell time corresponding to the bits
5 (W) set in this register during a write address mark
4 (W) operation. This is used to indicate to external logic
3 (W) when the clock pulse should be deleted from the
2 (W) outgoing data stream after encoding (*Note: This is only*
1 (W) *used if the 2,7 RLL ENDEC is not being utilized*). This
0 (W) then creates an illegal pattern which becomes the address mark.

7C SYNC COMPARE Value REG

7 (W) Sync Detect Control 0-7: After RG is turned on and the Bit Ring is stopped, a match
6 (W) between this register and the serial NRZ read data input will cause a sync
5 (W) detect (if *AMD input is active), the Bit Ring to restart
4 (W) at zero, and data to be gated into the ECC.
3 (W)
2 (W)
1 (W)
0 (W)

*NOTE: Only those bits enabled by SYNC COMPARE CNTRL REG (Reg 7F) can be set for comparison. Any don't care bits must be set to zero in this register. Whenever RG is turned on, the Bit Ring Oscillator stops within two byte times. For this reason, the sequencer word which turns on RG must have a COUNT FIELD of 01_H. At this point, the AIC-6110 starts to shift in the data on the NRZ pin (pin 23, 68 pin PLCC). The bit stream is compared with the contents of this register , in order to obtain byte sync. In addition to byte sync, an AM detect must also be observed on the *AMD pin for the Bit Ring Oscillator to start up. This comparison is independent of the Compare Enable function of the sequencer RAM, which is normally used for additional qualification such as differentiating between ID Address Mark and Data Address Mark.*

Some sync characters like FE_H can cause problems if the AIC-6110 is not set-properly for that particular function. This is due to the floating NRZ data line (when RG and WG are both deasserted) which shifts 1's into the internal sync comparison register. When RG is turned on over the preamble, an FE_H is framed, causing improper alignment. This can be avoided by first turning on the INVALID NRZ CONTROL bit (Bit 3, CONTROL FIELD) along with RG for a few byte times (this can be accomplished with a COUNT FIELD of 01_H). After this, the next word should also have a COUNT FIELD of 01_H, followed by the sync word.

7E GENERAL PURPOSE I/O REG

7 Reserved.
6 Reserved.
5 (R) Output pin: This bit indicates the state of the output bit in the sequencer RAM.
4 (R) Input pin: This bit indicates the state of the input pin, an external input that may be used as a branch condition for the sequencer.
3 Reserved. Will be low at all times.
2 Reserved. Will be low at all times.
1 Reserved. Will be low at all times.
0 Reserved. Will be low at all times.

7F STACK READ REG

7 (R) Stack bits 0-7: A read of this register will read and
6 (R) pop the top of the 8-byte stack. The stack will wrap
5 (R) around on the 8th pop.
4 (R)
3 (R)
2 (R)
1 (R)
0 (R)

7F SYNC COMPARE CNTRL REG

7 Reserved.
6 Reserved.
5 Reserved.
4 Reserved.
3 Reserved.
2 (W) Sync compare control bits 0 through 2 specify the
1 (W) number of bits to be used in the compare for the sync
0 (W) byte. The table below illustrates the use:

000 = No compare.
001 = Bit 7 compared.
010 = Bits 7,6 compared.
011 = Bits 7,6,5 compared.
100 = Bits 7,6,5,4 compared.
101 = Bits 7,6,5,4,3 compared.
110 = Bits 7,6,5,4,3,2 compared.
111 = All bits compared.

80:97 SEQUENCER NEXT ADDR FIELD

7 (R/W) Branch conditions when both ECC and read gate are
6 (R/W) active. These branches are taken at the end of ECC
5 (R/W) time.

000 = Continue, next address
001 = Stop on ECC error
010 = Stop on no compare equal
011 = Stop on no compare equal or ECC error
100 = Branch on good ECC and compare equal
101 = Branch on ECC error
110 = Branch on no compare equal
111 = Branch on no compare equal or ECC error

Branch conditions at all other times. These branches are taken when the operations specified by that word are complete.

000 = Continue, next address
001 = Stop if INPUT is active
010 = Stop if INDEX or SECTOR is active
011 = Stop on no compare equal
100 = Branch on carry (at expiration of COUNT FIELD)
101 = Branch on INPUT active
110 = Branch on INDEX or SECTOR active
111 = Branch on no compare equal

4 (R/W) Next Address 0-4: This is the address the sequencer will go to after the down
3 (R/W) counter has reached zero and
2 (R/W) a branch has not been taken. There are 24 possible next
1 (R/W) address locations. Addresses from 18_H to $1F_H$ establish
0 (R/W) the stopped condition.

A0:B7 SEQUENCER CONTROL FIELD

7 (R/W) Bits 7 and 6 control the operation of Read Gate (RG),
6 (R/W) and Write Gate (WG).

00 = No operation.
01 = Set Read Gate (RG): RG signal will be turned on when the sequencer word with this bit is executed. The RG latch will be set at Bit Ring 2 and reset at the end of ECC or when the sequencer goes to a stopped state. RG latch will not be set if WG is already on.

10 = Set Write Gate (WG): WG signal will be turned on when the sequencer word with this bit on is executed. The WG latch will be set at Bit Ring 4 time. After this is set, WG will be reset by asserting the Reset WG bits (Bits 7:6, CONTROL FIELD) or when the sequencer goes to a stopped state.
11 = Reset WG: Used to turn off WG. The WG latch will be reset at Carry and Bit Ring 4 time.

5 (R/W) CRC SELECT: If this bit is 0, the normal ECC function is selected. If this bit is set to 1, the 16-bit fixed CRC polynomial is used instead.
4 (R/W) STACK ENABLE: When on, read data is pushed onto 8-byte stack.
3 (R/W) INVALID NRZ CONTROL: When set with RG, this bit will block the NRZ data input and disable sync detection. This is used to allow VFO phase up.
2 (R/W) OUTPUT: This bit goes to the OUTPUT pin (pin 17, 68 pin PLCC), and is used to synchronize external logic functions with the sequencer.
1 (R/W) COMPARE ENABLE: When active along with RG, will allow a comparison between read data from the disk and sequencer RAM data, or memory data (if the SRCH OP bit (Bit 4, Reg 7A) is set to 1, and DATA TRANSFER is enabled).
0 (R/W) DATA TRANSFER: When set, the disk request signal to the buffer controller will be generated and data transfer to or from memory will take place.

C0:D7 SEQUENCER COUNT FIELD

7 (R/W) DATA TYPE 1: When the data transfer bit of the
6 (R/W) DATA TYPE 0: sequencer RAM is off, these bits are decoded for data
00 = Normal.
10 = Address Mark.
01 = ECC.
COUNT 7-6: When the DATA TRANSFER bit of the sequencer RAM is on these 2 bits are the most significant bits of the count field.

5 (R/W) COUNT 5 : When DATA TRANSFER is enabled, this is bit 5 of the COUNT FIELD.
4 (R/W) COUNT bits 4-0: These are the 5 least significant bits of the counter.
3 (R/W) These bits in addition to the 3 bits above give an eight bit counter that
2 (R/W) can count up to 256. The counter is decremented on Bit ring 7. When
1 (R/W) reaches zero a new state will be accessed from the sequencer RAM.
0 (R/W) The value specified here must be one less than the count to be executed.

E0:F7 SEQUENCER DATA FIELD

7 (R/W) DATA bits 0-7: This register is the source for all
6 (R/W) overhead data bytes used by the device during write
5 (R/W) operations. During read operations, it is one of the
4 (R/W) operands to the comparison logic. When DATA XFER is on
3 (R/W) with WG, the source of write data will be the external memory.
2 (R/W) However, when suppress transfer (SUPRES XFER (Bit 5, Reg 7A))is on with
1 (R/W) WG, this register will again be the source for write
0 (R/W) data.

7.1 INTRODUCTION TO THE AIC-6110 FUNCTIONAL OPERATION

The functional block diagram outlines the five functional blocks that make up the AIC-6110 Single Chip Synchronous SCSI storage controller. The five blocks are the microprocessor interface, the SCSI interface, the buffer controller, the disk controller, and the 2,7 RLL ENDEC. The functional operation section ties each of these blocks together to demonstrates how the AIC-6110 operates. The functional operation of the AIC-6110 can be divided into four major modules :

- Microprocessor access and AIC-6110 initialization.
- The SCSI interface block operation and its interface to the buffer controller.
- The Disk Controller block operation and its interface to the buffer controller.
- Operation of the 2,7 RLL ENDEC with programmable precompensation and interface to the data separator.

The following sections describe each of these modules in detail with flowcharts and descriptions.

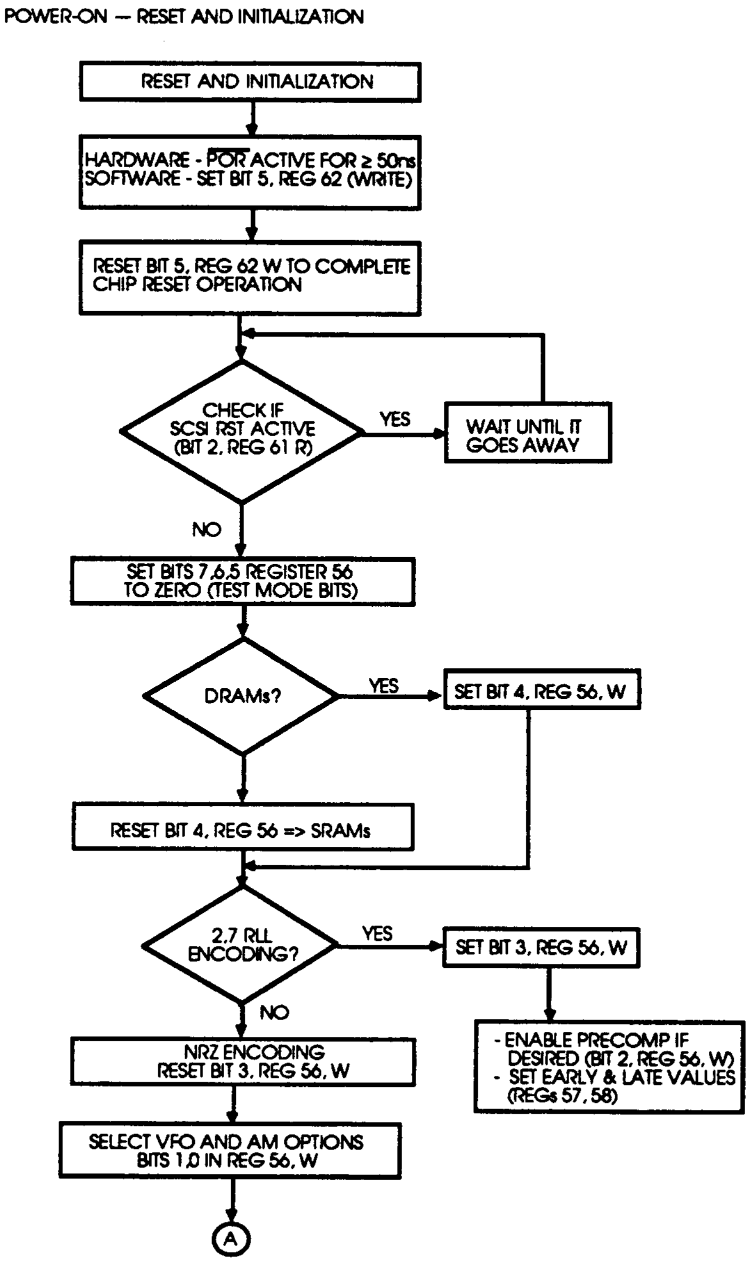

POWER-ON — RESET AND INITIALIZATION
RESET AND INITIALIZATION
HARDWARE - POR ACTIVE FOR ≥ 50ns
SOFTWARE - SET BIT 5, REG 62 (WRITE)
RESET BIT 5, REG 62 W TO COMPLETE CHIP RESET OPERATION
CHECK IF SCSI RST ACTIVE (BIT 2, REG 61 R)
YES
WAIT UNTIL IT GOES AWAY
NO
SET BITS 7,6,5 REGISTER 56 TO ZERO (TEST MODE BITS)
DRAMs?
YES
SET BIT 4, REG 56, W
RESET BIT 4, REG 56 => SRAMs
2,7 RLL ENCODING?
YES
SET BIT 3, REG 56, W
- ENABLE PRECOMP IF DESIRED (BIT 2, REG 56, W)
- SET EARLY & LATE VALUES (REGs 57, 58)
NO
NRZ ENCODING RESET BIT 3, REG 56, W
SELECT VFO AND AM OPTIONS BITS 1,0 IN REG 56, W
A

FIGURE 3.PROGRAMMING FLOWCHART

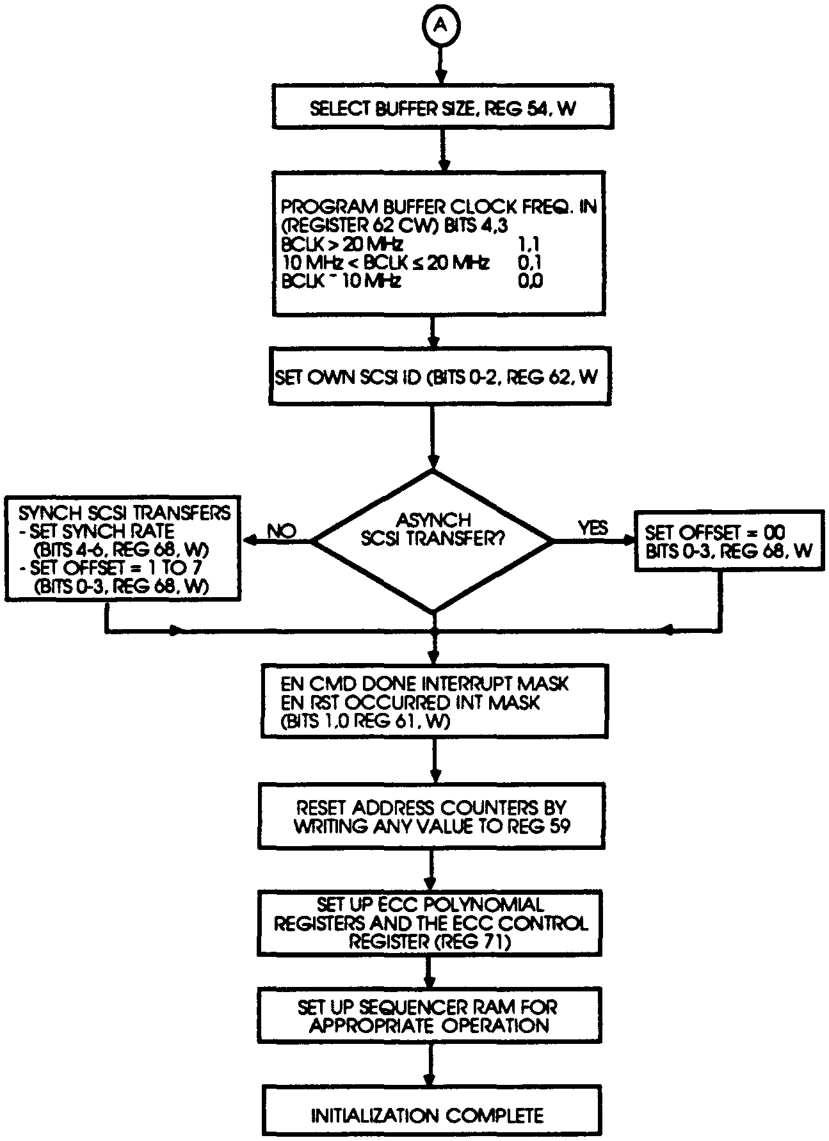


FIGURE 3. PROGRAMMING FLOWCHART CONTINUED

7.2 MICROPROCESSOR ACCESS AND AIC-6110 INITIALIZATION

7.2.1 Microprocessor Access

The AIC-6110 requires microprocessor control to initiate operations and commands and to check the status of the operation. All the registers in the AIC-6110 appear as unique memory or I/O locations and can be randomly accessed and operated upon. These register definitions are also compatible with the register definitions of the individual Adaptec chips (AIC-6250, AIC-011, AIC-270 and the AIC-300), which are integrated with in the AIC-6110.

The AIC-6110 has a separate 8-bit multiplexed bus for the microprocessor, allowing access to any register even while data transfer is occurring. Address information is presented on the AD bus which can be latched internally using the ALE signal with the CS signal active. Having latched in the address of the register, it can be accessed by asserting a *RD or *WR strobe, with the CS active. The specific timing details of the microprocessor interface are shown in section 8.3. It is important to note that most registers are read only or write only. Some registers however change function depending on whether they are being read or written.

NOTE: The sequencer RAM is setup to be accessed on the next cycle, if CS and ALE and the address line A7 are active. If the RAM is not accessed, then its contents may be corrupted. Hence it must be ensured that this condition be present only when the microprocessor desires to access the sequencer RAM.

7.2.2 AIC-6110 Power-On-Reset and Initialization

The local microprocessor controls the AIC-6110 power-on reset and initialization. The AIC-6110 can be reset under hardware control, by holding the *POR signal active for at least 50 ns or through firmware using the software reset bit in Control Register 0, (bit 5, Reg 62, W). The internal reset latch remains active after either a hardware or software reset and must be cleared by the microprocessor to put the AIC-6110 in operating mode, (bit 5, Reg 62, W). When a Power-On-Reset is done all outputs will be tristated and will remain in this condition until the Mode Select register is initialized. Once the reset operation is completed, the microprocessor needs to initialize the AIC-6110 to define its operating environment. Details of the programming are shown in the flow chart in Figure 3.

7.2.2.1 Selection of Operating modes

The first task of the initialization procedure is to select the various operating modes that are desired by programming the Mode Select Register (Reg 56). The bits 7:5 in Reg 56 should be set to zeroes as these are reserved. The user then selects whether the DRAM or SRAM option is being implemented for the buffer memory. The on-board ENDEC should be selected if 2,7 RLL encoding is desired or should be disabled for NRZ data. If the on board ENDEC is enabled, the user can enable the programmable precompensation if desired, and select the desired VFO option and AM option.

7.2.2.2 Programming Precompensation

If precompensation is enabled, the user needs to program the Early and Late precompensation values desired in registers 57 and 58. The base unit of the precompensation delay is determined by the RC time constant connected to the AIC-6110. The bit to be programmed in the registers 57 and 58 is determined by dividing the total delay desired, (in multiples of the RC time constant), by 0.05, to get the bit number.

7.2.2.3 Buffer Controller Initialization

Next, the size of the external buffer memory used should be programmed in Register 54. Depending on the frequency of the buffer clock being used, the clock rate bits must be set in the Control Register 0, (bits 3,4, Reg 62, W). The buffer clock is utilized to insure the minimum times on the SCSI bus are met by the AIC-6110. Writing any value to the Reset Address Counters (Reg 59) will reset the counters associated with the Read, Write and Stop Address Pointers.

7.2.2.4 SCSI Interface Block Initialization

The SCSI ID of the device using the AIC-6110 must be set in the Control Register 0, (bits 0-2, Reg 62, W). The SCSI transfer rate must be programmed in the Offset Counter Register, (bits 4-6, Reg 68, W). If the asynchronous SCSI protocol is being used, then the offset bits must be programmed to zero. If the synchronous SCSI protocol is being used, the the synchronous offset used must be programmed, (bits 0-3, Reg 68, W).

7.2.2.5 Disk Controller Block Initialization

Select the size of the ECC polynomial being used [32/48-bit ECC polynomial] using bit 7 in register 71. Next, program the actual polynomial in registers 72 through 77. Refer to Section 7.4.5 for details on the ECC related issues. Next the sequencer RAM needs to be setup according to the disk interface requirements. Specific details on this are provided in Section 7.4, with sample sequencer maps.

This completes the initialization process, and the AIC-6110 can be programmed to initiate a desired data operation, as outlined in the following sections.

7.3 SCSI INTERFACE OPERATION

The SCSI interface block in the AIC-6110 supports a single-ended SCSI bus. This single chip synchronous SCSI controller has the required 48 mA drivers and hysterisis receivers on the chip to implement a complete single-ended SCSI bus. The AIC-6110 supports the asynchronous SCSI protocol at a maximum transfer rate of 3 Mbytes/sec over a full length SCSI cable (the AIC-6110 can achieve higher asynchronous transfer rates over a shorter cable), and the synchronous SCSI protocol at a maximum transfer rate of 5 Mbytes/sec.

The AIC-6110 connections for a single-ended SCSI bus are shown in Figure 4. Although the AIC-6110 is ideally suited for embedded SCSI drive applications, where it would be used as a SCSI Target, the AIC-6110 can support the SCSI Initiator mode also, which can be used to implement the COPY command for Target drives or tape systems.

The following sections discuss in detail the SCSI related tasks that are carried out automatically by the AIC-6110 such as arbitration, selection, and reselection, the data transfer mechanism and parity generation and checking on the SCSI bus.

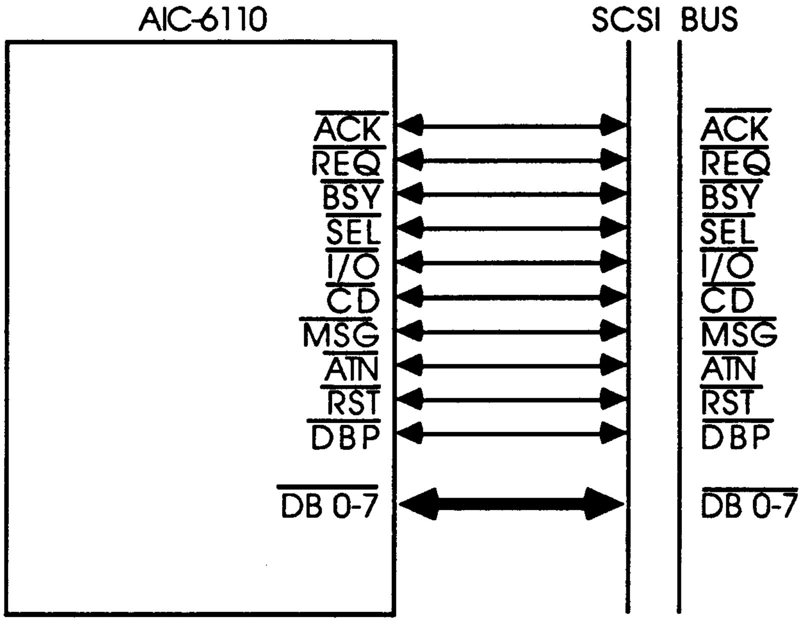


FIGURE 4. SINGLE-ENDED SCSI BUS INTERFACE FOR AIC-6110

7.3.1 Arbitration and Selection or Reselection

The AIC-6110 has the logic necessary to accomplish automatic Arbitration and Selection or Reselection. At the same time, the AIC-6110 is also capable of automatic response to a Selection or Reselection process on the SCSI bus.

The source and destination IDs of the SCSI devices must be loaded in the SCSI Data Register (Reg 65). The En Arbitration/Selection Start bit is set to start the Arbitration process, (bit 0, Reg 63). The AIC-6110 then checks to ensure that the SCSI Bus Free condition exists for 400 ns or else waits for the Bus Free condition and then asserts the SCSI *BSY signal after 800 ns while enabling its SCSI ID onto the SCSI bus. It examines the bus after 1400 ns to take one of two possible actions :

1. If during this time, another device has asserted the SCSI *SEL signal or the AIC-6110 determines that there is a higher ID present on the SCSI Bus, it deasserts its lower priority ID and the SCSI *BSY line. This signals the loss of Arbitration and the AIC-6110 will then wait for the next Bus Free phase to begin the Arbitration process again.

2. If the AIC-6110 has the highest priority ID during the Arbitration phase, it implies that the Arbitration has been won by the AIC-6110 and thus it can proceed further.

Depending on whether the AIC-6110 is a SCSI Initiator or a SCSI Target, (as determined by the state of the SCSI *I/O bit in the SCSI SIGNAL Register), the next process would either be a SELECTION or RESELECTION. The Selection process is initiated to start a connection between two SCSI devices while a Reselection is done by a Target device which disconnected from the Initiator and desires to resume the unfinished transfer. There are four possible situations the AIC-6110 could be in and these are described in detail below.

- Initiator (AIC-6110) selects a Target (AIC-6110 SELECTION)
- Target (AIC-6110) selected by Initiator (AIC-6110 SELECTED)
- Target (AIC-6110) reselects Initiator (AIC-6110 RESELECTION)
- Initiator (AIC-6110) reselected by Target (AIC-6110 RSELECTED)

NOTE: In the initiator mode, if a message out phase for the identify message is required after the selection is complete, then the EN AUTO ATN bit (Bit 1, Reg 63) must be set before selection. The attention signal can be deasserted by resetting the EN AUTO ATN bit (Bit 1, Reg 63).

AIC-6110 SELECTION - Initiator (AIC-6110) selects a Target

Once Arbitration has been won by the Initiator device, it initiates the selection process to select a Target device on the SCSI bus.

- The initiator (AIC-6110) asserts the SCSI *SEL signal and enables the source and destination ID's onto the SCSI Bus.

- It then removes the SCSI *BSY signal 200 ns later.

- Simultaneously the SCSI *ATN signal will be activated on the SCSI bus if the Initiator desires a message out phase (Identify message) to the Target following the completion of the selection process.

- The Target (destination ID on the SCSI bus) responds with the assertion of the SCSI *BSY signal.

- The Initiator (AIC-6110) then deasserts the SCSI *SEL signal 700 ns later.

- This establishes the link between the Initiator and the Target on the SCSI bus.

- The completion of the Selection process results in the generation of a Command Done interrupt reflected in the Status Register 1, (bit 1, Reg 61, R).

AIC-6110 SELECTED -Target (AIC-6110) Selected by Initiator

The Target device is selected by the initiator in a selection process. If the AIC-6110 is the Target then it must be anticipating a selection.

- This is done by enabling the Selected Interrupt mask bit (bit 0, Reg 60, W).

- The AIC-6110 checks for the following conditions to be true for at least 600 ns, and if so, it asserts the SCSI *BSY signal, signalling the acceptance of the Selection.

- SCSI *SEL signal is asserted with the ID that matches the Target's.

- No more than two ID bits are asserted on the SCSI bus.

- There is no SCSI parity error.

- The SCSI *I/O and *RST signals are not active.

- After the Initiator deasserts the SCSI *SEL signal, confirming the Selection process, the Target (AIC-6110) keeps the SCSI *BSY signal asserted and generates a Selected interrupt reflected in the Status Register 0, (bit 0, Reg 60, R).

AIC-6110 RESELECTION - Target (AIC-6110) Reselects Initiator

When an Initiator establishes a link with a Target on the SCSI bus, the Target sometimes disconnects before completion of the task, so as not to lock the SCSI bus, and releases the bus so that other devices may communicate, while it gets ready to finish the remainder of its task. When it is ready to continue with the task, the target reselects the concerned Initiator.

- Before initiating the Arbitration and Reselection process, the SCSI *I/O signal is asserted (which signals a Reselection process according to the SCSI protocol), through the SCSI Signal Register, (bit 6, Reg 64, W).

- The remainder of the process in terms of the SCSI activity is exactly similar to a Selection process as discussed above, except that the Target initiates the Selection process instead of the Initiator.

- The completion of the Reselection process is indicated by the Command Done interrupt in Status Register 1, (bit 1, Reg 61, R).

AIC-6110 RESELECTED - Initiator (AIC-6110) Reselected by Target

After the Target disconnects from the Initiator, it later does a Reselection process to restablish a link. The Initiator (AIC-6110) must be prepared to accept the Reselection process.

- The AIC-6110 is capable of automatic response to Reselection, when enabled by the Reselected Interrupt mask, (bit 1, Reg 60, W).

- When the Initiator device is Reselected it generates a Reselected interrupt reflected in the Status Register 0, (bit 1, Reg 60, R).

- The AIC-6110 checks for the following conditions to be true for a successful operation while responding to a Reselection on the SCSI bus :

- The SCSI *SEL signal is asserted with the ID matching that of the Initiator AIC-6110.

- No more than two ID bits are active on the SCSI bus.

- No SCSI parity error condition exists.

- SCSI *I/O signal is active and SCSI *RST signal is inactive.

Once the communication link has been established or restablished by the AIC-6110 with due support by the local processor, the next process is usually a data transfer on the SCSI bus.

7.3.2 SCSI Data Transfer

The AIC-6110 supports both Asynchronous SCSI data transfer at a maximum transfer rate of 3 Mbytes/sec. and Synchronous SCSI transfer at a maximum transfer rate of 5 Mbytes/sec., over a full SCSI cable length.

The table below outlines the various combinations of the disk and SCSI transfer rates possible and the speed requirements of the buffer memory.

NRZ freq (MHz) [1/T NRZ]	BUFFCLK freq (MHz) [1/T BCLK]	DREQ period (ns) [8T NRZ]	memory cycle time period(ns) [4T BCLK]	# of mem cycles in DREQ period	# of host xfers in DREQ period	SCSI rate (MB/s)
20.0	30	400	133	3.0	2.0	5.00
15.0	30	533	133	4.0	3.0	5.00
15.0	25	533	160	3.3	2.3	4.16
10.0	25	800	160	5.0	4.0	4.16
10.0	20	800	200	4.0	3.0	3.75
10.0	15	800	266	3.0	2.0	2.50
7.5	20	1066	200	5.3	4.3	4.03
7.5	15	1066	266	4.0	3.0	2.81
7.5	10	1066	400	2.6	1.6	1.55
5.0	20	1600	200	8.0	7.0	4.37
5.0	15	1600	266	6.0	5.0	3.12
5.0	10	1600	400	4.0	3.0	1.87

NOTE: To insure that the disk data is not lost due to a delayed memory access from the buffer controller, the NRZ frequency must not be greater than the buffer clock frequency.
T = one clock period
BCLK = Buffer Clock

The AIC-6110 handles SCSI Data transfer to and from the local buffer in three different modes, supporting both the Initiator and Target roles. They are DMA transfer mode, Automatic SCSI PIO and Microprocessor controlled or Manual SCSI PIO.

7.3.2.1 DMA Transfer Mode

In this method, the AIC-6110 can transfer data on the SCSI bus using either asynchronous or synchronous data transfer mechanisms.

The internal FIFO serves as a speed matching buffer for asynchronous transfers. The FIFO acts as an offset buffer for synchronous transfers.

The direction of transfer is set by the ROP/WOP bit in the Control Register 1, (bit 4, Reg 63). If the ROP/WOP bit is set to 0, a read from SCSI and write to disk through the buffer, direction is selected. The read from SCSI and write to buffer and the read from buffer and write to disk operation can be done independently, however the direction of data flow is fixed by the WOP/ROP bit. This implies that SCSI and disk cannot read the buffer simultaneously.

If the ROP/WOP bit is set to Ø, the data from the SCSI Bus is written to the buffer using the Write Address Pointer, while data is read from the buffer and written to the disk using the Read Address Pointer. If the WOP/ROP bit is set to 1, the data is written to the buffer from the disk using the Write Address Pointer while the data is read from the buffer using the Read Address Pointer and written to the SCSI Bus.

When a disconnect operation occurs in this mode - during a write to SCSI, the data in the FIFO is not transferred out. However, with the knowledge of the FIFO status from Reg 69 and the Read Address Pointer, the actual data transferred on the SCSI Bus can be determined and the Read Address Pointer can be setback accordingly to resume transfers at the correct point.

During a read from SCSI, the data in the FIFO after a disconnect operation has occurred, will still be transferred to buffer memory until the FIFO is empty. This method of transfer is most efficient and is completely automatic with respect to the handshake with the SCSI Bus and the buffer memory. The basic setup for initiating a SCSI data transfer using the DMA mode is shown in Figure 5. There are eight possible states the AIC-6110 can be in :

Asynchronous SCSI Protocol

1. AIC-6110 is a Target and it Reads from the SCSI bus.
2. AIC-6110 is a T arget and Writes to the SCSI bus.
3. AIC-6110 is an Initiator and Writes to the SCSI bus.
4. AIC-6110 is an Initiator and Reads from the SCSI bus.

Synchronous SCSI Operation

5. AIC-6110 is a Target and it Reads from the SCSI bus.
6. AIC-6110 is a Target and Writes to the SCSI bus.
7. AIC-6110 is an Initiator and Writes to the SCSI bus.
8. AIC-6110 is an Initiator and Reads from the SCSI bus.

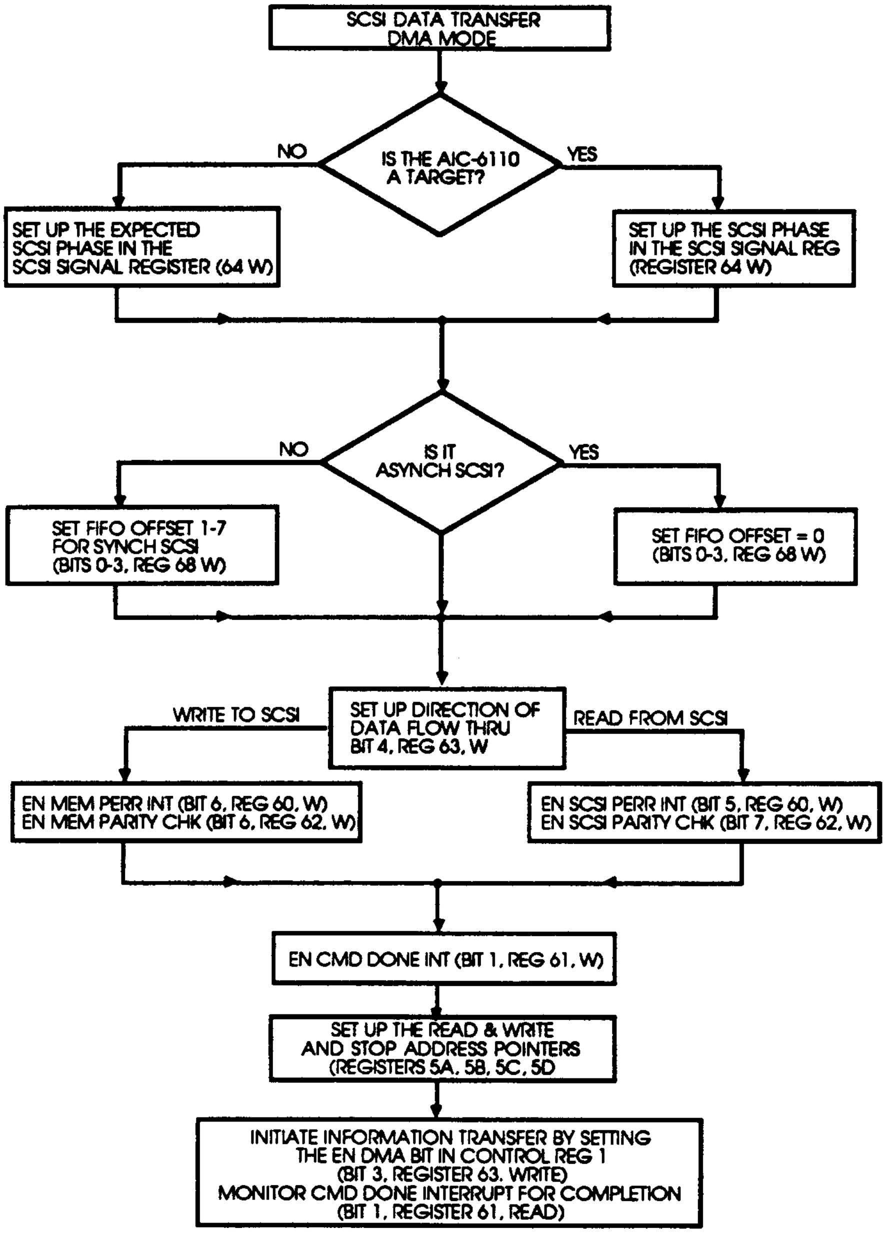

SCSI DATA TRANSFER DMA MODE
IS THE AIC-6110 A TARGET?
NO
YES
SET UP THE EXPECTED SCSI PHASE IN THE SCSI SIGNAL REGISTER (64 W)
SET UP THE SCSI PHASE IN THE SCSI SIGNAL REG (REGISTER 64 W)
IS IT ASYNCH SCSI?
NO
YES
SET FIFO OFFSET 1-7 FOR SYNCH SCSI (BITS 0-3, REG 68 W)
SET FIFO OFFSET = 0 (BITS 0-3, REG 68 W)
SET UP DIRECTION OF DATA FLOW THRU BIT 4, REG 63, W
WRITE TO SCSI
READ FROM SCSI
EN MEM PERR INT (BIT 6, REG 60, W)
EN MEM PARITY CHK (BIT 6, REG 62, W)
EN SCSI PERR INT (BIT 5, REG 60, W)
EN SCSI PARITY CHK (BIT 7, REG 62, W)
EN CMD DONE INT (BIT 1, REG 61, W)
SET UP THE READ & WRITE AND STOP ADDRESS POINTERS (REGISTERS 5A, 5B, 5C, 5D)
INITIATE INFORMATION TRANSFER BY SETTING THE EN DMA BIT IN CONTROL REG 1 (BIT 3, REGISTER 63, WRITE)
MONITOR CMD DONE INTERRUPT FOR COMPLETION (BIT 1, REGISTER 61, READ)

FIGURE 5. SETUP REQUIRED FOR TRANSFER OF DATA USING THE DMA MODE

Asynchronous SCSI Protocol

The Asynchronous SCSI protocol entails a single handshake of SCSI *REQ/*ACK between the Intiator and Target devices on the SCSI Bus. A single byte of data is transferred as a result of this handshake. For details on the protocol, refer to the ANSI Standard document. The AIC-6110 can operate as a target or an initiator, observing the SCSI protocol. The four possible situations for asynchronous SCSI read and write operations as a target or an initiator are described in detail below:

AIC-6110 is a Target and it Reads from the SCSI Bus

The target asserts the SCSI *REQ signal and waits for the data byte from the Initiator along with the SCSI *ACK signal. It then accepts the byte and deasserts the SCSI *REQ signal. Seeing the SCSI *REQ signal go away, the Initiator deasserts the SCSI *ACK signal. The target records that and gets ready to assert the SCSI *REQ signal for the next transaction. The AIC-6110 activates the appropriate signals to the SCSI timing specifications as shown in section 8.9.

In case the buffer controller is slow in removing data from the internal FIFO to the buffer memory and gets filled up, then the AIC-6110 will hold assertion of the next SCSI *REQ signal until space is available in the FIFO for a byte.

If the SCSI parity check interrupt mask is enabled, (bit 5, Reg 60, W) and Parity checking is enabled in the Control Register 0 (bit 7, Reg 62) and a SCSI parity error occurs, then the AIC-6110 generates a SCSI parity error interrupt reflected in the Status Register 0, (bit 5, Reg 60, R). The information transfer will not be stopped.

The data transfer continues until the Write Address Pointer (Reg 5C & 5D) matches the Stop Address Pointer, (Reg 5E & 5F), at which point the Command Done Interrupt is generated.

AIC-6110 is a Target and it Writes to the SCSI Bus

In this case, the AIC-6110 will prefetch the data from the buffer memory into the internal FIFO. The AIC-6110 then asserts the SCSI *REQ signal and puts out the data byte on the SCSI bus. The Initiator takes the data and asserts the SCSI *ACK. The timing details of the handshake are shown in Section 8.9.

In case the buffer controller is slower than the SCSI data transfer and the FIFO becomes empty, then the AIC-6110 holds assertion of the SCSI *REQ until a memory transfer has occurred into the FIFO. It then asserts the SCSI *REQ and continues with the transfer.

If the memory parity checking is enabled and a parity error occurs, then an interrupt is generated and the Status Register 0 reflects a memory parity error, (bit 6, Reg 60, R). The information transfer is suspended until the memory parity checking is disabled in the Control Register 0, (bit 6, Reg 62, W) and then the transfer continues.

Once the handshake for the last byte is completed, the AIC-6110 indicates completion of the data transfer by generating the Command Done interrupt.

AIC-6110 is an Initiator and it Writes to the SCSI Bus

In this case, the Initiator AIC-6110 will prefetch the data from the buffer memory to the FIFO using the Read Address Pointer. Then the AIC-6110 compares the SCSI Bus phase to the expected phase set in the SCSI Signal Register, once the Target asserts the SCSI *REQ signal.

The handshake occurs accordance to the SCSI timing specifications shown in section 8.9.

In case the FIFO falls empty, then the SCSI *ACK assertion is delayed by the Initiator until such time that a memory transfer puts a byte of data in the FIFO.

If the memory parity checking is enabled with the interrupt mask, (bit 6, Reg 62, W) and (bit 6, Reg 60, W), and a memory parity error occurs, an interrupt is generated and the status is reflected in the Status Reg 0 (bit 6, Reg 60, R). The AIC-6110 stops the transfer, until the memory parity checking is disabled and then the transfer continues.

After transfer of the last byte of data on the SCSI bus, and after the Target has deasserted the SCSI *REQ signal, the Initiator deasserts the SCSI *ACK signal and generates the Command Done interrupt signifying the completion of the data transfer.

AIC-6110 is an Initiator and it Reads from the SCSI Bus

Once the basic setup is done as outlined in the SCSI data transfer flowchart, and the DMA has initiated the read from SCSI operation, the AIC-6110 checks to see if the actual SCSI Bus phase matches the expected phase setup in the Initiator (SCSI Signal Register).

A phase mismatch can be determined by enabling the phase error interrupt (bit 3, Reg 60, W) and monitoring the Wrong phase bit in Status Register 0, (bit 3, Reg 60, R). If a phase mismatch occurs, then the correct phase must be loaded in the SCSI Signal Register (Reg 64, W), as read from the SCSI Bus, through the SCSI Signal Register (Reg 64, R).

The handshake is completed according to the timing specifications as outlined in Section 8.9.

If the FIFO becomes full at any point, then the Initiator simply holds the assertion of the SCSI *ACK until space becomes available in the FIFO.

If SCSI parity checking is enabled, (bit 5, Reg 60, W) along with the interrupt mask (bit 5, Reg 60, W) and a SCSI parity error occurs, a SCSI parity error interrupt is generated with the status reflected in the Status Register 0, (bit 5, Reg 60, R).

The data transfer continues until the Read Address Pointer matches the Stop Address Pointer. On receiving the last byte on the SCSI bus, the Initiator holds assertion of the SCSI *ACK until all the data in the FIFO has been moved to the buffer memory.

The completion of the data transfer is indicated by the Command Done interrupt reflected in Status Register 1.

Synchronous SCSI Protocol

In the Synchronous SCSI protocol, a Target device can assert a series of SCSI *REQs in a bufferred fashion, along with a byte of data for each *REQ. The Initiator device at the other end must be capable of accepting this data in a temporary storage until such time that it accepts it. When it does, it asserts a *ACK for each byte. Before initiating any transfers, both these devices establish the acceptable transfer rate and offset limit through the message protocol. For more details on the Synchronous SCSI protocol refer to the ANSI Standard. The AIC-6110 can operate as a target or an initiator and the chip observes the synchronous SCSI protocol. The four possible situations for synchronous SCSI read and write operations as a target or an initiator are described in detail below:

AIC-6110 is a Target and it Reads from the SCSI Bus

In this case, the target will put out a sequence of SCSI *REQ signals up to and equal to the maximum synchronous offset and in accordance to the transfer rate set. As the Target receives SCSI *ACK's from the Initiator, it can send out more SCSI *REQ's ensuring that at no time the difference between the total number of SCSI REQ's and SCSI *ACK's exceed the synchronous offset. The AIC-6110 will keep accepting data as long as the FIFO doesn't get full, or the offset count does not reach the maximum allowed, and if there is no SCSI parity error detected by the AIC-6110 (provided SCSI parity generation and checking is enabled).

If the FIFO is full or the offset count reaches the maximum, the SCSI *REQ signal is held off until some data gets transferred out of the FIFO or some SCSI *ACK pulses are received, after which a SCSI *REQ is asserted to request the next byte.

In case of a SCSI Parity error, reflected in the Status Register 0 (bit 5, Reg 60, R), the AIC-6110 asserts the SCSI parity error interrupt and continues the transfer without stopping.

Once the data transfer is completed, i.e,. the last SCSI *ACK has been deasserted, the FIFO is empty, and the Write Address Pointer equals the Stop Address Pointer, the AIC-6110 generates the Command Done interrupt reflected in the Status Register 1, (bit 1, Reg 61, R), indicating completion of the read from SCSI operation.

AIC-6110 is a Target and it Writes to the SCSI Bus

In this case, the AIC-6110 sends out a sequence of SCSI *REQ's in accordance with the transfer rate set, up to the maximum synchronous offset set. The Initiator responds with SCSI *ACK's in response to acceptance of each data byte transferred to its buffer memory.

The Target holds assertion of further SCSI *REQ's if the difference between the number of SCSI *REQ's and SCSI *ACK's is equal to the synchronous offset, or if the FIFO is empty. When either of these conditions is changed, then the AIC-6110 resumes the normal handshake.

If memory parity checking has been enabled and a memory parity error occurs, an interrupt is generated and the memory parity error status is reflected in the Status Register 0. The transfer is temporary halted, until the memory parity checking is disabled and then it can be resumed.

Once the handshake for the last byte is completed, i.e,. the Read Address Pointer matches the Stop Address Pointer, the AIC-6110 generates the Command Done interrupt indicating the completion of the data transfer operation.

AIC-6110 is an Initiator and it Writes to the SCSI Bus

When the AIC-6110 is an Initiator doing synchronous SCSI operation, it waits for the first SCSI *REQ from the Target. It then checks for a phase match between the expected phase in the SCSI Signal Register and the actual SCSI bus phase. If a phase match occurs, then it continues to put out the prefetched data (from buffer memory, in the FIFO), to the SCSI Bus.

Since the SCSI *REQ's from the Target could be buffered up to the maximum synchronous offset, it makes sure that it asserts the SCSI *ACK signal while putting out the data byte on the SCSI bus as long as there are SCSI *REQ's pending to be replied back to.

If the FIFO falls empty, the AIC-6110 will hold of assertion of the SCSI *ACK until such time that a memory transfer occurs and there is data available in the FIFO.

If memory parity checking is enabled along with the interrupt mask , then an interrupt is generated on the occurrence of a memory parity error, and the data transfer is halted until the microprocessor can intervene and disable the memory parity checking.

The transfer is completed when the Read Address Pointer matches the Stop Address Pointer. The AIC-6110 deasserts the SCSI *ACK signal after the transfer of the last byte and asserts the Command Done interrupt.

AIC-6110 is an Initiator and it Reads from the SCSI Bus

While attempting synchronous SCSI data transfer with the AIC-6110 as an Initiator device, the AIC-6110 waits for the Target to assert the SCSI *REQ. It then checks for a match of the SCSI Bus phase and the expected phase setup in the Initiator. In this mode, it can buffer SCSI *REQ's up to a maximum of the synchronous offset programmed in the Offset Control Register (Reg 68).

Once a phase match occurs, then it starts transfer of the data from the FIFO to the buffer memory. With every byte of data transferred it generates a SCSI *ACK. The rate of SCSI *REQ's and *ACK's is governed by the programmed synchronous transfer rate in the AIC-6110, the details of timing relationships are shown in Section 8.9.

If the FIFO is full or the offset count has reached the maximum allowed, then the data is transferred out to the buffer memory from the FIFO before the AIC-6110 asserts the next SCSI *ACK.

If SCSI parity checking is enabled along with the interrupt, then on occurrence of a SCSI parity, error, an interrupt will be generated. This is reflected in the Status Register 0, but the data transfer continues until the all data is transferred as per the Read Address Pointer.

The last SCSI *ACK is not deasserted by the Initiator until data has been completely transferred out of the FIFO to the buffer memory, after which the Initiator deasserts the SCSI *ACK and the Command Done interrupt is generated signifying completion of the operation.

7.3.2.2 Automatic SCSI PIO transfer mode

In this method of data transfer, the AIC-6110 automatically handles the SCSI bus handshake. The AIC-6110 performs a SCSI *REQ / SCSI *ACK handshake each time the SCSI PIO transfer is invoked through the Control Register 1, (bit 2, Reg 63, W). A single byte of data is transferred between the SCSI bus and a register on the AIC-6110. The buffer controller must use the memory PIO technique to transfer data between the buffer memory and this register. The memory PIO is decribed in Section 7.2.3.

The direction of SCSI transfer is determined by the SCSI *I/O line set in the SCSI Signal register, however the *WOP/ROP bit in the Control Register 1 must be set to for the proper direction of the disk transfers.

Only asynchronous SCSI operations are possible using this data transfer technique. Figure 5 shows a flow chart of the AIC-6110 setup to accomplish the SCSI PIO and Memory PIO operation. The AIC-6110 could be configured as an Initiator or a Target and could perform the SCSI read and write operations.

AIC-6110 is a Target and it Reads from the SCSI Bus

The microprocessor needs to setup the SCSI Bus phase in the SCSI Signal Register (Reg 64) and set the SCSI *I/O signal for a read from SCSI direction.

The AIC-6110 then sets the Enable SCSI PIO bit in the Control Register 1 to start the SCSI PIO operation, (bit 2, Reg 63, W) The AIC-6110 asserts the SCSI *REQ signal and the Initiator responds with the assertion of the SCSI *ACK signal along with the byte of data on the SCSI bus. The data is latched in the SCSI Latched Data Register (Reg 67) by the AIC-6110.

Once the *REQ/*ACK handshake is completed, the AIC-6110 generates a Command Done interrupt, reflected in the Status Register 1, (bit 1, Reg 61). The data byte in the SCSI Latched Data Register must be moved to the Buffer Data Register (Reg 70), to be transferred to buffer memory using Memory PIO, discussed in Section 7.2.3.

AIC-6110 is a Target and it Writes to the SCSI Bus

In this case the microprocessor needs to prefetch the data byte from the buffer memory using Memory PIO transfer (discussed in Section 7.2.3).

This data is available in the Buffer Data Register (Reg 70) and must be transferred to the SCSI Data Register (Reg 65).

The SCSI Bus phase is then setup in the SCSI Signal Register and the SCSI PIO operation is started, (bit 2, Reg 63, W). The AIC-6110 asserts the SCSI *REQ signal and puts out the data on the SCSI Bus.

When the *REQ/*ACK handshake is completed, the AIC-6110 generates the Command Done interrupt signalling the completion of the SCSI PIO operation.

AIC-6110 is an Initiator and it Writes To the SCSI Bus

In this case, the microprocessor needs to prefetch the data byte from buffer memory using Memory PIO, discussed in Section 7.2.3.

This data is in the Buffer Data Register (Reg 70) and needs to be transferred to the SCSI Data Register (Reg 65) and the SCSI PIO transfers are enabled through the Control Register 1, (bit 2, Reg 63, W).

Once the Target asserts the SCSI *REQ signal and the expected phase matches the actual phase on the SCSI bus, the Initiator asserts the SCSI *ACK signal and puts out the data on the SCSI Bus.

When the *REQ/*ACK handshake is completed, the AIC-6110 generates a Command Done interrupt signalling the completion of the SCSI PIO operation.

AIC-6110 is an Initiator and it Reads from the SCSI Bus

In this case, the microprocessor needs to setup the expected phase in the SCSI Signal Register (Reg 64) and enable the SCSI PIO transfers, (bit 2, Reg 63, W).

If the Target asserts the SCSI *REQ signal and phase match occurs, the Initiator reads the data byte into the SCSI Latched Data Register (Reg 67) and asserts the SCSI *ACK signal.

Once the *REQ/*ACK handshake is completed, the Command Done interrupt is generated, to indicate completion of the SCSI PIO operation. The data in the SCSI Latched Data Register (Reg 67), is transferred to the Buffer Data Register (Reg 70), to be moved eventually to the buffer memory using Memory PIO, discussed in Section 7.2.3.

7.3.2.3 Microprocessor Controlled SCSI PIO transfer mode

Microprocessor Controlled PIO is the mode of data transfer which requires the user to manipulate all the SCSI Bus signals. The individual SCSI Bus signals are controlled by the microprocessor through the SCSI Signal Register (Reg 64), to accomplish the SCSI handshake.

The microprocessor has complete control of the SCSI Bus signals. Information transfer may not be setup prior to the first SCSI *REQ.

Since the microprocessor has complete control of all the events which must occur to complete the information transfer and a Command Done interrupt will not be generated on completion of the operation.

The various modes of operation of the AIC-6110 are exactly similar to that described in Section 7.2.2.2. The individual SCSI handshake generating must be performed manually through the SCSI Signal Register.

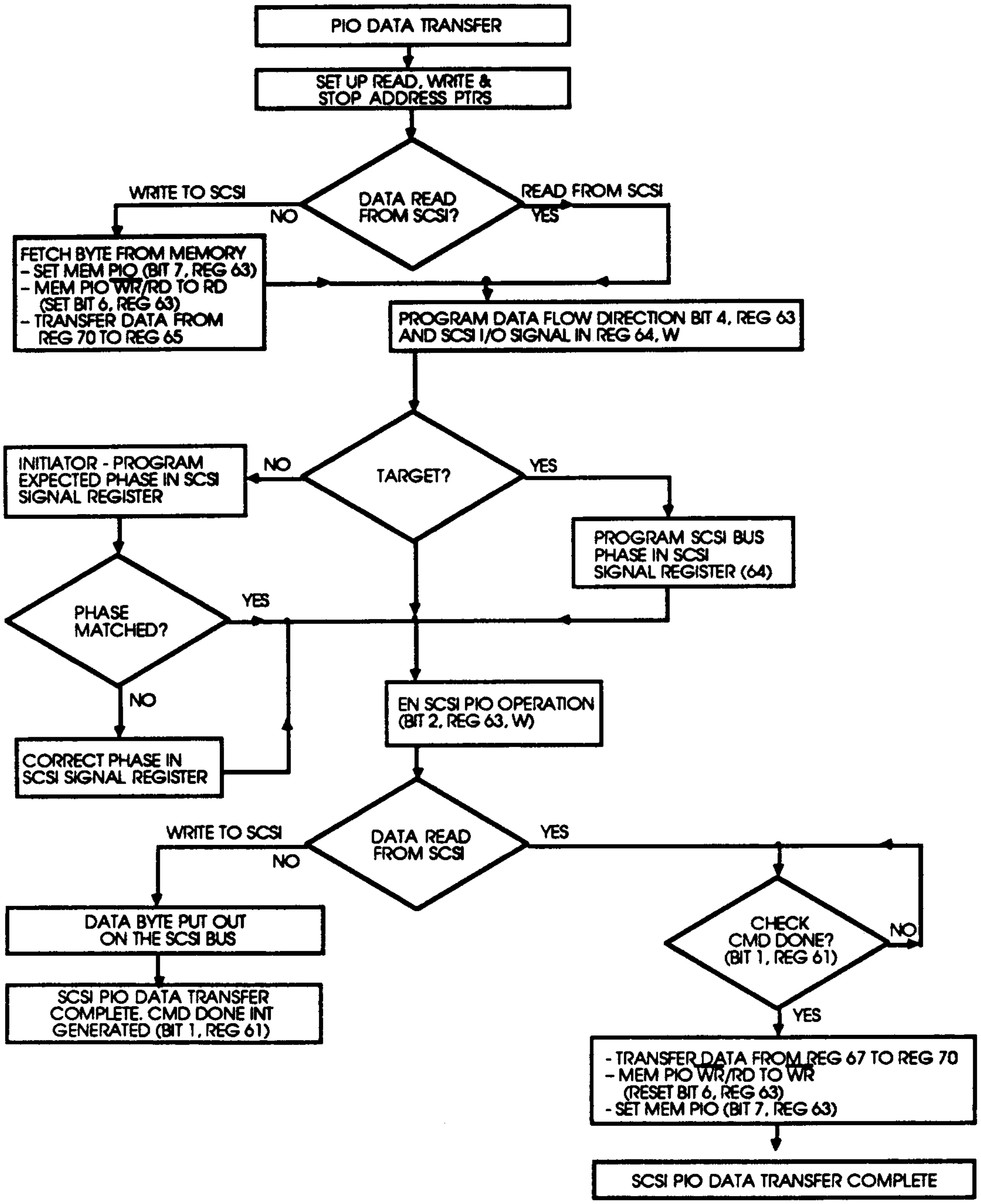

PIO DATA TRANSFER
SET UP READ, WRITE & STOP ADDRESS PTRS
DATA READ FROM SCSI?
WRITE TO SCSI
NO
READ FROM SCSI
YES
FETCH BYTE FROM MEMORY
- SET MEM PIO (BIT 7, REG 63)
- MEM PIO WR/RD TO RD (SET BIT 6, REG 63)
- TRANSFER DATA FROM REG 70 TO REG 65
PROGRAM DATA FLOW DIRECTION BIT 4, REG 63 AND SCSI I/O SIGNAL IN REG 64, W
TARGET?
NO
YES
INITIATOR - PROGRAM EXPECTED PHASE IN SCSI SIGNAL REGISTER
PROGRAM SCSI BUS PHASE IN SCSI SIGNAL REGISTER (64)
PHASE MATCHED?
YES
NO
CORRECT PHASE IN SCSI SIGNAL REGISTER
EN SCSI PIO OPERATION (BIT 2, REG 63, W)
DATA READ FROM SCSI
WRITE TO SCSI
NO
YES
DATA BYTE PUT OUT ON THE SCSI BUS
SCSI PIO DATA TRANSFER COMPLETE. CMD DONE INT GENERATED (BIT 1, REG 61)
CHECK CMD DONE? (BIT 1, REG 61)
NO
YES
- TRANSFER DATA FROM REG 67 TO REG 70
- MEM PIO WR/RD TO WR (RESET BIT 6, REG 63)
- SET MEM PIO (BIT 7, REG 63)
SCSI PIO DATA TRANSFER COMPLETE

FIGURE 6. DATA TRANSFER USING SCSI PIO AND MEMORY PIO

7.3.3 Memory PIO Data transfer mode

The memory PIO mode is used for transfer of data between the Buffer Data Register (Reg 70) and the buffer memory.

The Address Pointers are setup by the microprocessor initially. The MEM PIO *WR/RD bit in the Control Register 1 establishes the direction of Memory PIO transfers. If set to 0, the contents of the Write Address Pointer is used to write data in the Buffer Data Register to the buffer memory. If set to 1, the contents of the Read Address Pointer is used to read the data from buffer memory and latch it in the Buffer Data Register.

The Read and Write Address Pointers are automatically incremented for every Memory PIO transfer. The memory PIO operation is started by setting the bit in the Control Register 1, (bit 7, Reg 63, W). An automatic memory transfer cycle occurs between the buffer and the Buffer Data Register. This transfer mode is used by the microprocessor in conjunction with the SCSI PIO mode to transfer data between the buffer memory and the SCSI Bus. Figure 5 shows a flowchart for the setup needed to start a Memory PIO operation.

The memory PIO technique also provides a means for the microprocessor to access the buffer memory, provided the SCSI block is not accessing the buffer memory. Upon completion of a Memory PIO operation, a Command Done interrupt is generated by the AIC-6110 reflected in the Status Register 1, (bit 1, Reg 61, R).

7.4 BUFFER CONTROLLER OPERATION

The Buffer Controller block is designed to support both SRAMs and DRAMs. If the SRAM mode is selected, the RAM control signals MAS, *MOE, and *WE, are asserted to fully control industry standard SRAMs. Upon the receipt of a memory request from disk or host, the low address byte is asserted along with MAS (Memory Address Strobe) to latch the low byte into an external octal latch. The upper address byte is asserted next, along with the pertinent control signals to perform a read or write to RAM.

If the DRAM mode is selected the control signals *RAS, *CAS, and *WE are used. Upon the receipt of a request in this mode, the low address byte and *RAS are asserted first, and the upper address byte and *CAS next. The assertion of *WE depends on whether a read or write operation to memory is taking place. The DRAMs are supported in the "early- write" mode of operation only, where the DRAM's output is always enabled. Figure 12 shows the interface of DRAMs to the AIC-6110.

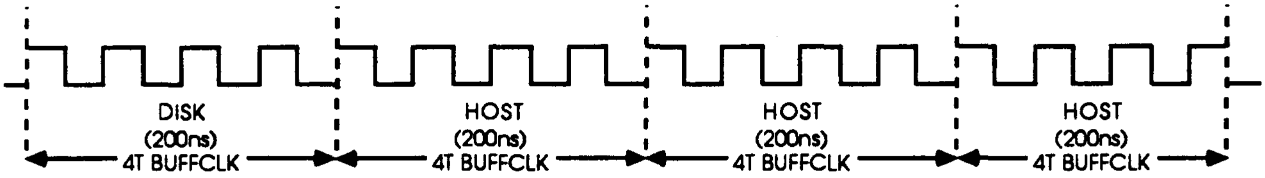
If there is no request from either the host or disk within 16 BUFFCLK cycles, the buffer controller automatically enters a refresh mode of operation. In this mode, a *CAS before *RAS refresh timing occurs, which will refresh DRAMs supporting *CAS before *RAS refresh with an internal refresh counter. Upon the receipt of a memory request from either the host, disk, or the local microprocessor the controller immediately completes the last refresh cycle and starts to service that request.

The local microprocessor can access the buffer memory through an internal register. To do this, the processor must first update the Read Address or Write Address Pointers. If a write to memory is desired, the data must be written to BUFFER DATA REG. The direction of transfer and the request for transfer is given to the buffer controller through an internal register. Upon the completion of data transfer, the buffer controller resets a bit which must be polled by the microprocessor. If a read from memory had taken place, the valid data could be found in BUFFER DATA REG.

The buffer controller performs a complete memory cycle in four BUFFCLK cycles. Using 200 ns DRAMs, the diagram below illustrates how a typical 10 MHz drive and a 20 MHz BUFFCLK oscillator can achieve a 3.75 Mbytes transfer rate on the SCSI interface. Since the NRZ data rate is 10 MHz, every 800 ns a byte wide data must be buffered from the disk as shown in the diagram.

The buffer controller requires four clock cycle for a memory cycle, thus in 800 ns it can at best achieve 4 memory cycles. If one cycle is from the disk request, the other three cycles can be allocated for host requests. This translates into a 3 byte transfer in 800 ns, or an effective transfer rate of 3.75 MHz. The Table on page 55 summarizes some facts on how various choices for disk clock and buffer clock can effect the system performance and cost. Faster RAMs generally cost more, but are not necessarily required to achieve the required performance.

With NRZ frequency = 10 MHz and BUFFCLK frequency = 20 MHz we have:



7.5 DISK DATA PATH FUNCTIONS

The disk data path circuitry in the AIC-6110 is responsible for transfer of data between the buffer memory and the disk media.

It consists of an interface to the buffer controller, a disk sequencer/formatter, serializer/deserializer and a 2,7 RLL encoder/decoder. The disk formatter/sequencer executes the command sequences to perform the following operations.

Read ID	(sector identification)
Read ID and Read Data	(sector read operation)
Read ID and Write Data	(sector write operation)
Write ID and Write Data	(disk format operation)

Besides these tasks, the sequencer also controls the various signals associated with the disk interface (data path).

The sequencer program consists of 96 bytes of RAM organized as a matrix of 24 x 4 bytes. These locations have to be programmed after power-up for proper operation. Under firmware control, the AIC-6110 sequencer can be made to sequence through different types of operations. The timing relationships of the various signals can be monitored and controlled by the sequencer. There are other registers in the sequencer that can be programmed to control the track format, byte sync, format data patterns, ECC polynomials, and padding characters.

Data to be written to the disk is transferred from a 8-bit parallel format (from the buffer memory), serialized and transmitted to the disk channel. ECC is calculated and appended to the bit stream. The AIC-6110 outputs NRZ data or 2,7 RLL data.

The disk data path functional operation can be best understood by examining the interface between the disk formatter/sequencer and the buffer controller function, the programming of the sequencer RAM, the format and the read/write operation.

Next a detailed soft-sectored operation is presented with information on sequencer map set-up, and the operation for read ID, read data, write data, and format operations.

The sequencer map and registers of the AIC-6110 can also handle extended data handling requirements and variable sector sizes, multi-sector transfers, verify sector and search sector data are explained.

The data then must be checked for any errors, the ECC circuit allows for error correction and checking with a fixed CRC polynomial, 32-bit, 48-bit and external ECC mechanisms.

Finally, the AIC-6110 has the ability to have the drive data available as NRZ or 2,7 RLL data encoded (decoded). The operation of the 2,7 RLL data endec is described in detail .

7.5.1 Interface between the Disk Formatter/Sequencer and the Buffer Controller

The interface between the disk controller section and the buffer controller is through an internal asynchronous handshake. Since the disk sequencer/formatter and buffer controller run with independent clocks, the internal request signal is synchronized with the buffer clock (BUFFCLK) and serviced as soon as possible.

A one-byte FIFO between the disk formatter/sequencer and the buffer controller ensures that the data will not be lost if the buffer controller cannot respond to a disk request immediately. The disk side request has the highest priority as compared to the SCSI side or the microprocessor.

The data flow can only be in two directions. In one case, data is read from SCSI and written to the buffer memory using the Write Address Pointer and can be read from the buffer memory using the Read Address Pointer and written to the disk.

In the other case, data is read off the disk and written to the buffer memory using the Write Address Pointer and it can be read from buffer memory using the Read Address Pointer and written to SCSI. The direction of data flow, SCSI to disk or vice versa is controlled by the WOP/ROP bit in the Control Register 1, (bit 4, Reg 63). At no time can the SCSI and the Disk sections access the buffer memory simultaneously in one direction. For more details on the buffer controller refer to Section 7.4.

7.5.2 Programming the Sequencer RAM

The control of data to and from the disk is handled by the contents of the sequencer RAM. The sequencer map has registers which provide maximum flexibility to setup the desired track format and the associated parameters like byte sync, gap lengths etc. The sequencer also controls the timing relationships between the disk interface output signals and monitors the disk interface input lines to branch to different sequencer locations.

The RAM based sequencer consists of 96 bytes organized as 24 words that are four bytes wide. These four bytes are identified as DATA, COUNT, CONTROL and the NEXT ADDRESS fields.

The DATA field contains data which may be used to initialize the track format, including gap, ID field, and format data fill patterns. This data can also be compared to the data coming in from the disk to identify specific fields or to execute sector compares.

The COUNT field specifies the initial value of the sequencer counter for the current word. The count field is a 5-bit counter if the data transfer bit in the CONTROL field is zero. The count field is an 8-bit counter if the data transfer bit in the CONTROL field is 1, as is the case during sector data transfers. The Sequencer counter is decremented once a byte period, (every eight Read/Reference clock cycles). When the count reaches zero, the sequencer will go to the next state.

The NEXT ADDRESS field is used to determine the next state the sequencer will go to. If a branch address has been programmed in the Branch/NA Register (Reg 78), the sequencer will branch to the branch address if the branch condition is satisfied.

The CONTROL field is used to generate and initiate all synchronous data handling operations.

The operation of the Sequencer revolves around the Branch/NA Register (Reg 78) and the Sequencer Start/Status Register (Reg 79, R/W). The sequencer start register is first loaded with the address where the sequencer is to begin execution. Thereafter, the AIC-6110 sequences through the sequencer RAM and the next address is executed. If a successful branch condition occurs, the next address is based on the contents of Register 78. Otherwise, it is based on the contents of the NEXT ADDRESS field at that address. Thus by setting up different branch conditions which are based on external or internal events, the chip can be made to sequence through different operations.

The AIC-6110 also has an eight-byte stack, available through register 7Fh. By enabling the stack during a read process, information read from the disk can be pushed on the stack and later examined by the microprocessor.

There are two types of disk media, soft-sectored drives and hard-sectored drives. Soft-sectored drives have an index location to mark the beginning of the track. They use an Address Mark pattern to delineate sectors. Hard-sectored drives have an Index location to mark the beginning of the track and a sector mark to locate each sector.

The sample programs of the sequencer memory maps for soft sectored MFM drives, soft sectored 2,7 RLL drives, and hard sectored ESDI drives are shown in Figures 6, 7, and 8. The user should select the Sequencer map appropriate to his application and modify it to meet his specific design requirements. Included in Appendix A, is a detailed description of the AIC-6110 sequencer programmed for a soft-sectored ESDI format operation. The example is presented using 'snapshots' of the AIC-6110 Sequencer and system activity descriptions.

There are three basic disk operations - FORMAT, READ and WRITE. These will be discussed in detail for a soft-sectored drive using 2,7 RLL encoding schemes.

7.5.3 Soft-sectored Disk Operation

This section describes in detail the format, read and write operation for a soft sectored 2,7 RLL drive. The Sequencer map used is shown in Figure 7,while Figure 9 gives a flow chart of the format operation. The process for an MFM type of soft-sectored drive would be very similar and would follow the sequencer map steps outlined in Figure 6.

7.5.3.1 Format operation

The format function consists of a special disk write operation that will write the sector IDs, the data fields, the fill and synchronizing characters for the whole track. The format yields 32 sectors per track with a sector size of 256 bytes. However, larger/variable sector sizes, such as 512 bytes can be achieved and the techniques to achieve variable sector sizes are discussed in Section 7.5.4. The sector sizes are flexible and can be accordingly adjusted to the required sizes in the sequencer map.

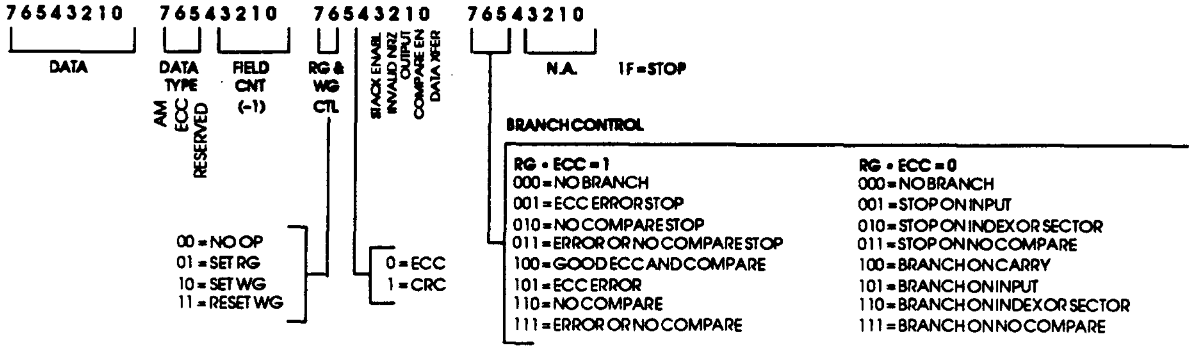
During the AIC-6110 initialization process, the sequencer is programmed with the appropriate map. When a format operation is desired, the microprocessor needs to set up certain registers of the sequencer.

The ID header fields (cylinder, head, sector, and flag bytes) for the first sector are setup in Registers E6 - E9 respectively.

The sector size is initialized in register CF (a value of FFh denotes 256 bytes per sector). Next the following Next Address Registers are loaded with the given addresses to setup the Sequencer for a format operation - Register A0 = 00h Register 80 = C0h, Register 8A = 0Bh, Register 90 = 11h, Register 91 = 01h.

FIGURE 7. SOFT-SECTORED MFM DRIVE (FORMAT, READ AND WRITE)

	E0 THRU F7 DATA		C0 THRU D7 COUNT		A0 THRU B7 CONTROL		80 THRU 97 NEXT ADRS.		ADRS. R79	BRANCH R78	COMMENTS
0	0	0	0	0	W-48 R=48	F=00	W=13 R=13	F-C0	0	1	WAIT FOR INDEX TO START FORMAT
1	4	E	0	A	8	0	0	2	1		WG , GAP 1 AFTER INDEX, GAP 3
2	0	0	0	0	C	0	0	3	2		WG , FOR 2,7 VFO LOCK ON
3	0	0	0	A	8	0	0	4	3		WG , FVO LOCK ON FOR ID
4	A	1	8	0	0	2	0	5	4		ID ADDRESS MARK
5	F	E	0	0	0	2	0	6	5		ID SYNCH BYTE
6	CYLINDER		0	0	1	2	0	7	6		CYLINDER
7	HEAD		0	0	1	2	0	8	7		HEAD
8	SECTOR		0	0	1	2	0	9	8		SECTOR
9	FLAG		0	0	1	0	0	A	9		FLAG
A	0	0	4	3	0	0	R=74 W=60	F=0B	A		ID ECC BR TO STOP ON ERROR
B	0	0	0	5	C	0	0	C	B		PAD. DATA FIELD TO GAP 3 (FMT)
C	0	0	0	B	8	0	0	D	C		WG , VFO LOCK ON FOR DATA
D	A	1	8	0	0	2	0	E	D		DATA ADDRESS MARK
E	F	8	0	0	0	2	0	F	E		DATA SYNCH BYTE
F	6	C	F	F	0	5	1	0	F		DATA FIELD (FMT FILL = GCL)
10	0	0	4	3	0	0	R=71 W=11	F=11	10		DATA ECC BR TO STOP ON ERROR
11	0	0	0	1	0	0	R=16 W=16	F=01 R=12	11		POST DATA FIELD
12	4	E	0	0	0	0	5	2	12		WAIT FOR INDEX & STOP
13	0	0	0	1	4	0	0	4	13		RG FOR ID
14	0	0	0	1	0	0	1	5	14		PAD ID TO DATA FIELD
15	0	0	0	1	4	0	0	D	15		RG FOR DATA
16	0	0	0	3	C	0	1	3	16		START FOR R/W DATA WG
17	X	X	X	X	X	X	X	X	17		DONT CARES



00 = NO OP
01 = SET RG
10 = SET WG
11 = RESET WG

0 = ECC
1 = CRC

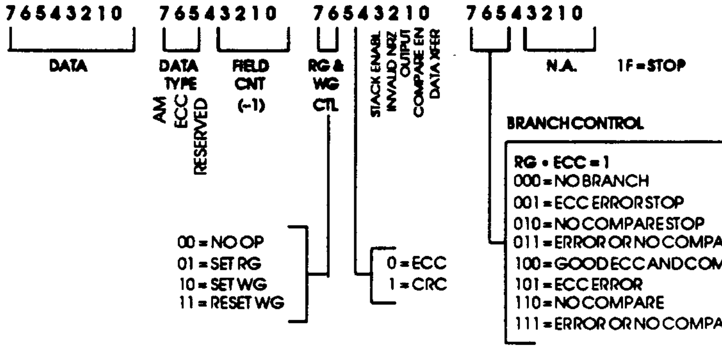
RG • ECC = 1	RG • ECC = 0
000 = NO BRANCH	000 = NO BRANCH
001 = ECC ERROR STOP	001 = STOP ON INPUT
010 = NO COMPARE STOP	010 = STOP ON INDEX OR SECTOR
011 = ERROR OR NO COMPARE STOP	011 = STOP ON NO COMPARE
100 = GOOD ECC AND COMPARE	100 = BRANCH ON CARRY
101 = ECC ERROR	101 = BRANCH ON INPUT
110 = NO COMPARE	110 = BRANCH ON INDEX OR SECTOR
111 = ERROR OR NO COMPARE	111 = BRANCH ON NO COMPARE

NOTE: This software has been shown as an example sequencer map and can only be used with Adaptec chips. The use of this software is not guaranteed by Adaptec.

FIGURE 8. SOFT-SECTORED 2,7 RLL DRIVE (FORMAT, READ AND WRITE)

SEQUENCE MEMORY BIT MAP

	E0 THRU F7 DATA		C0 THRU D7 COUNT		A0 THRU B7 CONTROL		80 THRU 97 NEXT ADRS.		ADRS. R79	BRANCH R78	COMMENTS
0	0	0	0	0	R=48 W=48	F=00	R=13 W=13	F=C0	0	1	WAIT FOR INDEX TO START FMT
1	A	A	0	A	8	0	0	2	1		WG , GAP 1 AFTER INDEX, GAP 3
2	A	A	0	0	C	0	0	3	2		WG FOR 2,7 VFO LOCK ON
3	0	0	0	A	8	0	0	4	3		WG VFO LOCK ON FOR ID PR EA
4	5	E	8	0	0	2	0	5	4		ID ADDRESS MARK
5	A	1	8	0	0	2	0	6	5		ID SYNCH BYTE
6	CYLINDER		0	0	1	2	0	7	6		CYLINDER (ID BYTE 1)
7	HEAD		0	0	1	2	0	8	7		HEAD (ID BYTE 2)
8	SECTOR		0	0	1	2	0	9	8		SECTOR (ID BYTE 3)
9	FLAG		0	0	1	0	0	A	9		FLAG (ID BYTE 4)
A	0	0	4	3	0	0	R=74 W=60	F=0B	A		ID ECC, BR TO STOP ON ERROR
B	0	0	0	5	C	0	0	C	B		ID POSTAMBLE (GAP 3), WG
C	0	0	0	A	8	0	0	D	C		WG , VFO LOCK ON FOR DATA
D	5	E	8	0	0	2	0	E	D		DATA ADDRESS MARK
E	A	0	8	0	0	2	1	7	E		DATA SYNC BYTE
F	6	C	F	F	0	5	1	0	F		DATA FIELD (FORMAT FILL=6C)
10	0	0	4	3	0	0	R=71 W=11	F=11	10		DATA ECC, BR TO STOP ON ERROR
11	0	0	0	1	0	0	R=16 W=16	F=01 R=12	11		DATA POSTAMBLE (GAP 3)
12	A	A	0	0	0	0	5	2	12		WAIT FOR INDEX AND STOP
13	0	0	0	1	4	0	0	5	13		RG FOR ID FIELDS
14	0	0	0	1	0	0	1	5	14		ID POSTAMBLE TO DATA
15	0	0	0	1	4	0	0	E	15		RG FOR DATA FIELD
16	0	0	0	3	C	0	1	3	16		START FOR R/W DATA, WG
17	F	8	0	0	0	2	0	F	17		DATA FIELD IDENTIFIER



BRANCH CONTROL

RG • ECC = 1	RG • ECC = 0
000 = NO BRANCH	000 = NO BRANCH
001 = ECC ERROR STOP	001 = STOP ON INPUT
010 = NO COMPARE STOP	010 = STOP ON INDEX OR SECTOR
011 = ERROR OR NO COMPARE STOP	011 = STOP ON NO COMPARE
100 = GOOD ECC AND COMPARE	100 = BRANCH ON CARRY
101 = ECC ERROR	101 = BRANCH ON INPUT
110 = NO COMPARE	110 = BRANCH ON INDEX OR SECTOR
111 = ERROR OR NO COMPARE	111 = BRANCH ON NO COMPARE

NOTE: This software has been shown as an example sequencer map and can only be used with Adaptec chips. The use of this software is not guaranteed by Adaptec.

FIGURE 9. HARD-SECTORED ESDI DRIVE (FORMAT, READ AND WRITE)

SEQUENCE MEMORY BITMAP

	E0 THRU F7 DATA		C0 THRU D7 COUNT		A0 THRU B7 CONTROL		80 THRU 97 NEXT ADRS.		ADRS. R79	BRANCH R78	COMMENTS
0	0	0	0	0	0	0	C	0	0	1/1F	FORMAT START/STOP ON INDEX
1	0	0	0	B	8	0	0	2	1		WRITE VFO SYNC (ID PREAMBLE)
2	ID SYNC		8	0	1	2	0	3	2		ID SYNC BYTE
3	CYLINDER		0	0	1	2	0	4	3		CYLINDER
4	HEAD		0	0	1	2	0	5	4		HEAD
5	SECTOR		0	0	1	2	0	6	5		SECTOR
6	FLAG		0	0	1	0	0	7	6		FLAG
7	0	0	4	3	0	0	R=72 W=6A	F=08	7		ID ECC
8	0	0	0	2	0	0	0	9	8		ID POSTAMBLE PAD BEFORE WG
9	0	0	0	0	C	0	0	A	9		WG AT END OF ID (SPLICE)
A	0	0	GAP 2		8	0	0	B	A		DATA VFO SYNC (DATA PREAMBLE)
B	DATA SYNC		8	0	0	2	0	C	B		DATA SYNC BYTE
C	DATA FILL		F	F	0	1	0	D	C		DATA FIELD = 512 BYTES
D	DATA FILL		F	F	0	1	0	E	D		(SUPRESS XFER = 1 DURING FMT)
E	0	0	4	3	0	0	R=70 W=15	F=00	E		DATA ECC
F	X	X	X	X	X	X	X	X	F		DON'T CARES
10	0	0			0	0	START=D0 STOP=1F		10	11	R/W START/STOP ON SECTOR
11	0	0	0	1	4	0	0	2	11		RG FOR ID
12	0	0	0	1	0	0	1	3	12		PAD FOR 2 BYTES
13	0	0	0	1	4	0	0	B	13		RG FOR DATA FIELD
14	X	X	X	X	X	X	X	X	14		DON'T CARES
15	0	0	0	2	0	0	1	6	15		DATA ECC PAD BEFORE WG
16	0	0	0	0	2	0	1	0	16		
17	X	X	X	X	X	X	X	X	17		DON'T CARES

76543210	76543210		76543210		76543210		
DATA	DATA TYPE AM ECC RESERVED	FIELD CNT (-1)	RG & WG CTL	STACK ENABL INVALID NRZ OUTPUT COMPARE EN DATA XFER		N.A.	1F = STOP

RG & WG CTL:
00 = NO OP
01 = SET RG
10 = SET WG
11 = RESET WG

0 = ECC
1 = CRC

BRANCH CONTROL

RG • ECC = 1	RG • ECC = 0
000 = NO BRANCH	000 = NO BRANCH
001 = ECC ERROR STOP	001 = STOP ON INPUT
010 = NO COMPARE STOP	010 = STOP ON INDEX OR SECTOR
011 = ERROR OR NO COMPARE STOP	011 = STOP ON NO COMPARE
100 = GOOD ECC AND COMPARE	100 = BRANCH ON CARRY
101 = ECC ERROR	101 = BRANCH ON INPUT
110 = NO COMPARE	110 = BRANCH ON INDEX OR SECTOR
111 = ERROR OR NO COMPARE	111 = BRANCH ON NO COMPARE

NOTE: This software has been shown as an example sequencer map and can only be used with Adaptec chips. The use of this software is not guaranteed by Adaptec.

Bit Maps

Now load the Branch/NA register (Reg 78) with 01h to ensure that the sequencer will branch on the index pulse to address 01h and then write a GAP1.

If during the format operation, a fixed pattern is to be written repetitively in the data field, then program the pattern desired in Register EF = 6Ch for this example. Set the suppress transfer bit in the Operation Control Register (bit 5, Reg 7A). If this bit is not set, then the data field will be written with the contents of the buffer memory.

Check the index past bit in the Operation Control Register to ensure that an index pulse has passed by since the last read of the register (bit 0, Reg 7A). Next, load the Sequencer Start Address Register (Reg 79) with 00h to start the format operation.

ADDRS	DESCRIPTION
00	The AIC-6110 waits for an index pulse to start the format operation . The sequencer program forces the AIC-6110 to loop on location 00h until the index pulse is received.
01	Once the index pulse is received (index past bit is set in Register 7A), Write Gate is asserted and eleven bytes of GAP1 with a pattern of AAh is written on the disk.
02	Next Write Gate is deasserted.
03	Write Gate is reasserted for writing sixteen bytes of 00h for the ID preamble or VFO sync field.
04	Followed by one byte of Address Mark pattern 5Eh [Sequencer address 4]. If the Enable AM Active Interrupt mask is set in the Interrupt Mask Register 1 (bit 7, Reg 61, W), then an interrupt is generated and status is reflected in the Disk Status Register (bit 7, Reg 79). The bit is reset when the ECC bytes are written for the ID filed.
05	The Sequencer then writes on byte of the ID Sync field with a pattern of A1h .
06	Next the ID header fields of Cylinder
07	Head
08	Sector, and
09	Flag are written to the disk.
0A	Then the ID ECC field of four bytes (32 bit polynomial selected) are written to the disk.
0B	The GAP3 pad or ID postamble field of six bytes of 00h is padded after the ID ECC field then Write Gate is deasserted.
0C	The format operation continues with the assertion of Write Gate and the writing of sixteen bytes of the Data VFO sync or preamble with a pattern of 00h is written to the disk.
0D	Followed by one byte of Data Address Mark 5Eh.
0E	Then the Data Sync byte with a pattern of A0h is written.
17	The Data Sync field is followed by one byte of F8h which constitutes the data field identifier.
0F	Next the Data field is filled with the 6Ch pattern repetitively [Sequencer address 0Fh]. If the Data transfer interrupt mask were enabled in the Interrupt Mask Register 1 (bit 6, Reg 61, W), then an interrupt would be generated and the Data Xfer status would be reflected in the Disk Status Register (bit 6, Reg 79, R).

While the Sequencer is busy writing the Data field of the present sector, the microprocessor could setup the ID header fields for the next sector by updating Registers E6 - E9.

10 The Sequencer then follows the Data field with four bytes of ECC for the data field.

11 It then write the data postamble or post data field two bytes of 00h, completing the format of one sector. If the next sector to be written is not the last sector, then the Sequencer repeats the format steps, starting at Sequencer address 1, on the next sector. However if the next sector to be written is the last sector, the microprocessor loads Register 91 = 12h

12 Then it continues to write the GAP 4 pattern of AAh, until the Index pulse is reached. This stops the Sequencer, thus completing the format operation on the whole track. The stopped bit in the Disk Status Register (bit 4, Reg 70) is set when the sequencer stops execution and an interrupt would be generated if the En Stopped Int mask bit is set in the Interrupt Mask Register 1 (bit 4, Reg 61, W).

7.5.3.2 Read/Write Operation

The read operation consists of a read of the sector ID to identify the sector and if it is the desired sector, then a read of the sector's data field is performed. On the other hand, a disk write operation consists of read of the sector ID (to locate the desired sector), followed by a write operation on the sector's data field. The three operations, viz Read ID, Read Data and Write Data are discussed for a soft sectored 2,7 RLL drive using the sequencer map outlined in Figure 7 and a sector size of 256 bytes is assumed.

A requirement of the Read Operation is to synchronize the incoming data on byte boundaries and then either process the data or pass the data through to the buffer memory. The AIC-6110 uses a bit-ring to maintain the synchronization at byte boundaries and to synchronize internal activities. The data synchronization and bit-ring initialization occurs when a specific data stream is detected and the ADDRESS MARK DETECT (*WAM/*AMD) input is active. The process begins when the CONTROL field of the current Sequencer word activates Read Gate and the bit-ring is stopped within two byte times. A match between the Sync Compare Control Register (Reg 7C) and the serial NRZ read data input will cause a sync detect (if *AMD is active), the bit-ring to restart at zero and data will be gated into the ECC logic. It should be noted that in the 2,7 RLL mode, *AMD is an internal signal which is automatically generated by the internal ENDEC.

Only those bits enabled by Sync Compare Control Register (Reg 7F) can be set for comparison. Any Don't Care bits must be set to zero. The comparison is independent of the Compare Enable function of the sequencer RAM, which is normally used for additional qualification such as differentiating between ID Address Mark and Data Address Mark. Typically, the first byte after the synchronization byte is used to differentiate between the ID and Data fields of the sector. After synchronization, the Sequencer has the ability to enable the comparison of the incoming data against the DATA field of the sequencer and can also capture the incoming data on the internal 8-byte stack.

The programmable compare capability of the sequencer can be used to automatically access the correct sector by recognizing the proper sector ID. If the sector ID does not match the DATA field of the current Sequencer word, then the Sequencer can be programmed to stop. The Sequencer can then be restarted after a delay to look for the sector again. The delay minimizes the danger of false synchronization, especially during Write Splices.

Sequencer setup for read/write operation

On power-up initialization, the 2,7 RLL soft sectored version of the sequencer map is programmed in the AIC-6110. Depending on whether a read or write operation is desired, certain registers need to be setup accordingly. These are outlined below:

Seq. addr	Read Op	Write Op	Description
E6	cyl	cyl	ID field
E7	head	head	ID field
E8	sector	sector	ID field
C4	FFh	FFh	sector size
A0	48h	48h	Control field
80	13h	13h	Next Address
8A	74h	6Ch	Next Address
90	71h	11h	Next Address
91	16h	16h	Next Address
79	16h	16h	Sequencer Start Address

Sector Identification (Read ID operation)

The sector identification function consists of reading the ID fields of the sector to determine the address of the current sector. The function is performed with a comparison of the Address Mark and then capturing the ID header fields on the stack, achieved by programming the Stack Enable bit in the CONTROL field of the Sequencer. Also, any of the data may be compared against the sequencer's DATA field by programming both the DATA field and the Compare Enable bit in the CONTROL field. The sector identification is considered successful if no ECC error occurs. A branch condition is usually programmed at the end of the sector identification function such that if the compared bytes match and there is no ECC error then the sequencer will continue execution of the succeeding steps. Figure 10 shows a flow chart of the disk read operation.

Once the Sequencer Start Address Register (Reg 79) is loaded with the start address 16h, the sector identification process begins:

ADDRS DESCRIPTION

13 Read Gate is asserted, VFO lock on is initiated and the bit ring is stopped. After VFO lock ON occurs, the AIC-6110 ENDEC looks for the address mark pattern on the incoming encoded data at the 2,7 RLLI input pin. Detection of the address mark will lock the ENDEC to the NRZ bit boundary and start passing decoded NRZ data to the serializer/deserializer. Note that the address mark byte 5Eh is filtered out by the ENDEC during the read operation and thus is not seen by the SERDES. Next, when the sync byte A1 is detected by the SERDES, the bit ring starts to run again and synchronizes to the data byte boundaries.

05 The sync byte from the disk is compared with the contents of the current sequencer data field (A1h) to qualify it as a valid sync byte.

06 Then the ID fields are read in - Cylinder

07 Head

08 Sector

09 and Flag are compared to see if it is the desired sector.

0A The sequencer then reads the ID ECC bytes and checks to see if there is any error.

If an ECC error or a compare mismatch occurs then the Sequencer stops and the error condition (ECC error and stopped) is reflected in the Disk Status Register, (bit 4 and bit 2, Reg 79). An interrupt will be generated if the corresponding mask has been enabled in the Interrupt Mask Register 1, (bit 4 and bit 3, Reg 61).

If the En Comp EQ Int mask is set in the Interrupt mask register, then an interrupt will be generated, if the ID fields read from the disk, while the COMPARE ENABLE bit in the CONTROL field of the Sequencer is set, match the corresponding bytes in the DATA field of the Sequencer.

The Disk Status Register (Reg 79) will have bit 1 set if the byte read from the disk is less than the Sequencer RAM DATA field byte.

These bits are valid only after the ECC bytes have been read in. If the ECC check is successful then the next address is executed.

14 Read Gate will be deasserted after two bytes into the ID Postamble. This then completes a Read ID operation successfully.

Sector Read (Read Data operation)

If a disk read operation is in progress then the Read ID operation will be followed by a Read Data operation.

ADDRS DESCRIPTION

15 Read Gate is asserted and the bit ring is stopped. The bit ring will restart after successful VFO lock on, address mark detect and sync detect.

0E The sync byte A0h is compared to qualify it as the data field sync.

17 The data field identifier (F8h) is read and compared as an additional check.

0F The AIC-6110 then reads the Data field of the sector. If the En Data Xfer mask is set in the Interrupt Mask Register 1, (bit 6, Reg 61), then an interrupt will be generated and the status is reflected in the Disk Status Register (bit 6, Reg 79).

If the AIC-6110 is performing a multi-sector read, then the microprocessor can update the ID field, Register E6 - E8 with the next sector ID while the data is being transferred to the buffer memory.

10 The Data ECC checking is done next. If all compares matched and no ECC error occurred, the Sequencer will proceed to the next address. Otherwise the Sequencer will stop.

11 The Read Gate is deasserted automatically after the ECC field in the ID Postamble. The AM active bit will be reset in the Disk Status Register (bit 7, Reg 79) when the ECC function is complete.Test for ECC error or error from the compare bit. If either is set, it implies that a Data field error requiring a user-specified correction process is implied. If this is the last sector, then load Register 79 with 1Fh and monitor the Stopped interrupt status in the Disk Status Register (bit 4, Reg 79) or else repeat the Read ID - Read Data operation again on the new sector.

Sector Write (Write Data operation)

The disk write operation consists of a Read ID operation followed by a write of the Data field of the sector. The sequencer registers are setup as outlined in the table above for a write mode.

ADDRS DESCRIPTION

0A The write operation begins when the CONTROL field in the current sequencer word activates Write Gate. Write Gate is activated in the write splice after the ID ECC field for a write data operation. Once the Read ID function has been completed and it is the desired sector, the Sequencer proceeds with the write data operation by jumping to address 0C.

0C Write Gate is asserted and sixteen bytes of 00h pattern are written for the Data VFO sync field (Data Preamble).

0D Then the Data Address Mark byte of 5E is written on the disk.

0E Followed by a Data Sync byte of A0.

0F Then the AIC-6110 writes the Data field from the buffer memory, indicated by a Data Xfer interrupt indicated in the Disk Status Register, (bit 6, Reg 79), if the corresponding mask has been disabled in the Interrupt Mask Register 1.

If a multiple sector write is in progress, then the microprocessor can update the Registers E6 - E8 with the next sectors ID while data is being transferred while data is being written to the disk.

10 The Data ECC bytes are then written to the disk.

11 Followed by a two byte Pad (Data Postamble) of 00h

16 and the Write Gate is deasserted. If another sector is to be written, then the Sequencer repeats the sequence starting at address 13 else If no more sectors are to be written then the Register 79 is loaded with 1Fh and the microprocessor waits for the interrupt generated due to the stopped condition in the Disk Status Register (bit 4, Reg 79), provided the corresponding mask is enabled in the Interrupt Mask Register 1, indicating the end of the write sequence. A flow chart for the write operation is shown in Figure 11.

7.5.4 Extended Data Handling with the AIC-6110

The AIC-6110 Disk Sequencer is capable of handling a number of extended disk operations. It allows the use of variable length sectors through some manipulation, although the Data field of the Sequencer RAM allows a maximum size of 256 bytes. Multisector read and write operations are possible and certain special operations like Verify sector and Search sector data can be accomplished.

Variable Sector Size: The AIC-6110 has an eight-bit counter (when the data xfer bit is set in the CONTROL field) used for specifying the size of a sector. The sector size is loaded from the COUNT field of the Sequencer. The COUNT field is programmable and, by setting the field to any value from 00h to FFh, any sector length up to 256 bytes can be transferred.

If a sector size greater than 256 bytes is desired then the AIC-6110 needs to be manipulated to achieve sector sizes from 256 bytes to a full track. Larger sector sizes can be achieved by using multiple sequencer words to implement the desired sector size. The next address field of the first word points to the next and so on. For example a 532 byte sector would require three Sequencer words - two with a field count of 256 bytes and a third with a field count of 20 bytes to make up the 532 byte sector size.

Another approach would be to prevent the sequencer from jumping to the next address at the expiration of the sector count, but repeat the present address until the desired sector size has been achieved. By setting the Inhibit Carry bit in the Operation Control Register, (bit 7, Reg 7A) during a data transfer before the expiration of the count in the COUNT field of the sequencer, the AIC-6110 sequencer will be inhibited from going on to the next sequencer address and another 256 bytes could be transferred. The Inhibit Carry bit will automatically be reset when the sector counter overflows. For example a 532 byte sector could be transferred by setting up the COUNT field of the sequencer word corresponding to the Data field with 14h and enabling the Inhibit Carry bit twice.

Multisector Read/Write operations: Multisector read or write operations can be accomplished by two methods. The easiest approach is to load the ID information for the next sector to be accessed (Reg E6 - E9), while Data Transfer is in progress for the previous sector and restarting the read or write operation immediately after the present sector.

The data xfer condition can be easily monitored by observing the Data Xfer bit in the Disk Status Register, (bit 6, Reg 79). The other approach is to load multiple ID field values and a single read/write data field in the sequencer and setting up the Branch/NA Register (Reg 78).

Verify Sector: A verify sector operation can be accomplished by setting the Suppress Transfer bit in the Operation Control Register, (bit 5, Reg 7A) and then performing a Disk Read Data operation. The incoming data will be verified for good ECC, but will not be transferred to the buffer memory.

Search Sector Data : The sector read function can be modified to a Search Data function. When the Data field of a sector is being read, the contents of the buffer memory will be compared, byte-for-byte, with the incoming data from the drive. The comparison of the DATA field is enabled by setting the Search Operation bit in the Operation Control Register, (bit 4, Reg 7A) and the Compare Enable bit in the CONTROL field of the Sequencer word that starts the data transfer. The buffer controller should also be setup for a write operation by setting the *WOP/ROP bit in the Control Register 1, (bit 4, Reg 63) to zero. The result of the comparison is latched into the Disk Status Register, (bits 0 & 1 , Reg 79). The Search operation bit in the Operation Control Register and the Compare Enable bit in the sequencer CONTROL word must be reset by the microprocessor after completion of the search data operation.

7.5.5 Error Detection and Correction

The serial data flow portion of the disk formatter/sequencer consists of a 16-bit CRC/32-bit ECC/48 bit ECC generator and the serializer/deserializer. Data to be written to the disk enters the AIC-6110 in a byte-wide format. It is serialized and run through a the ECC generator. The serial bit stream is then shifted out to the drive as NRZ or 2,7 RLL.

The choice between the 16-bit CRC-CCITT polynomial or the 32/48-bit computer generated ECC polynomial can be programmed in the COUNT field of the Sequencer RAM. It can also be fully suppressed to use external ECC if desired.

The 16-bit CRC-CCITT polynomial used is : $x^{16} + x^{12} + x^5 + 1$

The 32/48-bit ECC polynomials are computer generated. The 32-bit polynomial is given in this document and the 48-bit polynomial is provided under license from Adaptec. When used for error detection only, any single burst not exceeding 32/48 bits in length is guaranteed to be detected regardless of sector size.

Double burst error detection is guaranteed, provided the sum of the two bursts does not exceed 11 bits for a 32-bit polynomial and 16-bits for a 48-bit polynomial. The miscorrection probability for a selected polynomial depends on the sector size (s), correction span (x), polynomial size (n). It is given by the following formula - $P_{mc} = s * 2^{(x-1)} / 2^n$

For a sector size of 512, for the optimum probability of miscorrection, it is recommended that for correction of double burst errors, a correction span of 5-bits be used for MFM type data and a correction span of 8-bits be used for 2,7 RLL type data.

During a write operation, 32/48-bits of ECC redundancy are calculated by shifting the sync byte and data field bytes through the serial shift register. The sequencer word that sets AM data type (bit 7 of the COUNT field), while Write Gate is active initiates the ECC calculation.

The 32/48 ECC redundancy bits are then appended to the end of the data and written to the disk. When the data is read back, a syndrome is generated by shifting the sync byte, data and ECC redundancy bits through the serial shift register.

When Write Gate is asserted, ECC is cleared when the AM data type bit in the Sequencer is active, and ECC calculation will begin when the AM bit goes inactive. When Read Gate is asserted, ECC calculation begins after detection of the sync byte. In case of no error, the syndrome will be all zeros after the ECC bytes are read in and Read Gate is deasserted. The ECC error interrupt status in the Disk Status Register (bit 2, Reg 79) will be set if an ECC error occurs and an interrupt will be generated if the corresponding mask is enabled in the Interrupt Mask Register 1, (Reg 61).

It is necessary for the PAD byte following the ECC/CRC data field in the sector layout to have a value of 00h to ensure proper operation of the ECC/CRC circuitry. If an ECC error takes place, then the microprocessor should determine if the error is correctable and if so, start the correction procedure. The AIC-6110 provides an ECC Correction shift register/counter (Reg 72h) to simplify the correction procedure. It can be used as a shift register to reverse the syndrome, or as a counter to count the number of bytes that have been shifted during the correction algorithm. The correction algorithm consists of five steps :

1. Reverse the syndrome and the polynomial.
2. Determine if the error is correctable.
3. Determine the error location from the beginning of the record.
4. Calculate the error length.
5. Correct the erroneous data in the memory by XOR-ing the error pattern with the data.

5.5.5.1 Using the 32-bit polynomial with an 8-bit correction span for ECC

The ECC polynomial has to be setup during the initialization process. The microprocessor selects the size of the ECC polynomial used in the ECC Control Register, (bit 7, Reg 71). The WAM signal timing is selected in the WAM Control Register, (Reg 7B is set to 10h) and the sync byte to be used for compare during sync detect is setup in the Sync Control Register, (Reg 7C is set to A0), while the Sync Compare Control Register must be set to 07, (Reg 7F). The 32-bit polynomial used is as follows

FORWARD POLYNOMIAL : $x^{30} + x^{24} + x^{18} + x^{14} + x^{8} + x^{7} + x^{2} + 1$

REVERSE POLYNOMIAL : $x^{30} + x^{25} + x^{24} + x^{18} + x^{14} + x^{8} + x^{2} + 1$

The forward polynomial is setup during power-on initialization. The reverse polynomial is only used during error correction time, and hence must be replaced by the forward polynomial once the correction process is completed. The register setup for programming the 32-bit polynomial in the AIC-6110 is given below.

Register	Value	Description
72	FF	Forward/Reverse Polynomial
73	FF	Forward/Reverse Polynomial
74	C2	Forward Polynomial
	82	Reverse Polynomial
75	20	Forward/Reverse Polynomial
76	82	Forward/Reverse Polynomial
77	20	Forward Polynomial
	21	Reverse Polynomial
7E	00	Don't Care

When an ECC error is detected, the sector must be retried before a correction is attempted. Once it is determined that it is indeed a hard error, the microprocessor off loads the 32-bit syndrome into the local RAM. The syndrome is shifted back into the ECC register in reverse order, swapping the syndrome end to end. The Reverse polynomial is loaded. Then the ECC is shifted until all bits, except the high order (24-31) bits are zero implying that the error is correctable. If the number of shifts is greater than the number of bits in the record, then the error is not correctable and the sector is mapped as a bad sector. For a correctable error, the number of shifts represents the displacement of the error from the end of the record (the last bit of the ECC). The error pattern is located in bits 24-31 of the ECC register. This pattern is XORed with the appropriate bytes in memory to correct the data bytes in error.

The detailed programming steps pertaining to the AIC-6110 are outlined below.

1. After a read error is detected, disable feedback by setting Reg 71 = C4h.
2. Store 00h in RAM (location x).
3. Store 00h in RAM (location x+1).
4. Store contents of Reg 73h in RAM (location x+2).
5. Shift ECC eight times by setting Reg 71 = C6h eight times.
6. Store contents of Reg 73 in RAM (location x+3).
7. Shift ECC eight times by setting Reg 71 = C6h eight times.
8. Store contents of Reg 73 in RAM (location x+4).
9. Shift ECC eight times by setting Reg 71 = C6h eight times.
10. Store contents of Reg 73 in RAM (location x+5).
11. Clear ECC and keep feedback disabled by setting Reg 71 = C8h and then to C4h.
12. Test if RAM location (x+5), bit 0 is set. If set, then load Reg 71h = 07h. If not set, then load Reg 71 = 06h. Repeat this for the other seven bits at that location.
13. Repeat the test in step 12 for RAM locations (x+4), (x+3), (x+2), (x+1) and x until all 32-bits of the syndrome are loaded into the ECC in reverse order along with the 16 zero bits.
14. Load Reg 74 = 82h and Reg 77 = 21h to enable the reverse polynomial and disable the forward polynomial.
15. Compute record length in bits : Number of bits per data field = ECC + Data + AM + SYNC.For example a 256 byte record has (4 + 8 + 256 + 2)*8 = 2160 bits.
16. Enable feedback by setting Reg 71 = C0h.
17. Shift ECC once by setting Reg 71 = C2h and increment a software counter.
18. Test to see if the software counter is greater than the record length. If yes, the error is not correctable; re-enable the forward polynomial and end correction operation.
19. If Reg 72 = 00h then go to step 20, or else go to step 17.
20. Subtract hardware offset of 7 from the shift count. If a correctable error is located within the ECC or SYNC or AM bytes i.e. (shift count is less than 33), then the data field is not corrupted and hence no further action is required. Otherwise, subtract 32 from the shift count.21. The bit displacement (shift count) must now be converted to a byte offset by right shifting the count three times. This value of the shift count equals the bit displacement from end of the record.
22. Now the contents of Reg 73 is the mirror image of the error pattern. Form the error mask data (two bytes) by concatenating Reg 73 with a zero byte.
23. Get the shift count (E) for error mask data by extracting the lower three bits from the shift count obtained in step 20.
24. Right shift the error mask data with the most significant bit set to zero. Repeat this process on 'E' one more time.
25. Mirror the error mask data byte-by-byte. For example, if the original error mask data is 5C 9A, then after mirroring, it becomes 3A 59.
26. The two byte error mask data may now be XORed with the data in memory (available in Reg 70 through a Memory PIO), to correct the error. The byte offset to locate the data byte in error is available from step 21.

NOTES:

1. For proper 32-bit ECC polynomial operation, Reg 72 and Reg 73 must be loaded with FFh.

2. If a correction span of 5-bits is desired, then the following modification is necessary.

Step 17: Go to Step 13 if Reg 72 = 00h and Reg 73 - bits 0, 1 & 2 are zero.

Step 18: The mirror image of the error pattern are bits 3 to 7 of Reg 73.

5.5.5.2 Using the 48-bit polynomial with an 8-bit correction span for ECC

The ECC polynomial has to be setup during the initialization process. The microprocessor selects the size of the ECC polynomial used in the ECC Control Register, (bit 7, Reg 71). The WAM signal timing is selected in the WAM Control Register, (Reg 7B is set to 40h) and the sync byte to be used for compare during sync detect is setup in the sync control register, (Reg 7C is set to A0), while the sync compare control register must be set to 07, (Reg 7F).

The details of the 48-bit polynomial used are available under a license agreement from Adaptec Inc.

The forward polynomial for error detection and the reverse polynomial for error correction, must be loaded into the AIC-6110 in Registers 72 through 77, very similar to that of the 32-bit polynomial as discussed in the previous section.

When an ECC error is detected, the sector must be retried before a correction is attempted. Once determined, that it is indeed a hard error, the microprocessor off loads the 48-bit syndrome into the local RAM. The syndrome is shifted back into the ECC register in reverse order, swapping the syndrome end to end. The Reverse polynomial is loaded.

Then the ECC is shifted until all bits, except the high order (40 - 47) bits are zero. This implies that the error is correctable. If the number of shifts is greater than the number of bits in the record, then the error is not correctable and the sector is mapped as a bad sector.

For a correctable error, the number of shifts represents the displacement of the error from the end of the record (the last bit of the ECC). The error pattern is located in bits 40-47 of the ECC register. This pattern is XORed with the appropriate bytes in memory to correct the data bytes in error.

The detailed programming steps pertaining to the AIC-6110 are outlined below.

1. After a read error is detected, disable feedback by setting Reg 71 = 44h.
2. Store contents of Reg 73h in RAM (location x).
3. Shift ECC eight times by setting Reg 71 = 46h, eight times.
4. Store contents of Reg 73h in RAM (location x+1).
5. Shift ECC eight times by setting Reg 71 = 46h, eight times.
6. Store contents of Reg 73h in RAM (location x+2).
7. Shift ECC eight times by setting Reg 71 = 46h, eight times.
8. Store contents of Reg 73 in RAM (location x+3).
9. Shift ECC eight times by setting Reg 71 = 46h eight times.
10. Store contents of Reg 73 in RAM (location x+4).
11. Shift ECC eight times by setting Reg 71 = 46h eight times.
12. Store contents of Reg 73 in RAM (location x+5).
13. Clear ECC and keep feedback disabled by setting Reg 71 = 48h and then to 44h.
14. Test if RAM location (x+5), bit 0 is set. If set, then load Reg 71h = 07h. If not set, then load Reg 71 = 06h. Repeat this for the other seven bits at that location.
15. Repeat the test in step 14 for RAM locations (x+4), (x+3), (x+2), (x+1) and x until all 48-bits of the syndrome are loaded into the ECC in reverse order .
16. Load Reg 72 through 77 to enable the reverse polynomial and disable the forward polynomial.
17. Compute record length in bits : Number of bits per data field = ECC + Data + AM + SYNC. For example a 256 byte record has (6 + 8 + 256 + 2)*8 = 2176 bits.

18. Enable feedback by setting Reg 71 = 40h.
19. Shift ECC once by setting Reg 71 = 42h and increment a software counter.

20. Test to see if the software counter is greater than the record length. If yes, the error is not correctable; re-enable the forward polynomial and end correction operation.
21. If Reg 72 = 00h then go to step 22, or else go to step 19.
22. Subtract hardware offset of 7 from the shift count. If a correctable error is located within the ECC or SYNC or AM bytes i.e. (shift count is less than 49), then the data field is not corrupted and hence no further action is required. Otherwise, subtract 48 from the shift count.
23. The bit displacement (shift count) must now be converted to a byte offset by right shifting the count three times. This value of the shift count equals the bit displacement from end of the record.
24. Now the contents of Reg 73 is the mirror image of the error pattern. Form the error mask data (two bytes) by concatenating Reg 73 with a zero byte.
25. Get the shift count (E) for error mask data by extracting the lower three bits from the shift count obtained in step 22.
26. Right shift the error mask data with the most significant bit set to zero. Repeat this process on 'E' one more time.
27. Mirror the error mask data byte-by-byte. For example, if the original error mask data is 5C 9A, then after mirroring, it becomes 3A 59.
28. The two byte error mask data may now be XORed with the data in memory (available in Reg 70 through a Memory PIO), to correct the error. The byte offset to locate the data byte in error is available from step 23.

Note : If a correction span of 5-bits is desired, then the following modification is necessary. Step 17: Go to Step 15 if Reg 72 = 00h and Reg 73 - bits 0, 1 & 2 are zero. Step 18: The mirror image of the error pattern are bits 3 to 7 of Reg 73.

7.5.6 2,7 RLL ENDEC Operation

The type of data encoding used is selectable through software, (bit 3, Reg 56) in the AIC-6110. It can generate NRZ type data or if the on chip encoder/decoder is enabled it can generate 2,7 RLL type data. This ENDEC is very similar to Adaptec's AIC-270. The NRZ type data encoding can be used directly with ESDI type drives or with the use of an external MFM or 1,7 RLL encoder/decoder.

The ENDEC works with an external Data separator like Adaptec's AIC-6225 to read/write data to the disk. Figure 12 shows the hardware interface between the AIC-6110 and the AIC-6225. The disk interface pins in the AIC-6110 are dual function in nature and are configured depending on the NRZ or 2,7 RLL mode selected. This is shown in the table below.

AIC-6110 Pin	Type	NRZ mode	2,7 RLL mode
INPUT/2FOSC	I	INPUT	2FOSC
OUTPUT	O	OUTPUT	OUTPUT
SECTOR/RST1S	I	SECTOR	1'S RESET
RG/2,7RLLO	O	READ GATE	2,7 RLLO
INDEX	I	INDEX	INDEX
WPCOMP	O	---	WPCOMP
WG	O	WRITE GATE	WRITE GATE
NRZ/2,7RLLI	I/O	NRZ DATA	2,7 RLLI
*WAM/*AMD/VFOENBL	I/O	*WAM/*AMD	VFOENBL
RRC/2FVFO	I	RRC	2FVFO

Write to disk

Data is written to the disk during a regular write data operation and a format operation. The 2FOSC is used as the write oscillator clock source which must be twice the frequency of the RLL data. During a format operation, the AIC-6110 automatically generates one of two possible VFO sync fields (preamble patterns). The VFO sync field is generated between the time write gate goes high and the first low to high transition on the internal NRZ data line. The two possible patterns are shown below, selectable using the VFO OPT bit in the Mode Select Register, (bit 1, Reg 56).

1001001001001001...	(VFO OPT = 0)
1000100010001000...	(VFO OPT = 1)

On the first low to high NRZ transition, the device stops sending the preamble data and generates an address mark. There are two possible address mark patterns used which are selectable using the AM OPT bit in the Mode Select Register, (bit 0, Reg 56). The first option (AM OPT = 1), uses a 5EA hexadecimal pattern that satisfies the 2,7 constraints, but never occurs during a normal encoding sequence. The second option (AM OPT = 0), uses a FF hexadecimal pattern for syncing. This option requires some additional external circuits to prevent false syncing, such as a DC erased area on the disk.

For a regular write operation of the Data field, the AIC-6110 ENDEC block generates 2,7 RLL encoded data on the 27RLLO line to the disk. If precompensation is desired on the 2,7 RLLO output data, then it can be enabled through the Mode select Register, (bit 2, Reg 56). An external resistor and capacitor network will establish the desired ramp rate on the WPCOMP ramp control open drain output. The output is redirected into the chip where a set of comparators establish the desired delay for precompensation. The early and late precompensation values can be programmed in registers 57 and 58. The incremental precompensation delay is equal to 0.05 * RC. The precompensation technique adopted in the AIC-6110 is illustrated below :

```
time  ------------------------------------------------------->

...0001000100100100000100010010010010001000...
      O   L  O  E     O   L  O  O  E   O

O = write on-time ;     L = write late ;     E = write early ;
```

Read from Disk

In the case of a disk read operation, the 2,7 RLL ENDEC block works in conjunction with the external VFO circuitry (AIC-6225) to convert 2,7 RLL data to NRZ data internally. Inside the AIC-6110, the disk controller block activates the Read Gate signal to the ENDEC block (the Read Gate signal will not appear on the output when in the 2,7 RLL mode) to begin the read operation and starts to look at the clock on the 2FVFO input from the Data Separator. This triggers a sequence of events which is outlined below for a better understanding of the ENDEC operation.

1. An internal counter will count 24 transitions of the 1FOSC input, (derived from 2FOSC/2). If at any time a greater than expected space between bit transitions is detected, a 1's Reset will occur from the external one shot, which will restart the counter. This ensures that the VFO is enabled only when reading the VFO sync field (preamble).

2. When the count reaches 24, the output signal VFOENBL is asserted which allows the Data Separator to start syncing on the disk data input.

3. The counter continues to count for 72 more transitions of the 1FOSC input, (derived from 2FOSC/2). During this period, if a is rest occurs, the counter will reset and the whole process will resume from step 1. If a 1's Reset does not occur by count 96, an internal latch will be set that allows for the detection of an address mark and disallows any further 1's Reset from affecting the counter. This also switches the disk controller block's Read Reference Clock (RRC) source from the write oscillator frequency to the VFO frequency, (2FVFO/2 input).

4. The counter continues to run while the AM detector is looking for an address mark. If the counter reaches count 128 without an address mark detected, the counter will be reset, the VFOENBL output will go low, the Read Reference Clock source to the Disk Controller block in the AIC-6110 will switch back to the write oscillator, and the search for an address mark will be disabled. The whole process will restart from step 1 at this time.

5. However if the address mark is detected before count 128, any further contributions from the counter will be disabled, and the RLL input data will be read, decoded, and passed on to the disk controller block in the AIC-6110.

6. Once the read operation is completed, the disk controller block in the AIC-6110 turns off Read Gate, thus resetting the counter and stopping the transfer of data.

This summarizes the basic operation of the ENDEC block in the AIC-6110. This block is designed to be capable of handling data transfers from 5 Mbits/sec. to 15 Mbits/sec.. A typical system design for a embedded SCSI drive using 2,7 RLL encoding scheme is shown in

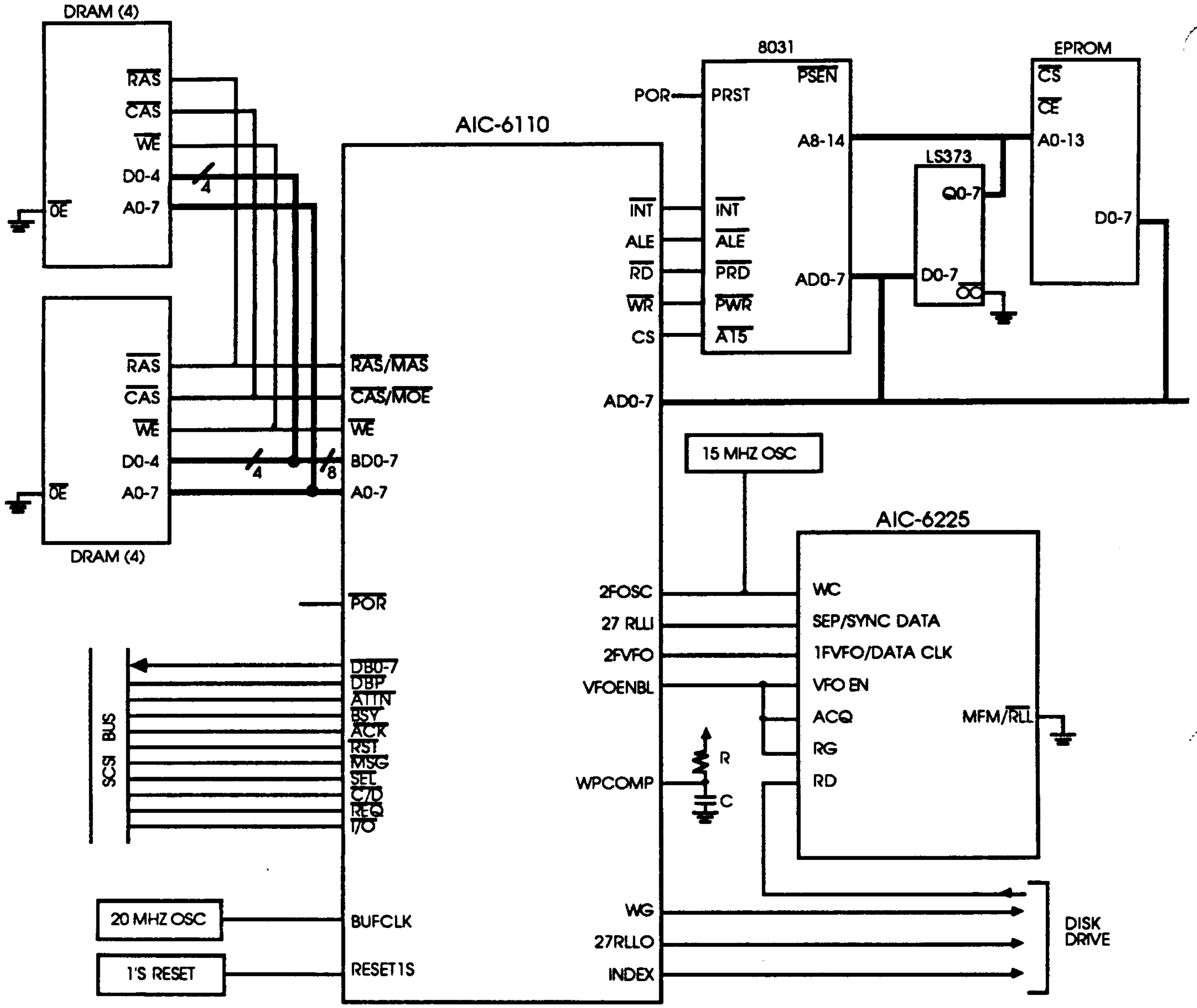


FIGURE 10. TYPICAL SYSTEM CONFIGURATION

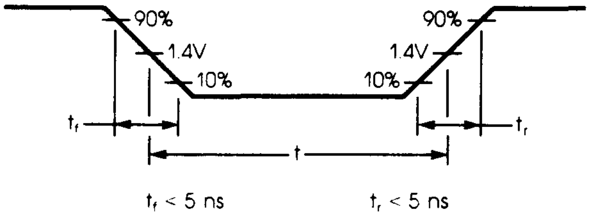
NOTE: The following timings are advanced information and subject to change without notice.

8.1 Absolute Maximum Ratings:

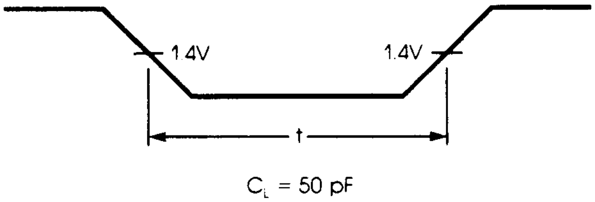
Ambient Temperature Under Bias O° C to 70° C
Storage Temperature -65° C to 150° C
Voltage on any pin -0.5 to Vcc+0.5 Volts
Power Dissipation 0.8 Watts
Power Supply Voltage 0 to 7 Volts

AC CHARACTERISTICS (Conditions Vcc = 5.0V ± 5%, 0°C < T < 70°C)

A.C. INPUT TIMING CONDITIONS



A.C. OUTPUT TIMING CONDITIONS

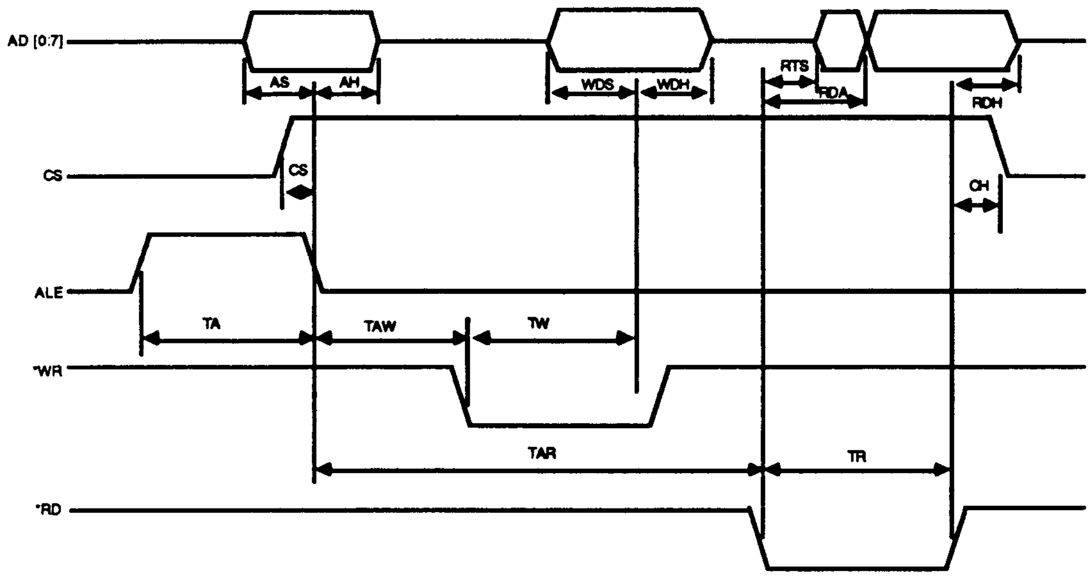


8.2 DC Parameters

(Operating Conditions: Vcc = 5.0V +/- 5%, 0° C < T < 70° C)

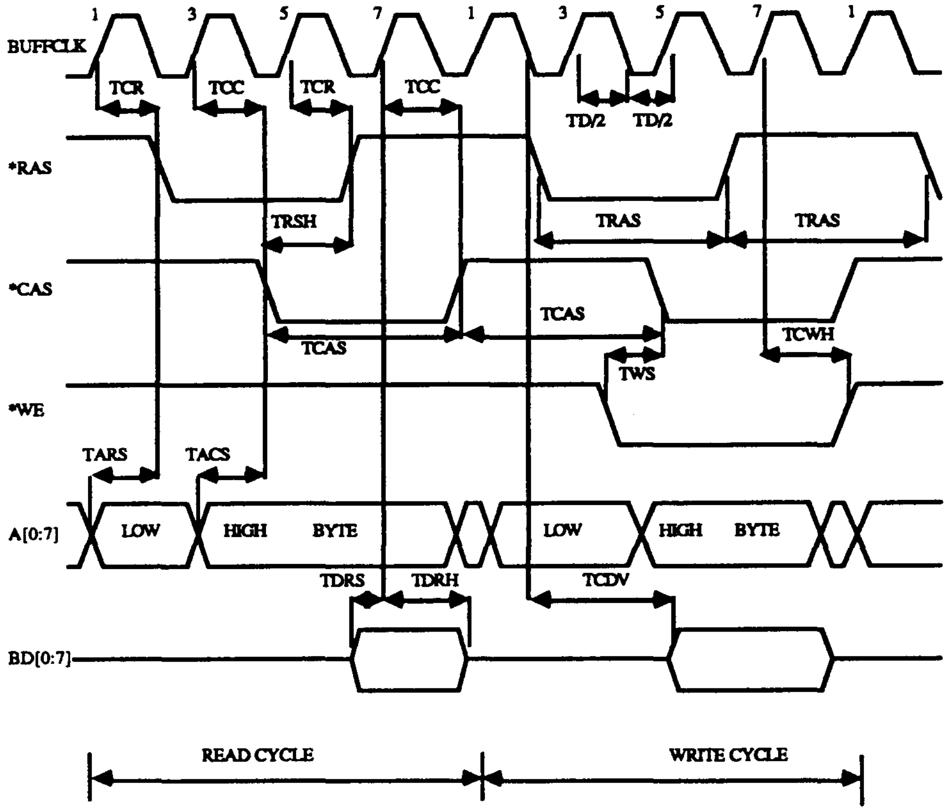
SYMBOL	*PARAMETER*	*MIN*	*MAX*	*UNITS*	*CONDITIONS*
IDD	QUIESCENT CURRENT		5	mA	VDD=5.25
IDD	OPERATING CURRENT		150	mA	
VIH	INPUT HIGH VOLTAGE	2.0		V	
VIL	INPUT LOW VOLTAGE		0.8	V	
IL1	INPUT LEAKAGE	-10	10	μ A	
H	HYSTERSIS	200		mV	
VOH1	OUTPUT HIGH VOLTAGE (ALL OUTPUTS OTHER THAN SCSI)	2.4		V	IOL=-400 μ A
VOH2	OUTPUT HIGH VOLTAGE (SCSI BUS INTERFACE SIGNALS)				OPEN DRAIN
VOL1	OUTPUT LOW VOLTAGE (ALL OUTPUTS OTHER THAN SCSI)		0.5	V	IOL=5 mA
VOL2	OUTPUT LOW VOLTAGE (SCSI BUS INTERFACE SIGNALS)		0.5	V	IOL=48 mA
VOL3	OUTPUT LOW VOLTAGE (*INT ONLY)		0.5	V	IOL=10 mA
Rpcomp	ALLOWABLE RESISTANCE (WPCOMP)	200		Ω	

8.3 Microprocessor Interface Timing



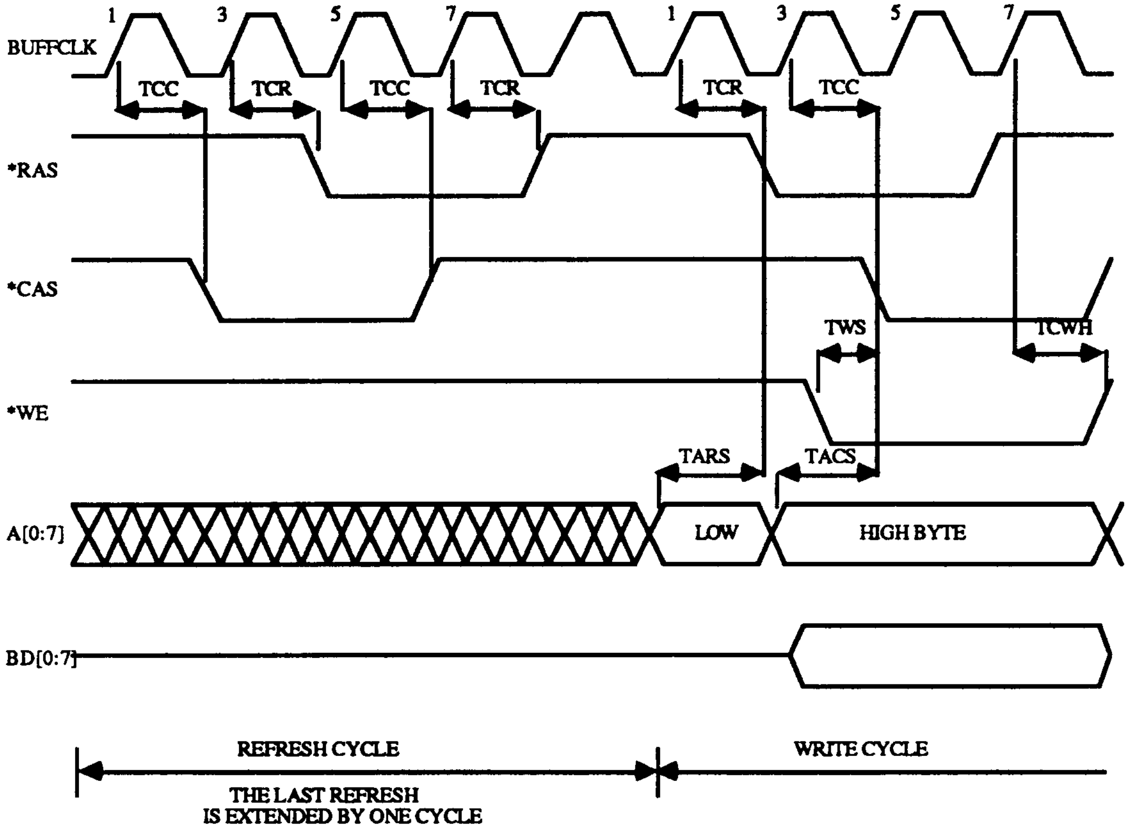
SYMBOL	PARAMETER	MIN	MAX	UNITS
TA	ALE WIDTH	15		nS
TAW	ALE TO *WR	15		nS
TAR	ALE TO *RD	15		nS
TW	*WR WIDTH	40		nS
TR	*RD WIDTH	40		nS
AS	ADDRESS VALID TO ALE	7.5		nS
AH	ALE TO ADDRESS VALID	5		nS
CS	CS TO ALE	5		nS
CH	*RD OR *WR TO CS	0		nS
WDS	DATA VALID TO *WR	60		nS
WDH	*WR TO DATA VALID	5		nS
RTS	*RD TO DATA ACTIVE	15		nS
RDA	*RD TO DATA VALID		50	nS
RDH	*RD TO DATA INACTIVE	10	30	nS

8.4 DRAM Read/Write Timing



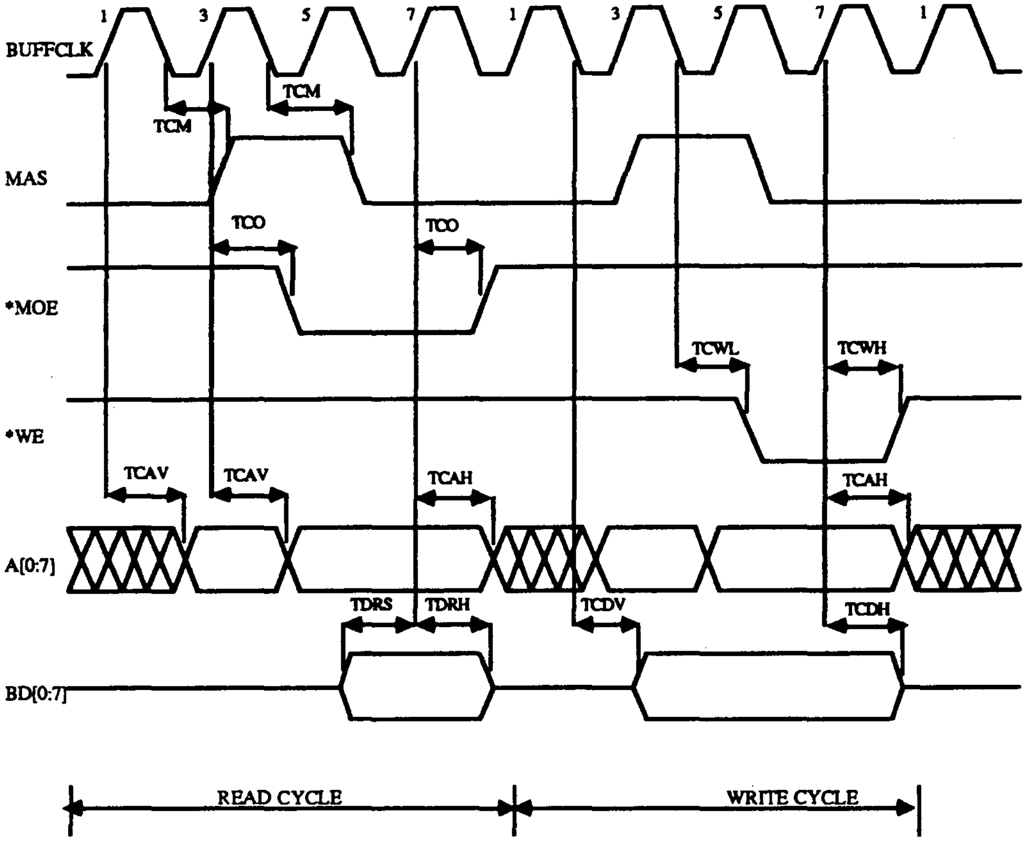
SYMBOL	PARAMETER	MIN	MAX	UNITS
TD/2	BUFFCLK/2 IN DRAM MODE	25		nS
TRAS	*RAS LOW AND HIGH WIDTH	2TD		nS
TCAS	*CAS LOW AND HIGH WIDTH	2TD		nS
TCR	CLK TO *RAS	10	40	nS
TCC	CLK TO *CAS	10	40	nS
TRSH	*RAS HOLD TIME	TD-5		nS
TARS	ADDRESS TO *RAS SETUP	3	14	nS
TACS	ADDRESS TO *CAS SETUP	3	14	nS
TDRS	READ DATA SETUP	0		nS
TDRH	READ DATA HOLD	10		nS
TWS	*WE TO CAS SETUP	5	21	nS
TCWH	CLK TO *WE HIGH	4	15	nS
TCDV	CLK TO DATA VALID	7	26	nS
TRCD	RAS LOW TO CAS LOW	T-10	T+10	nS

8.5 DRAM Refresh Timing



SYMBOL	*PARAMETER*	*MIN*	*MAX*	*UNITS*
TCR	CLK TO *RAS	10	40	nS
TCC	CLK TO *CAS	10	40	nS

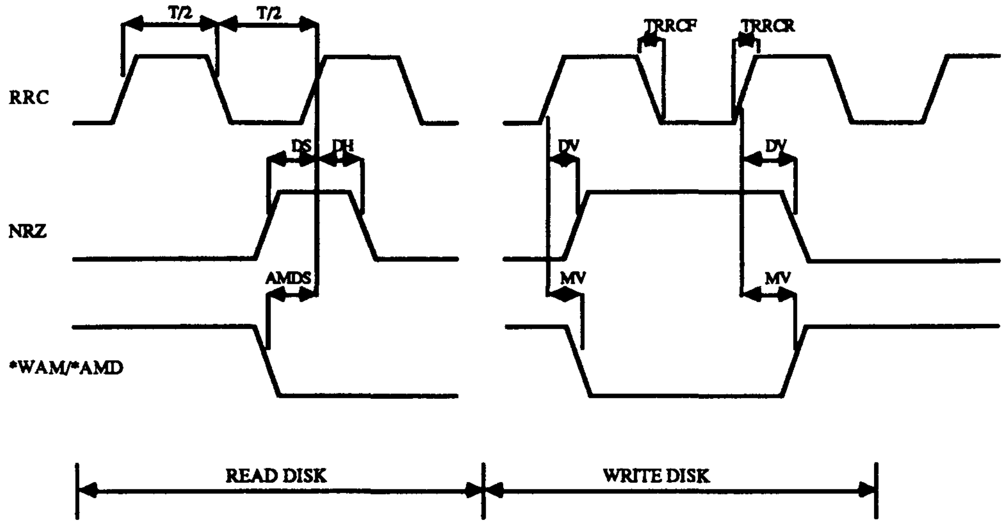
8.6 SRAM Read/Write Timing



SYMBOL	PARAMETER	MIN	MAX	UNITS
TD/2	BUFF CLK/2 IN SRAM MODE	16		nS
TCM	CLK TO MAS	5	23	nS
TCO	CLK TO *MOE	10	40	nS
TCAV	CLK TO ADDRESS VALID	6	26	nS
TCAH	CLK TO ADDRESS HOLD	10	35	nS
TDRS	READ DATA SETUP	0		nS
TDRH	READ DATA HOLD	10		nS
TCWL	CLK TO *WE LOW	5	18	nS
TCWH	CLK TO *WE HIGH	4	15	nS
TCDV	CLK TO DATA VALID	7	26	nS
TWAH	*WE TO ADDRESS HOLD	5		nS
TWDH	*WE TO DATA HOLD	5		nS
TAVMH	ADDRESS VALID TO MAS	TD/2-6		nS

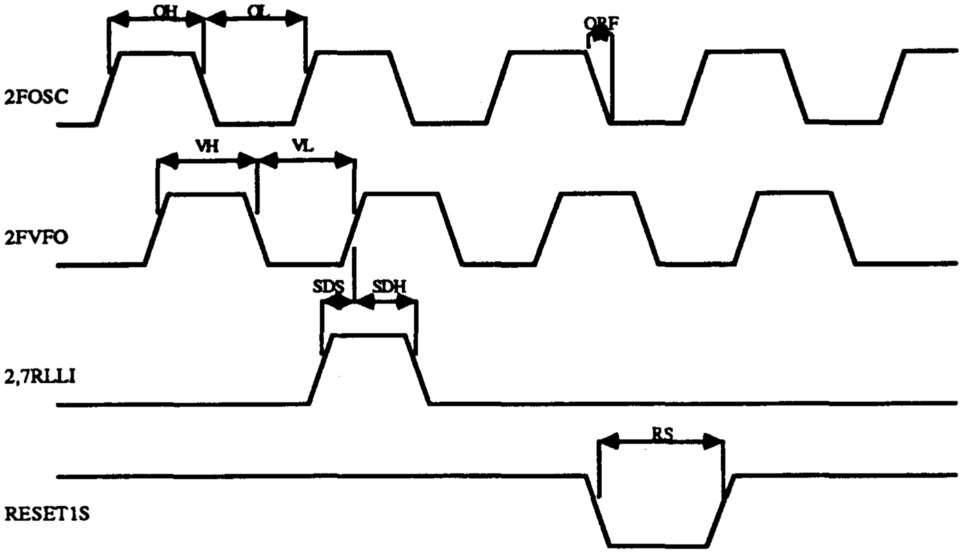
NOTE: SRAM access time must not be greater than 2TD-20 nS.

8.7 Disk Interface Timing



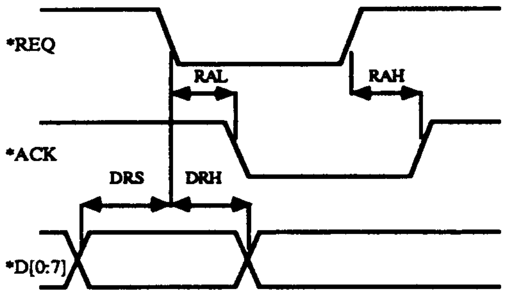
SYMBOL	PARAMETER	MIN	MAX	UNITS
T/2	RRC HALF CYCLE	20		nS
TRRCR	RRC RISE TIME		5	nS
TRCF	RRC FALL TIME		5	nS
DS	DATA IN SETUP	0		nS
DH	DATA IN HOLD	10		nS
AMDS	*AMD SETUP	0		nS
DV	RRC TO DATA OUT	5	25	nS
MV	RRC TO *WAM	5	25	nS

8.9 RLL Interface Timing

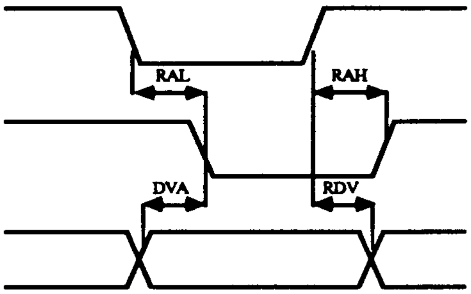


SYMBOL	PARAMETER	MIN	MAX	UNITS
OH	WRITE OSC HIGH	16		nS
OL	WRITE OSC LOW	16		nS
ORF	WRITE OSC RISE & FALL		5	nS
VH	2FVFO HIGH	16		nS
VL	2FVFO LOW	16		nS
SDS	STANDARD DATA SETUP	0		nS
SDH	STANDARD DATA HOLD	10		nS
RS	ONE'S RESET LOW	20		nS

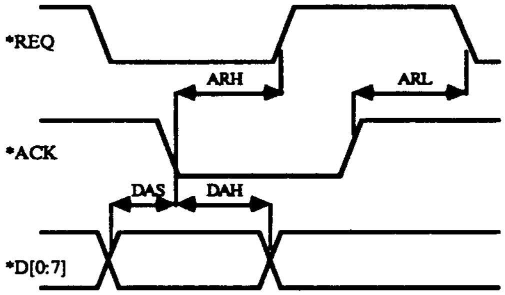
8.10 SCSI Interface Timing



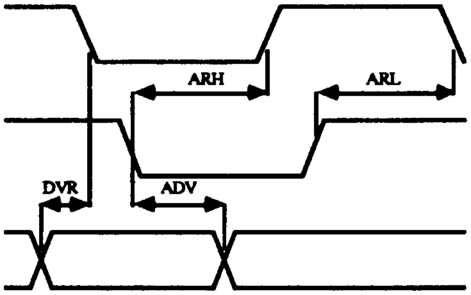
DATA FROM SCSI INITIATOR



DATA TO SCSI INITIATOR



DATA FROM SCSI TARGET



DATA TO SCSI TARGET

SYMBOL	*PARAMETER*	*MIN*	*MAX*	*UNITS*
DRS	DATA TO *REQ SETUP	50		nS
DRH	DATA TO *REQ HOLD	50		nS
DAS	DATA TO *ACK SETUP	50		nS
DAH	DATA TO *ACK HOLD	50		nS
DVA	DATA VALID TO *ACK	2TD		nS
RDV	*REQ TO DATA VALID	0		nS
DVR	DATA VALID TO *REQ	2TD		nS
ADV	*ACK TO DATA VALID	0		nS
RAL	*REQ TO *ACK LOW	100+T	100+2T	nS
RAH	*REQ TO *ACK HIGH		40	
ARL	*ACK TO *REQ LOW	100+T	100+2T	nS
ARH	*ACK TO *REQ HIGH		40	nS
SLR	FALLING SIGNAL SLEW	8	32	nS

NOTE: Slew is measured with 110/165 resistive and 300pF capacitive load.

SEQUENCER MAP FOR SOFT-SECTOR ESDI DRIVES

	E0 THRU F7 DATA		C0 THRU D7 COUNT		A0 THRU B7 CONTROL		80 THRU 97 NEXT ADRS.		ADRS. R79	BRANCH R78	COMMENTS
0	CYLINDER		0	0	1	2	0	1	0	1/1F	CYLINDER
1	HEAD		0	0	1	2	0	2	1		HEAD
2	SECTOR		0	0	1	2	0	3	2		SECTOR
3	FLAG		0	0	1	0	0	C	3		FLAG
4	DATA FILL		LENGTH		0	1	0	D	4		DATA FIELD
5	0	0	0	1	4	0	0	6	5		RG FOR ID
6	ID SYNC		8	0	1	2	0	0	6		ID SYNC CHARACTER
7	0	0	0	1	0	0	0	B	7		PAD 2 BYTES
8	0	0	0	0	C	0	0	9	8		WG @ END OF ID
9	0	0	GAP 2		8	0	0	A	9		DATA VFO SYNC AREA
A	DATA SYNC				0	2	0	4	A		DATA SYNC CHARACTER
B	0	0	0	1	4	0	0	A	B		RG FOR DATA
C	0	0	4	3	0	0	11=FMT 67=RD	69=WR	C		ID ECC
D	0	0	4	3	0	0	92=FMT 77=RD	12=WR	D	12/13	DATA ECC
E	0	0	0	0	C	0	10=FMT	12=WR	E		WG @ END OF DATA
F	0	0	0	B	0	0	0	6	F		WRITE VFO SYNC AREA
10	0	0	GAP 1 & 3		8	0	1	4	10		WG FOR GAP 1 & 3
11	0	0	0	2	0	0	0	8	11		IC ECC PAD - 3 BYTES
12	0	0	0	2	0	0	0	E	12		DATA ECC PAD - 3 BYTES
13	0	0	0	1	0	0	1	6	13		GAP 4 TILL END OF FORMAT
14	0	0	0	2	0	4	0	F	14		WRITE AM ENABLE
15	0	0	0	0	0	0	D	5	15	OE	FORMAT START ON INDEX
16	0	0	0	0	0	0	5	6	16		FORMAT STOP ON INDEX
17	0	0	0	0	0	4	D7=START 1F=STOP		17	05	R/W START ON AM FOUND
	7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0				

DATA (bits 7–0 of DATA)

DATA TYPE (bits 7–5 of COUNT): AM, ECC, RESERVED

FIELD CNT (-1) (bits 4–0 of COUNT)

CONTROL bits: ECC/RC, STACK ENABL, INVALID NRZ, OUTPUT, COMPARE EN, DATA XFER

NEXT ADRS.: bits 7–5 BRANCH CONTROL; bits 4–0 N.A. 1F = STOP

BRANCH CONTROL

RG • ECC = 1	RG • ECC = 0
000 = NO BRANCH	000 = NO BRANCH
001 = ECC ERROR STOP	001 = STOP ON INPUT
010 = NO COMPARE STOP	010 = STOP IN INDEX OR SECTOR
011 = ERROR OR NO COMPARE STOP	011 = STOP ON NOT EQUAL
100 = GOOD ECC AND COMPARE	100 = BRANCH ON CARRY
101 = ECC ERROR	101 = BRANCH ON INPUT
110 = NOT EQUAL	110 = BRANCH ON INDEX OR SECTOR
111 = ERROR OR NOT EQUAL	111 = BRANCH ON NOT EQUAL

NOTE: This software has been shown as an example sequencer map and can only be used with Adaptec chips.The use of this software is not guaranteed by Adaptec.

SOFT-SECTORED FORMAT SEQUENCE

The following example shows the operation of the AIC-6110 (programmed as an ESDI controller) during a soft-sector format operation. The example is presented using "Snap-shots" of the AIC-6110 and system activity descriptions. The time references assigned to each snapshot or event represent the approximate times associated with an ESDI disk drive. The drive is formatted for 256-byte sectors.

AIC-6110 STATE

Current Address:	1FH

System Activity: The AIC-6110 is stopped. Host initializes AIC-6110 registers:

Branch register = 0EH
Register D0H = Gap 1 length
Register E0H = I.D. Cylinder Value
Register E1H = I.D. Head Value
Register E2H = I.D. Sector Value
Register E3H = I.D. Flag Value

Host writes 15H to AIC-6110 sequencer start register (Register 79H).

AIC-6110 STATE

Current Address:	15H
Next Address:	15H Loop until Index
Branch Control:	110 (RG = 0) Branch on Index or Sector
Branch Register:	0EH
Control:	N.A.
Data Field:	N.A.
Count:	N.A.

System Activity: AIC-6110 waits for Leading edge of Index to begin writing Gap1. When Index is detected by the AIC-6110, the branch Active Flag is set.

AIC-6110 STATE

Current Address:	15H
Next Address:	10H
Branch Control:	000 No Branch
Branch Register:	0EH
Control:	20H (WG off)
Data Field:	00H
Count:	00H

System Activity: The AIC-6110 de-asserts Write Gate prior to writing Gap1.

AIC-6110 STATE

Current Address:	10H
Next Address:	14H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	80H (WG on)
Data Field:	00H Gap 1 Character
Count:	Set by host during initialization

System Activity: AIC-6110 has asserted Write Gate and the NRZ output is Gap 1 character. The host has detected the index pulse and has loaded the branch register with 12H (preparatory to writing the next sector when the present sector write is complete).

AIC-6110 STATE

Current Address:	14H
Next Address:	0FH
Branch Control:	000 No Branch
Branch Register:	12H
Control:	04H (Output also on)
Data Field:	00H
Count:	02H

System Activity: The output pin is turned on for 3 bytes time. This is interpreted by the drive as the assertion of the Address Mark Enable signal.

AIC-6110 STATE

Current Address:	0FH
Next Address:	06H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	00H
Data Field:	00H
Count:	0BH (Write 12 bytes of 00 sync code)

System Activity: AIC-6110 NRZ output is 00H Sync characters. The host is waiting for data transfer.

AIC-6110 STATE

Current Address:	06H
Next Address:	00H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	12H (Not valid with RG=0)
Data Field:	Sync byte required by the drive
Count:	1000 0000 (Not used with RG=0)

System Activity: AIC-6110 NRZ output is the host specified sync character. This is dependent on the drive.

AIC-6110 STATE

Current Address:	00H
Next Address:	01H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	12H (Not valid with RG=0)
Data Field:	Cylinder value established by host
Count:	00H (Write 1 byte)

System Activity: AIC-6110 NRZ output is I.D. cylinder, head, sector, flag bytes as microsequencer executes locations 00, 01, 02 and 03. The next location after 03H (Flag byte write) is 0CH (I.D. ECC write).

AIC-6110

Current Address:	0CH
Next Address:	11H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	00H
Data Field:	N.A.
Count:	0100 0011 (Write 4 bytes of ECC)

System Activity: AIC-6110 NRZ output is I.D. ECC characters (4 bytes).

AIC-6110 STATE

Current Address:	11H
Next Address:	08H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	00H
Data Field:	00H
Count:	02H (Write 4 bytes of ECC)

System Activity: The AIC-6110 writes 3 bytes of zeros after the I.D. ECC in order to pad this field.

AIC-6110 STATE

Current Address:	08H
Next Address:	09H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	20H (WG off)
Data Field:	00H
Count:	00H (1 byte Write splice)

System Activity: AIC-6110 de-asserts Write Gate for 1 byte time to cause a write splice.

AIC-6110 STATE

Current Address:	09H
Next Address:	0AH
Branch Control:	000 No Branch
Branch Register:	12H
Control:	80H
Data Field:	00H
Count:	Set-up by host during initialization

System Activity: AIC-6110 NRZ outputs as many bytes of 00H as specified by the host (this is Gap 2).

AIC-6110 STATE

Current Address:	0AH
Next Address:	04H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	02H (Not valid with RG=0)
Data Field:	Sync byte required by the drive
Count:	1010 0000 (CLKA and the bit ring will be synchronized)

System Activity: AIC-6110 NRZ output is the host specified sync character. This is dependent on the drive.

AIC-6110 STATE

Current Address:	04H
Next Address:	0DH
Branch Control:	000 No Branch
Branch Register:	12H
Control:	01H (Data Transfer bit set)
Data Field:	Set-up by host at initialization
Count:	FFh (256 byte or multiple)

System Activity: AIC-6110 NRZ output is the sector data from the host buffer or a fill value specified byt the host. Host buffer data will be transferred if the "Suppress Transfer" bit (Register 7A, Bit 5) is off. If this bit is set, then the fill byte will be output. At this point, the Data Transfer flag (Register 79, Bit 6) will be set and the host can update the sector I.D. registers. Setting "Inhibit Data Field Carry" will allow a second 256-byte transfer.

AIC-6110 STATE

Current Address:	0DH
Next Address:	12H
Branch Control:	100 Branch after ECC complete
Branch Register:	12H
Control:	00H
Data Field:	00H
Count:	43H (Write 4 ECC bytes)

System Activity: AIC-6110 NRZ output is Data Field ECC. At the end of the ECC, branch is active.

AIC-6110 STATE

Current Address:	12H
Next Address:	0EH
Branch Control:	000 No Branch
Branch Register:	12H
Control:	00H
Data Field:	00H
Count:	02H (Write 3 bytes of 00H bytes)

System Activity: IAC-6110 NRZ output is 3 bytes of 00H. This is the post data field ECC pad.

AIC-6110 STATE

Current Address:	0EH
Next Address:	10H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	20H (WG off)
Data Field:	00H
Count:	00

System Activity: AIC-6110 turns off Write Gate prior to writing gap 3.

AIC-6110 STATE

Current Address:	10H
Next Address:	14H
Branch Control:	000 No Branch
Branch Register:	12H
Control:	80H (WG on)
Data Field:	00H
Count:	Set by host during initialization

System Activity: At this point, the AIC-6110 is writing Gap 3, which is the same as when it was writing Gap 1. Following this, VFO sync field, the I.D. AM and the I.D. Field will be written. Then the Data AM and Data Field will be written. The sequence of writing I.D. and Data Fields will continue until the last sector is written. After the last sector, the host loads the Branch Register with 13H instead of 12H.

AIC-6110 STATE

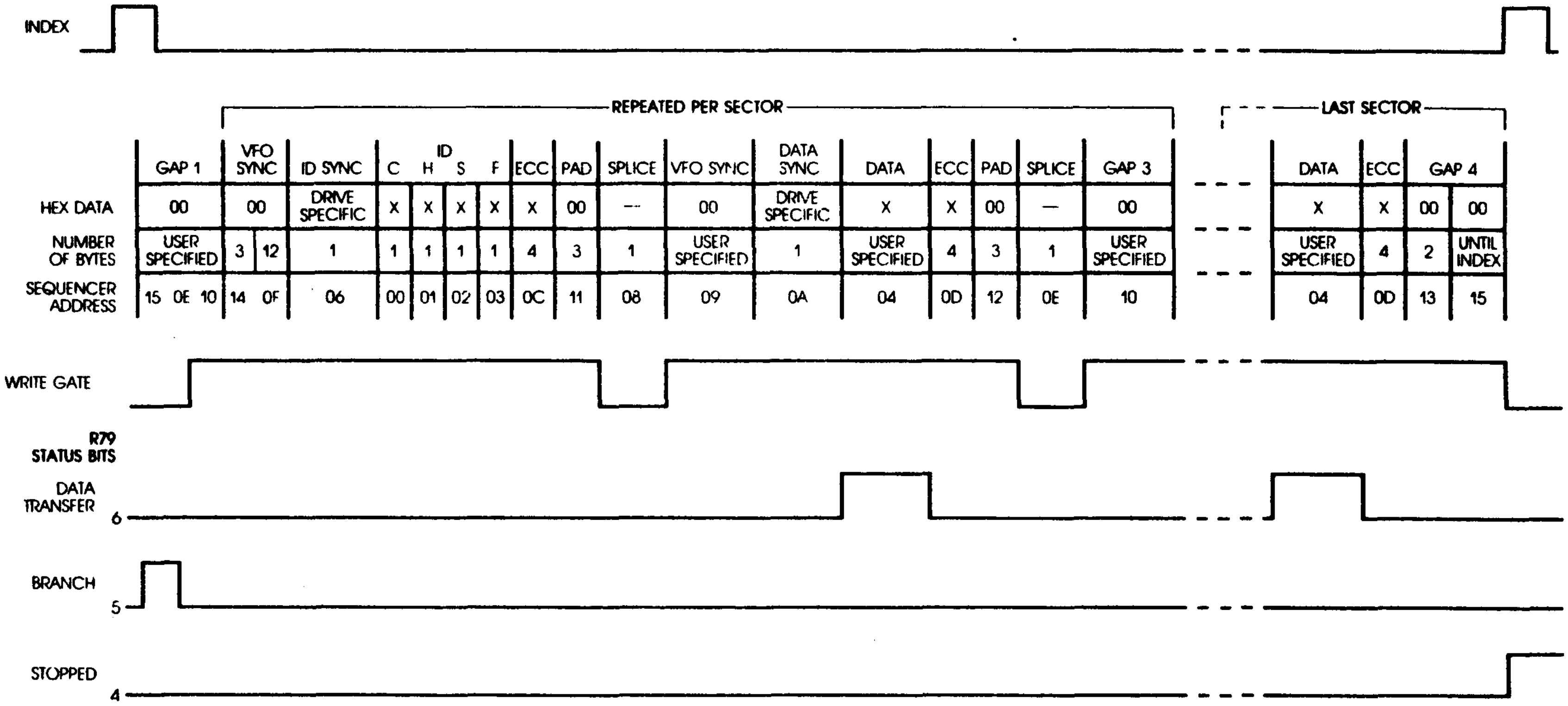
Current Address:	13H
Next Address:	15H
Branch Control:	000 No Branch
Branch Register:	13H
Control:	00H
Data Field:	00H
Count:	01H (Write 2 bytes of 00H bytes)

AIC-6110 STATE

Current Address:	15H
Next Address:	15H
Branch Control:	010 Stop on Index
Branch Register:	13H
Control:	00H
Data Field:	00H
Count:	00H (Write 1 byte of 00H then loop to 15H)

System Activity: The AIC-6110 continues to write 00H bytes until Index is encountered. At Index, the AIC-6110 stops and the track format is complete.

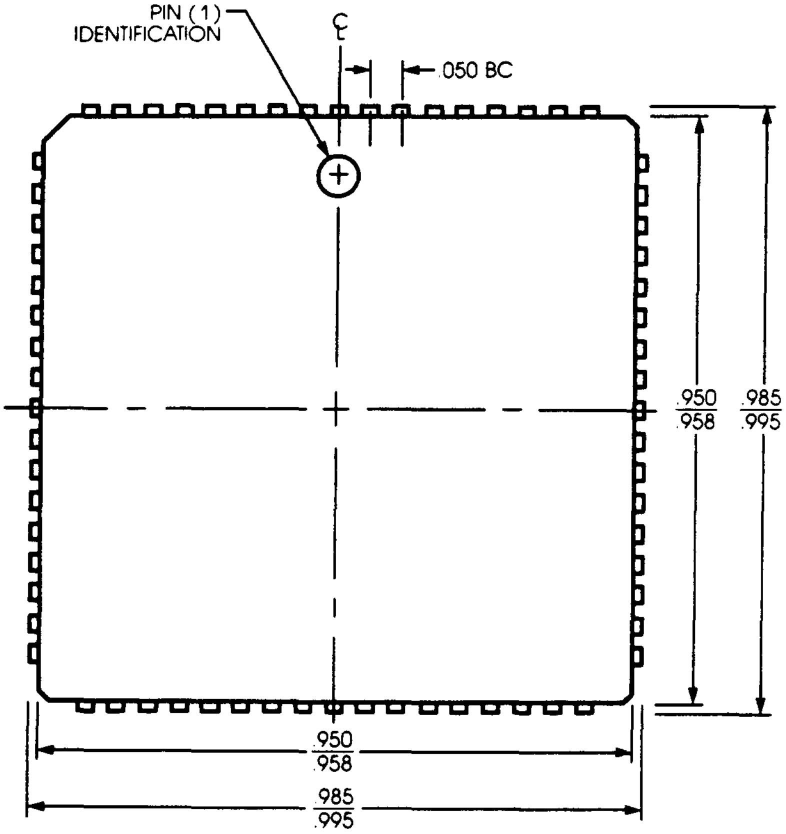
The track layout for the preceeding format sequence is shown on the following page.



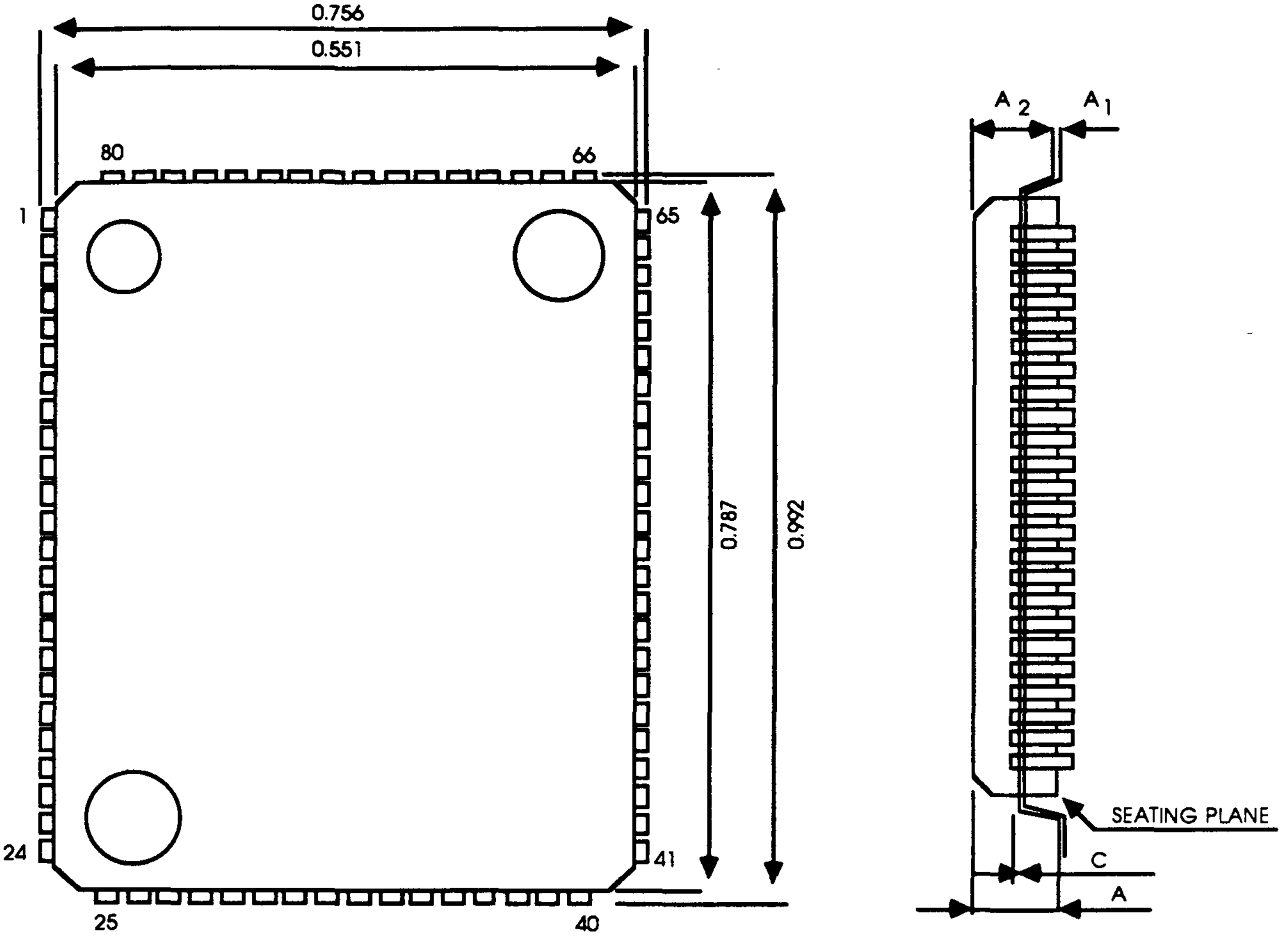
	REPEATED PER SECTOR																	LAST SECTOR			
	GAP 1	VFO SYNC	ID SYNC	ID C	H	S	F	ECC	PAD	SPLICE	VFO SYNC	DATA SYNC	DATA	ECC	PAD	SPLICE	GAP 3	DATA	ECC	GAP 4	
HEX DATA	00	00	DRIVE SPECIFIC	X	X	X	X	X	00	--	00	DRIVE SPECIFIC	X	X	00	—	00	X	X	00	00
NUMBER OF BYTES	USER SPECIFIED	3 12	1	1	1	1	1	4	3	1	USER SPECIFIED	1	USER SPECIFIED	4	3	1	USER SPECIFIED	USER SPECIFIED	4	2	UNTIL INDEX
SEQUENCER ADDRESS	15 0E 10	14 0F	06	00	01	02	03	0C	11	08	09	0A	04	0D	12	0E	10	04	0D	13	15

Soft-Sector Format Layout

Packaging Information - 68-pin PLCC AIC-6110



Packaging Information - 80-pin Quad Flat Pack AIC-6110


0.756
0.551
80
66
1
65
24
41
25
40
0.787
0.992
A_2
A_1
SEATING PLANE
C
A